ADVANCED METRICS WITH PROMETHEUS AND GRAFANA

Integration Guide

Applicable Services:

VirtuCrypt Advanced Metrics

TABLE OF CONTENTS

[1] INTEGRATION OVERVIEW 3

[1.1] Architecture 3

[1.2] VirtuCrypt CryptoTunnels 3

[1.3] Prometheus 3

[1.4] Grafana 4

[1.5] VirtuCrypt Monitoring Metric Reference 5

[2] PREREQUISITES 7

[3] CONFIGURATION IN THE VIRTUCRYPT INTELLIGENCE PORTAL (VIP) 8

[3.1] Log in to the VirtuCrypt Intelligence Portal (VIP) 8

[3.2] Create a CryptoTunnel for the connection between the CryptoTunnel Guardian and the Prometheus Proxy 8

[3.3] Enable the CryptoTunnel 9

[3.4] Download a PKCS #12 client TLS connection certificate for the customer Prometheus instance 10

[3.5] Test a connection to the Advanced Metrics endpoint 12

[4] PROMETHEUS INSTALLATION AND CONFIGURATION 13

[4.1] Prometheus Installation 13

[5] GRAFANA INSTALLATION AND CONFIGURATION 15

[5.1] Install Grafana 15

[5.2] Access Grafana's user interface 15

[5.3] Add and configure a new Data Source 15

[5.4] Create a new dashboard and add a panel to visualize metrics 15

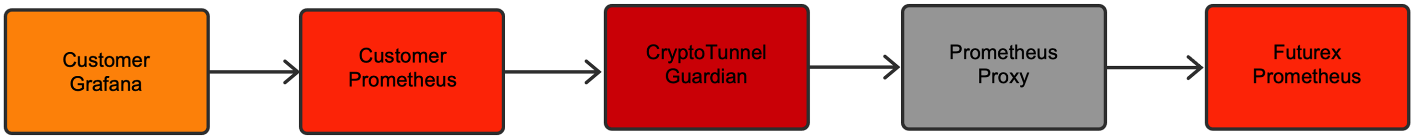
APPENDIX A: VIRTUCRYPT ENDPOINTS 18

APPENDIX B: XCEPTIONAL SUPPORT 19

[1] INTEGRATION OVERVIEW

[1.1] ARCHITECTURE

The overall architecture of this integration involves the components shown in the diagram below. Steps for configuring each of these components is included in the sections that follow.



[1.2] VIRTUCRYPT CRYPTOTUNNELS

In the VirtuCrypt world, trust is a two-way street. The CryptoTunnel uses three components to establish trust, starting with a private key local to your device. When you generate the PKI, creating the private key, the system signs the key under a VirtuCrypt CA tree, which is the second component. The VirtuCrypt CA tree that signed it is the authority that establishes trust between the server and the client. After the private key is signed under the CA tree, it becomes a signed certificate, the final component.

When you send the signed certificate through the CryptoTunnel, the server knows the certificate is signed under the VirtuCrypt CA tree and thus is authentic. That is how the server establishes trust in the application.

To establish trust in the opposite direction, from the application to the server, the server sends the server-side signed certificate to the application. The application client then validates the server identity, establishing the trusted relationship with mutual authentication.

After this handshake, you can encrypt all the data, satisfying PCS-DSS compliance requirements.

[1.3] PROMETHEUS

Prometheus is an open-source systems monitoring and alerting toolkit. It was originally developed by SoundCloud in 2012 and is now a graduated project of the Cloud Native Computing Foundation, which is part of the Linux Foundation and also hosts projects like Kubernetes and Fluentd.

Prometheus's main features are:

1. **Multi-dimensional data model:** Prometheus stores all data as time series, and each time series is uniquely identified by its metric name and a set of key-value pairs, also known as labels.
2. **PromQL (Prometheus Query Language):** Prometheus provides a flexible query language to leverage its dimensional data model. PromQL allows you to select and aggregate time series data in real time.
3. **No reliance on distributed storage:** Prometheus's main unit of reliability is the individual node, which is fully standalone, not depending on network storage or other remote services.

4. **Collection happens via pull model:** Prometheus collects metrics from monitored targets by scraping HTTP endpoints on these targets. However, it also supports an intermediary gateway for scenarios where a pull model is not suitable.
5. **Targets are discovered via service discovery or static configuration:** Prometheus employs various service discovery mechanisms to dynamically discover scrape targets.
6. **Multiple modes of graphing and dashboarding support:** While Prometheus itself provides built-in expression browser for exploring metrics, it also seamlessly integrates with graphical dashboard builder like Grafana for advanced visualization.
7. **Alerting functionality:** Prometheus has a highly flexible alerting system. It allows you to define alerting rules for your metrics, and if those conditions are met, it sends alert notifications via its Alertmanager component.

Prometheus is designed for reliability, to be the system you go to during an outage to allow you to quickly diagnose problems. It is used by many organizations for monitoring their IT infrastructure, from microservices, containers, and Kubernetes at scale to IoT devices. It also supports a robust ecosystem of exporters for extending its monitoring capabilities.

[1.4] GRAFANA

Grafana is a popular open-source tool for visualizing large-scale measurement data. It provides a powerful and elegant way to create, explore, and share dashboards and data with your team and the world.

Grafana is most commonly used for visualizing time series data for infrastructure and application analytics, but it's also used in other domains including industrial sensors, home automation, weather, and process control. It supports a wide variety of data sources, including but not limited to Prometheus, InfluxDB, Elasticsearch, AWS CloudWatch, MySQL, and PostgreSQL.

Here are some key features of Grafana:

1. **Dashboard and Visualizations:** Grafana provides a feature-rich data-modeling interface for creating dashboards. These dashboards can contain a variety of visualization widgets or panels (such as graphs, tables, single stats, gauges, maps, etc.). It's easy to switch the visualization type to compare different visual formats of the same data.
2. **Data Source Support:** Grafana supports a plethora of databases and data sources, from time-series databases to relational databases and cloud services. You can create dashboards that pull data from multiple sources for a unified view.
3. **Alerting:** Grafana provides robust alerting functionality. You can define alert rules for your data and get notified via several channels when an alert is triggered.
4. **Annotations:** Grafana allows you to annotate your graphs with rich events when something noteworthy happens. This helps correlate the insights between different events and metrics.
5. **Dashboard Sharing:** Dashboards can be shared in various ways - by link, as a snapshot, as a PDF, or embedded in other web pages. This makes it easy to collaborate with your team.

6. **Teams and Authentication:** Grafana supports user authentication, allowing you to control access to your dashboards. It also has a multi-tenant architecture, so you can set up and manage multiple independent organizations, each with their own users, dashboards, and data sources.
7. **Plugins:** Grafana features a plug-in architecture and offers a plethora of plugins that allow you to extend and customize Grafana's capabilities.

Grafana is a powerful tool for building visual dashboards for observing metrics in real-time and is widely used in various industries.

[1.5] VIRTUCRYPT MONITORING METRIC REFERENCE

Note: Please refer to section 5.4 for instructions on how to select and visualize metrics in Grafana.

Metric Usage

Format: `example_metric{label_1=0, label_2=us-east}`

Metric Name	Type	Description	Labels
`ct_instance_port_status`	Gauge int	**CT Instance Port Status** (open -> 1 or closed -> 0)	company_name (str), host (str), region (str), tunnel_name (str)
`ct_instance_api_type`	Gauge int	**CT Instance API Type**	company_name (str), host (str), region (str), tunnel_name (str)
`ct_instance_service_enabled`	Gauge int	**CT Instance Service Enabled** (True -> 1, False -> 0)	company_name (str), host (str), region (str), tunnel_name (str)
`ct_instance_service_latency_ms`	Gauge int	**CT Instance Service Latency in ms**	company_name (str), host (str), region (str), tunnel_name (str)
`ct_instance_accepting_connections`	Gauge int	**CT Instance Accepting Connections** (True -> 1, False -> 0)	company_name (str), host (str), region (str), tunnel_name (str)
`ct_instance_certificate_validity`	Gauge int	**CT Instance Certificate Validity**	company_name (str), host (str), region (str), tunnel_name (str)
`ct_instance_clients_connected_total`	Gauge int	**Total clients connected to CT instance**	company_name (str), host (str), region (str), tunnel_name (str)

API Type Mappings

Value	Mapping
0	"None"
1	"International"
2	"Excrypt"
3	"JSON"

Certificate Validity Mappings

Value	Mapping
1	"Max Validity"
2	"Under 90 Days"
3	"Under 60 Days"
4	"Under 30 Days"
5	"Under 7 Days"
6	"Expired"

[2] PREREQUISITES

- VIP account with the **Advanced Metrics** feature enabled
- OpenSSL

[3] CONFIGURATION IN THE VIRTUCRYPT INTELLIGENCE PORTAL (VIP)

This section walks users through the process for logging in to the VirtuCrypt Intelligence Portal (VIP), creating a new CryptoTunnel to allow connections between the CryptoTunnel Guardian and the Prometheus Proxy defined by the VIP user, and downloading a PKCS #12 client TLS connection certificate for the customer Prometheus instance.

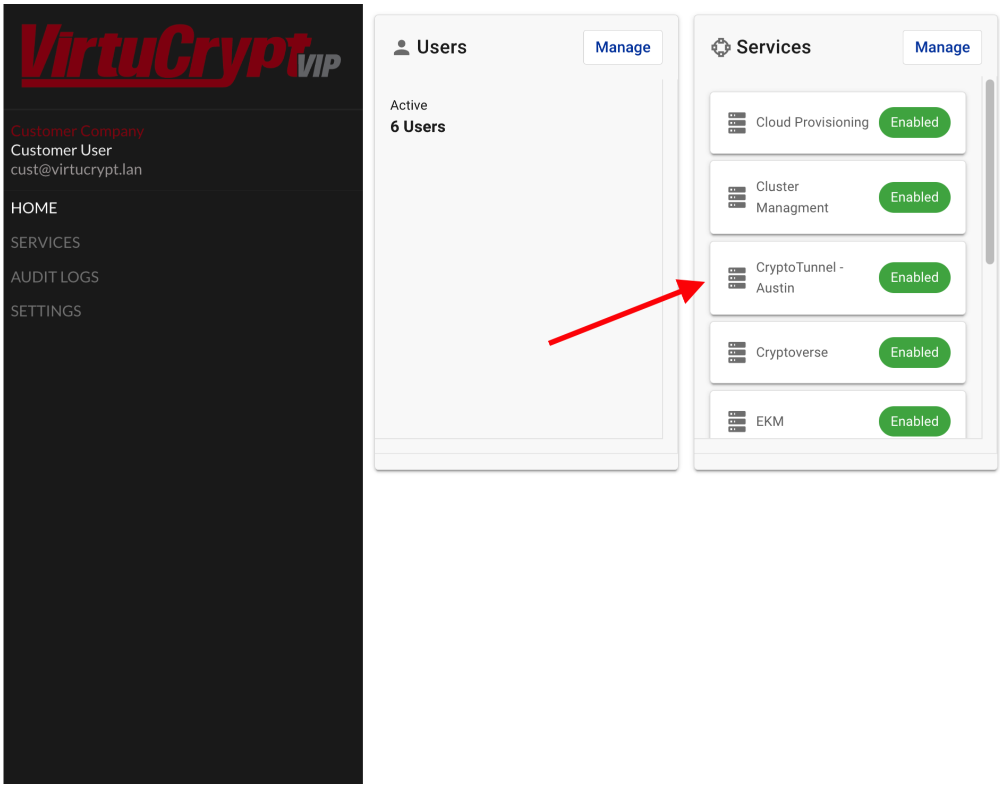
[3.1] LOG IN TO THE VIRTUCRYPT INTELLIGENCE PORTAL (VIP)

1. Log in at https://vip.virtucrypt.com/login with an account identity that is authorized to access the **CryptoTunnel** and CryptoVerse services in your VIP account.

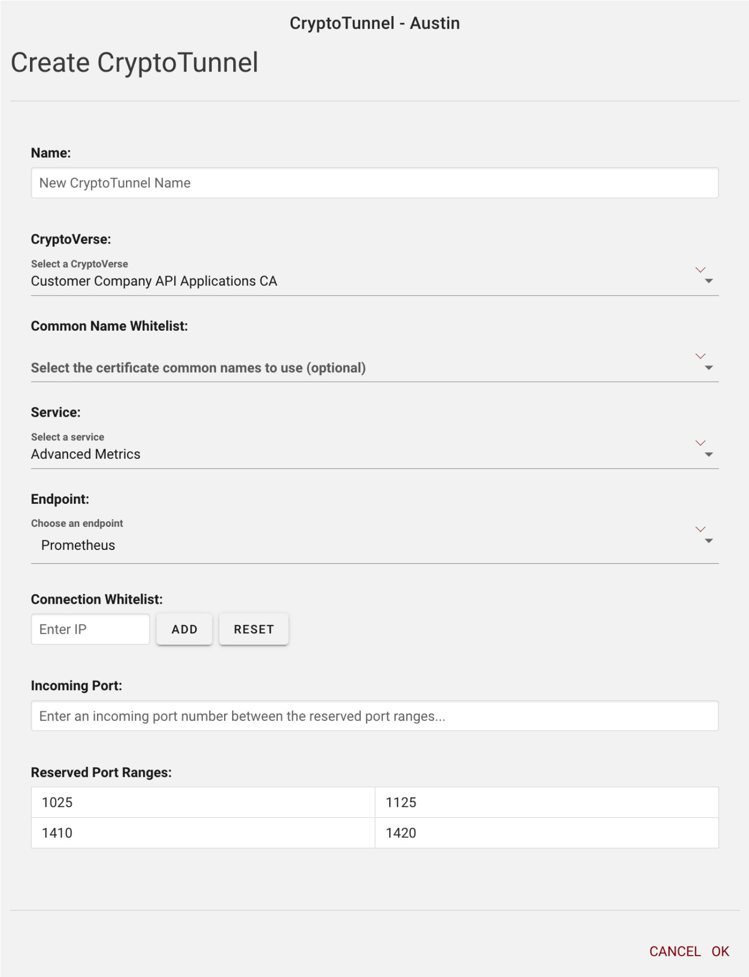
[3.2] CREATE A CRYPTOTUNNEL FOR THE CONNECTION BETWEEN THE CRYPTOTUNNEL GUARDIAN AND THE PROMETHEUS PROXY

Note: If Advanced Metrics is set up for your VIP account, you will have at least two CryptoTunnels. One for the service you are using and another one for the metrics related to that service.

1. Select **CryptoTunnel** in the services list.



2. Select the **[Add CryptoTunnel]** button at the top of the page.
3. Configure the CryptoTunnel per the normal process, selecting **Advanced Metrics** in the Service dropdown menu.



CryptoTunnel - Austin

Create CryptoTunnel

Name:

New CryptoTunnel Name

CryptoVerse:

Select a CryptoVerse

Customer Company API Applications CA

Common Name Whitelist:

Select the certificate common names to use (optional)

Service:

Select a service

Advanced Metrics

Endpoint:

Choose an endpoint

Prometheus

Connection Whitelist:

Enter IP

Incoming Port:

Enter an incoming port number between the reserved port ranges...

Reserved Port Ranges:

1025	1125
1410	1420

Note: CryptoVerse is the company-specific applications CA used to verify clients.

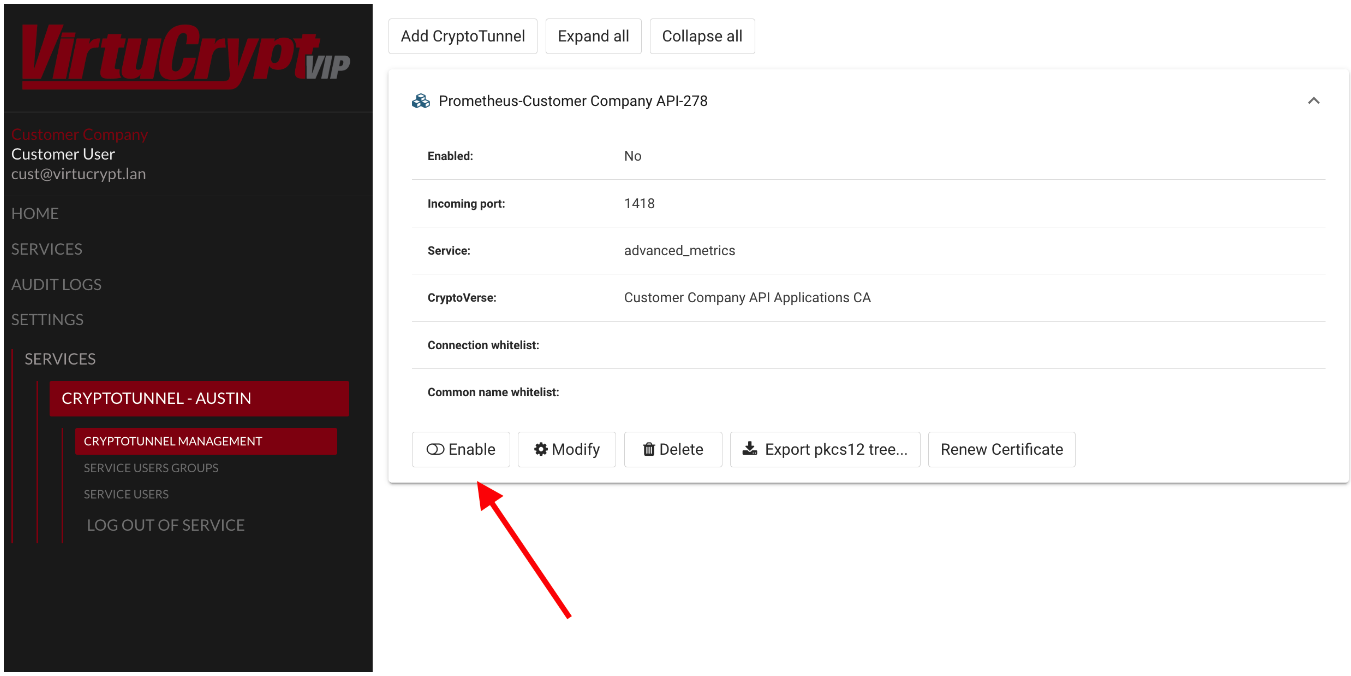
Note: Endpoint is the destination the CryptoTunnel Guardian will connect to after proxying users.

4. Click **[OK]**.

[3.3] ENABLE THE CRYPTOTUNNEL

1. The new CryptoTunnel should be listed now in the **CryptoTunnel Management** menu. Expand it by selecting the down arrow on the far right.

2. Select **[Enable]**.

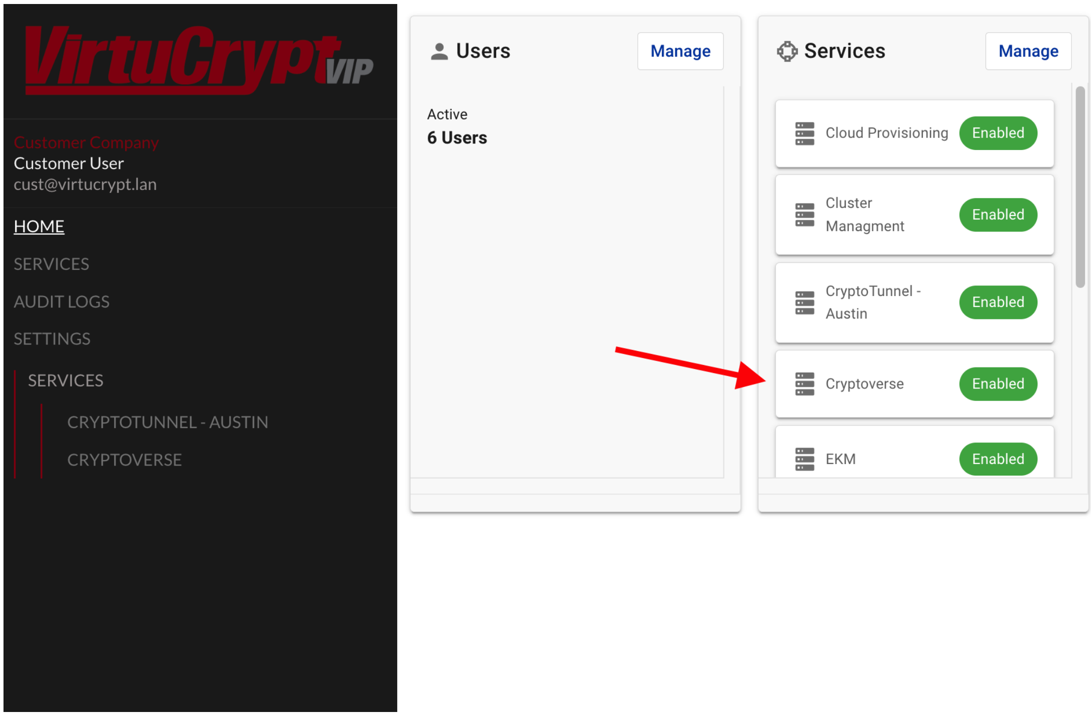


Prometheus-Customer Company API-278

Enabled:	No
Incoming port:	1418
Service:	advanced_metrics
CryptoVerse:	Customer Company API Applications CA
Connection whitelist:	
Common name whitelist:	

[3.4] DOWNLOAD A PKCS #12 CLIENT TLS CONNECTION CERTIFICATE FOR THE CUSTOMER PROMETHEUS INSTANCE

1. Navigate back to the home page of your VIP account.
2. Under **Services**, select **CryptoVerse**.

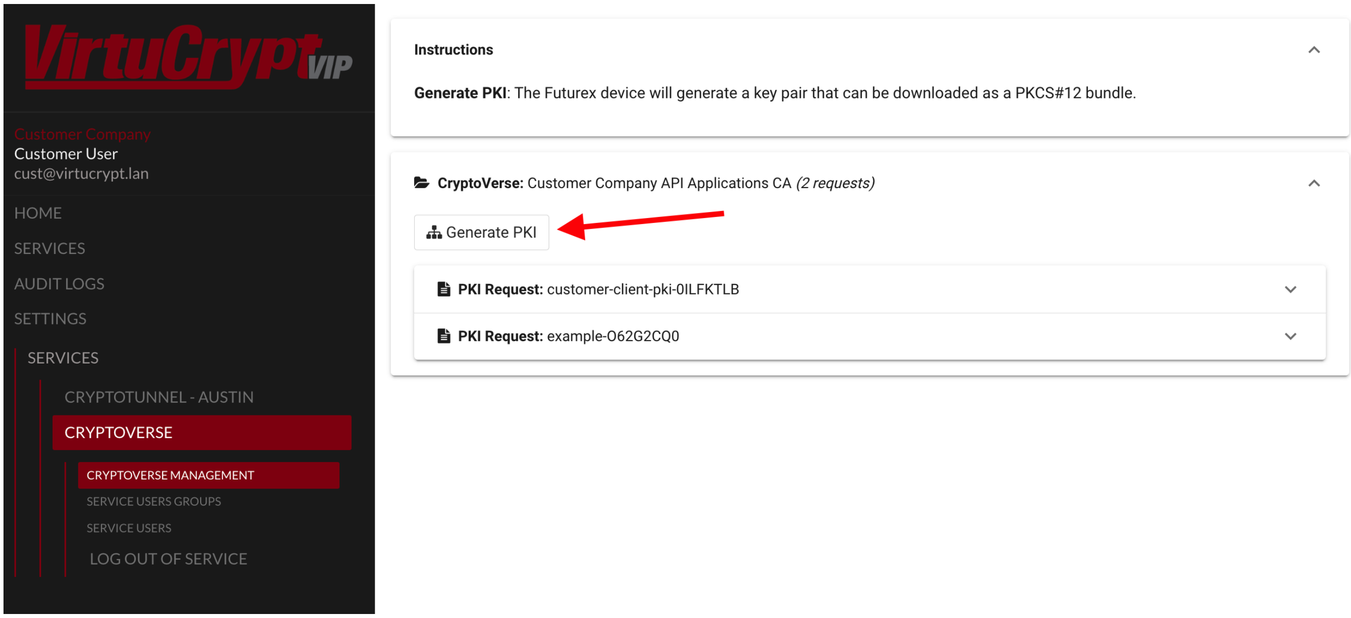


Active
6 Users

Cloud Provisioning	Enabled
Cluster Managment	Enabled
CryptoTunnel - Austin	Enabled
Cryptoverse	Enabled
EKM	Enabled

3. Expand the company-specific applications CA by clicking the down arrow on the far right.

4. Click the **[Generate PKI]** button.



Instructions

Generate PKI: The Futurex device will generate a key pair that can be downloaded as a PKCS#12 bundle.

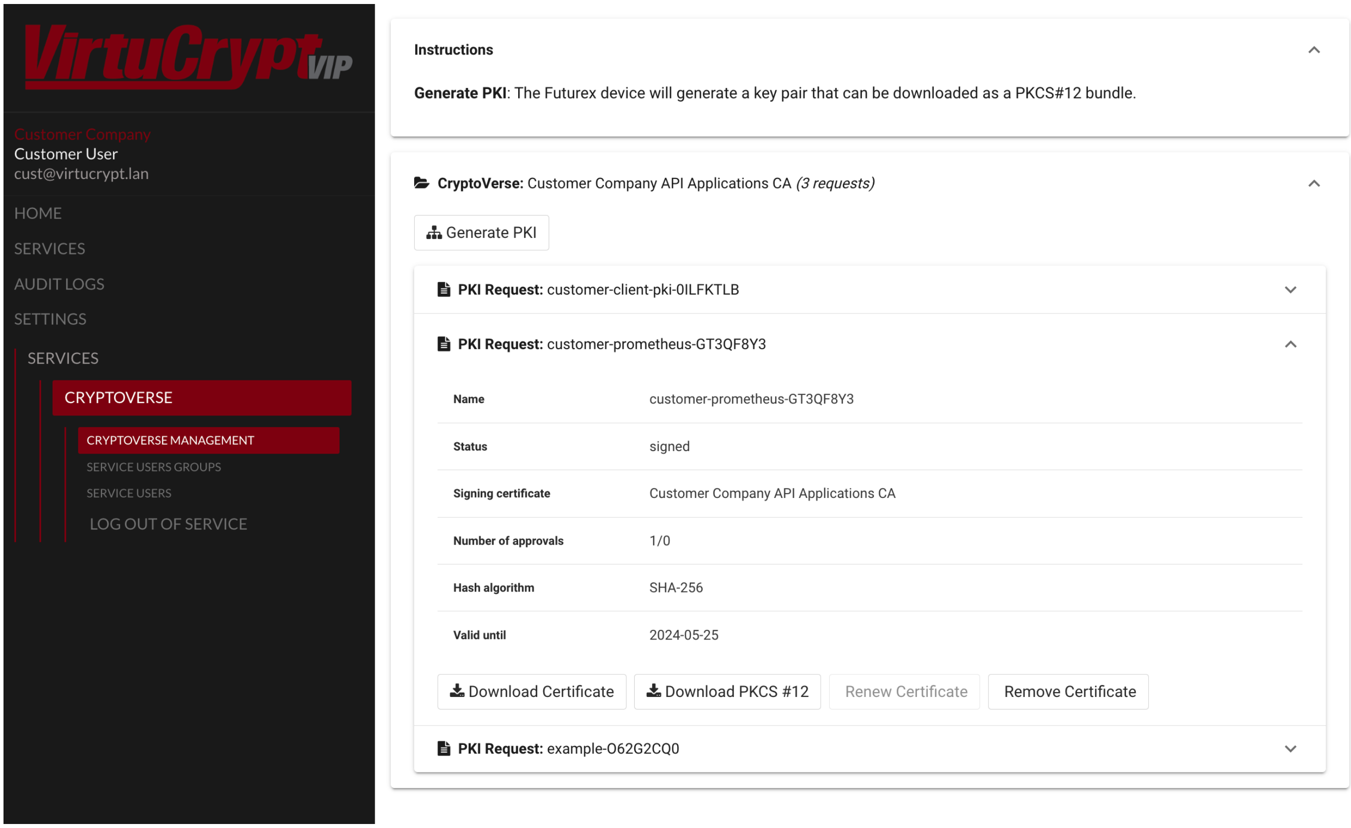
CryptoVerse: Customer Company API Applications CA *(2 requests)*

PKI Request: customer-client-pki-0ILFKTLB

PKI Request: example-062G2CQ0

5. Enter a **Name** and **Password** for the PKI, then click **[OK]**.

6. Expand the PKI you just generated to display information about the certificate.



Instructions

Generate PKI: The Futurex device will generate a key pair that can be downloaded as a PKCS#12 bundle.

CryptoVerse: Customer Company API Applications CA *(3 requests)*

PKI Request: customer-client-pki-0ILFKTLB

PKI Request: customer-prometheus-GT3QF8Y3

Name	customer-prometheus-GT3QF8Y3
Status	signed
Signing certificate	Customer Company API Applications CA
Number of approvals	1/0
Hash algorithm	SHA-256
Valid until	2024-05-25

PKI Request: example-062G2CQ0

7. Click the **[Download PKCS #12]** button, then enter the PKCS #12 password and select **[OK]**.

8. Select the location where you want to save the PKCS #12 file locally.

[3.5] TEST A CONNECTION TO THE ADVANCED METRICS ENDPOINT

Run the **curl** command below to confirm that you're able to successfully connect to your Advanced Metrics endpoint using the TLS certificates contained within the PKCS #12 file:

```
curl -k 'https://us01crypto01test.virtucrypt.com:3126/api/v1/query' --cert-type P12 --cert Test_
App-Y96S6OJG.p12:safest --data-urlencode 'query=ct_instance_service_latency_ms{}' | jq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   400    0   358  100    42   2546    298 --:--:-- --:--:-- --:--:--  2962
{
  "status": "success",
  "data": {
    "resultType": "vector",
    "result": [
      {
        "metric": {
          "__name__": "ct_instance_service_latency_ms",
          "company_name": "Futurex Demo Applications",
          "host": "austestcryptoguard02",
          "instance": "austestcryptoguard02-exporter:8000",
          "job": "vc-monitor",
          "region": "us-central-1",
          "tunnel_name": "testapp1-VirtuCrypt Demo-1961"
        },
        "value": [
          1686941969.131,
          "0"
        ]
      }
    ]
  }
}
```

[4] PROMETHEUS INSTALLATION AND CONFIGURATION

Instructions are provided below for installing, configuring, and running Prometheus locally.

[4.1] PROMETHEUS INSTALLATION

The instructions below for downloading, installing, and configuring Prometheus are tailored towards Unix-like systems, such as Linux and MacOS, but the process is similar for Windows with minor changes.

[4.1.1] Download Prometheus

1. Open your web browser and go to the official Prometheus download page at https://prometheus.io/download/.
2. Find the appropriate version for your operating system under the "Prometheus" section and click on it to start the download.

[4.1.2] Extract the Downloaded Archive

1. Once the download is complete, open your terminal.
2. Navigate to the directory where you downloaded the Prometheus archive.
3. Extract the downloaded file using the tar command:

```
tar xvfz prometheus-*.tar.gz
```

4. This command will create a new directory named something like "**prometheus-2.x.x**". Navigate into this directory:

```
cd prometheus-*
```

[4.1.3] Extract the client certificate, client private key, and CA Certificate chain from the PKCS #12 file for configuration in Prometheus

In a terminal, use the following OpenSSL commands to extract the client certificate, client private key, and CA certificate tree from the PKCS #12 file you downloaded in the VirtuCrypt Intelligence Portal.

Note: The PKCS #12 file name needs to modified in each of the commands below to match the name of your PKCS #12 file.

Note: Each of these commands will prompt for the PKCS #12 file password.

1. Extract the client certificate.

```
openssl pkcs12 -in pki.p12 -clcerts -nokeys -out clientcert.pem
```

2. Extract the client private key.

```
openssl pkcs12 -in pki.p12 -nocerts -nodes -out clientkey.pem
```

3. Extract the CA certificate chain.

```
openssl pkcs12 -in certificate.p12 -cacerts -nokeys -out cacert.pem
```

Note: The client certificate, client private key, and CA certificate chain PEM files will be configured inside the Prometheus configuration file in the next section.

[4.1.4] Configure Prometheus

1. Create a file named "prometheus.yml" in the extracted **Prometheus** directory. This file will contain the configuration settings for Prometheus. Here's a sample configuration to get started:
2. Here's an example configuration that instructs Prometheus to monitor a VirtuCrypt endpoint at us01crypto01test.virtucrypt.com on port 1234. Please refer to Appendix A for the full list of public VirtuCrypt endpoints.

```
global:
  scrape_interval:     3s

  external_labels:
      monitor: 'virtucrypt'

scrape_configs:
  - job_name: 'prometheus'
    static_configs:
      - targets: ['localhost:9090']

  - job_name: 'TLS_Test'
    metrics_path: '/federate'
    params:
      'match[]':
        - '{__name__=~".+"}'
    static_configs:
      - targets: ['us01crypto01test.virtucrypt.com:1234']
    scheme: 'https'
    tls_config:
      ca_file: '/certs/cacerts.pem'
      cert_file: '/certs/clientcert.pem'
      key_file: '/certs/clientkey.pem'
```

[4.1.5] Run Prometheus

1. Save the **prometheus.yml** file and close your text editor.
2. In your terminal, run the Prometheus binary with:

```
./prometheus --config.file=prometheus.yml
```

3. Prometheus should now be running and available at **http://localhost:9090/** by default.

[5] GRAFANA INSTALLATION AND CONFIGURATION

The steps below outline how to configure a Prometheus connection in Grafana so you can create dashboards and panels that use this data source.

[5.1] INSTALL GRAFANA

1. Install Grafana in either Linux, macOS, or Windows by following the instructions on Grafana's official website.

[5.2] ACCESS GRAFANA'S USER INTERFACE

1. Access Grafana's user interface by navigating to its IP address or domain name. For example, if Grafana is running on your local machine, the URL would be **http://localhost:3000**.
2. Log in with your admin account.

[5.3] ADD AND CONFIGURE A NEW DATA SOURCE

1. In the left menu, select **Configuration** > **Data Sources**.
2. Click the **[Add new data source]** button.
3. Select **Prometheus** as the data source type.
4. Provide the necessary details for the Prometheus data source:
    - **Name**: Enter a name for the data source (e.g., "Prometheus").
    - **URL**: Enter the URL of your Prometheus server (e.g., **http://localhost:9090**).
    - **Auth**: Select an authentication option based on your Prometheus server's configuration.
    - **Basic Auth**: Enable or disable based on your Prometheus server's configuration. If your Prometheus server uses TLS (HTTPS), you might need to configure the TLS Auth settings. Here, you'll have to specify the Certificate, Client Certificate, Client Key, and CA Certificates for your Prometheus instance. Please consult the Prometheus documentation if you're unsure about these settings.
    - Other settings you can configure include **Alterting**, **Type and version**, **Misc**, and **Exemplars**. Please consult the Grafana documentation if you're unsure about these settings.
5. Click the **[Save & test]** button at the bottom of the page when finished.

[5.4] CREATE A NEW DASHBOARD AND ADD A PANEL TO VISUALIZE METRICS

A Grafana dashboard is a customizable container for panels, organized and laid out in a grid-like structure. It allows users to visualize, analyze, and track metrics and logs from various data sources in real-time. Dashboards can be interactive, and they serve as a central space for monitoring key data points, displaying performance metrics, identifying trends, and exploring potential issues.

A panel is a basic visual element in Grafana, which can hold a graph, single stat, table, map, or other types of data visualization.

Follow the steps below to create a new dashboard and panel for visualizing metrics pulled from Prometheus.

Create a new dashboard

1. In the left menu, select **Dashboards** > **New dashboard**. A new dashboard with an empty panel will be created.

Add a panel

1. On your new dashboard, you'll see a box prompting you to create a new panel or row. Click **[Add a new panel]**. This will bring you to the panel editor screen.

Configure the panel

1. Under the **Queries** tab, you can configure your panel's data source and set up queries. Click on the drop-down menu next to **Data source**. You'll see a list of all data sources that have been added to Grafana. Select the your Prometheus data source.
2. After selecting your Prometheus data source, you can create your query. In the **Metric** dropdown, select the metric you want to visualize in this panel. Optionally, you can configure **Label filters** as well.
3. Now, click **[Run queries]** button to run the query you configured and see the results visualized in the graph above.

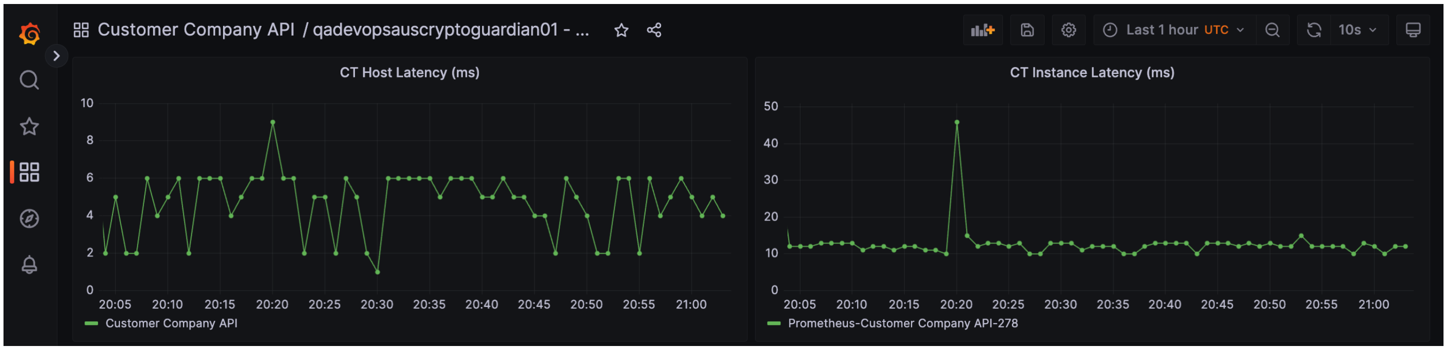
Customize the panel

1. After setting up your queries, you can move on to visualizing the data. Click on the **Visualization** tab next to **Queries**. Here, you can select the type of visualization (graph, gauge, table, etc.) and customize it to your liking.
2. Next, click on the **General** tab. Here you can name your panel and add a description.
3. After you have configured your panel, click on the **Apply** button in the top right corner.

Save the dashboard

1. After adding and setting up your panel, remember to save your dashboard. Click the disk icon in the top right corner.
2. A save dashboard dialog will appear, asking for a name and description for your dashboard. Fill these in and click the **[Save]** button.

Below is an example Dashboard with a couple of Panels added to visual Latency metrics.


Customer Company API / qadevopsauscryptoguardian01 - ...
Last 1 hour UTC
10s
CT Host Latency (ms)
10
8
6
4
2
0
20:05
20:10
20:15
20:20
20:25
20:30
20:35
20:40
20:45
20:50
20:55
21:00
Customer Company API
CT Instance Latency (ms)
50
40
30
20
10
0
20:05
20:10
20:15
20:20
20:25
20:30
20:35
20:40
20:45
20:50
20:55
21:00
Prometheus-Customer Company API-278

APPENDIX A: VIRTUCRYPT ENDPOINTS

Public DNS Names

Every connection to VirtuCrypt will have endpoint details (PUBLIC_DNS_NAME:PORT). Here is a list of some of the data centers. If you need help finding your endpoint details, please email futurex_support@futurex.com and we will be glad to help identity the correct endpoint and ports for your VirtuCrypt service.

Test

Austin Test Public DNS Name: us01crypto01test.virtucrypt.com

Phoenix Test Public DNS Name: us02crypto01test.virtucrypt.com

Amsterdam Test Public DNS Name: eu-netherlands-west-crypto-uat.virtucrypt.com

Mumbai Test Public DNS Name: ap-india-west-crypto-uat.virtucrypt.com

Hyderabad Test Public DNS Name: ap-india-central-crypto-uat.virtucryt.com

Production

Austin Production Public DNS Name: us01crypto01.virtucrypt.com

Phoenix Production Public DNS Name: us02crypto01.virtucrypt.com

Amsterdam Production Public DNS Name: eu-netherlands-west-crypto.virtucrypt.com

Frankfurt Production Public DNS Name: eu-germany-west-crypto.virtucrypt.com

Mumbai Production Public DNS Name: ap-india-west-crypto.virtucrypt.com

Hyderabad Production Public DNS Name: ap-india-central-crypto.virtucryt.com

Singapore Production Public DNS Name: ap-singapore-crypto.virtucrypt.com

Kamloops Production Public DNS Name: kidc01prod01.virtucrypt.com

Quebec Production Public DNS Name: qidc-1prod01.virtucrypt.com

APPENDIX B: XCEPTIONAL SUPPORT

In today's high-paced environment, we know you are looking for timely and effective resolutions for your mission-critical needs. That is why our Xceptional Support Team does whatever it takes to ensure you have the best experience and support possible. Every time. Guaranteed.

- 24x7x365 mission critical support
- Level 1 to level 3 support
- Extremely knowledgeable subject matter experts

At Futurex, we strive to supply you with the latest data encryption innovations as well as our best-in-class support services. Our Xceptional Support Team goes above and beyond to meet your needs and provide you with exclusive services that you cannot find anywhere else in the industry.

- Technical Services
- Onsite Training
- Virtual Training
- Customized Consulting
- Customized Software Solutions
- Secure Key Generation, Printing, and Mailing
- Remote Key Injection
- Certificate Authority Services

Toll-Free: 1-800-251-5112

E-mail: support@futurex.com

ENGINEERING CAMPUS

864 Old Boerne Road
Bulverde, Texas, USA 78163
Phone: +1 830-980-9782
+1 830-438-8782
E-mail: info@futurex.com

XCEPTIONAL SUPPORT

24x7x365
Toll-Free: 1-800-251-5112
E-mail: support@futurex.com

SOLUTIONS ARCHITECT

E-mail: solutions@futurex.com