appviewX

CLMaaS-AppViewX Cloud Connector User Guide

Version: 2021.1.0

Copyright AppViewX, Inc.

External Reference Links

This product includes software developed by the CentOS Project (www.centos.org).

This product includes software developed by Red Hat, Inc. (www.redhat.com).

This product includes software developed by VMware, Inc. (www.vmware.com).

Contact Information

AppViewX, Inc.

222 Broadway, FL 19

New York, NY 10038

Email: info@appviewx.com

Web: www.appviewx.com

Contents

Preface 5

Revision History 5

About this Guide 5

Audience 5

Text Conventions 5

Chapter 1. AppViewX CLM as a Service 6

AppViewX CLM as a Service (CLMaaS) 6

Overview 6

AppViewX CLM as a Service (CLMaaS) 6

The AppViewX Cloud Connector 7

Chapter 2. Features of the AppViewX Cloud Connector 8

Data Center-based Routing 8

Cloud Connector High Availability 9

Custom Certificates for Core Communication 9

Chapter 3. Setting up the AppViewX Cloud Connector 10

Step 1: Checking Prerequisites 10

Hardware 10

Operating System 10

Docker Prerequisites 10

Server and Network Prerequisites 11

Executing the Prerequisite Check Script 12

Step 2: Accessing the Setup Interface 15

Step 3: Installing the AppViewX Cloud Connector 17

Configuring Network Proxy Settings 17

Downloading the Installer 19

Installing the AppViewX Cloud Connector Agent 23

Reviewing the Installation 26

Chapter 4. Managing F5 BIG IP and A10 Devices..... 27
Managing F5 BIG-IP and A10 Devices..... 27
F5 BIG-IP Devices..... 27
A10 Devices..... 27
Chapter 5. Troubleshooting the AppViewX Cloud Connector..... 28
Troubleshooting the AppViewX Cloud Connector..... 28
Managing Certificates on F5 BIG-IP Devices..... 28
AppViewX Cloud Connector Health..... 28
Connectivity Checks..... 29
Installation Errors..... 29
Log Analysis..... 33
Checking Pod Status..... 33
Restarting Pods..... 34
Chapter 6. Managing the AppViewX Cloud Connector..... 36
Managing the AppViewX Cloud Connector..... 36
Understanding the AppViewX Cloud Connector Inventory..... 36
AppViewX Cloud Connector Actions..... 41
Monitoring the Health of the AppViewX Cloud Connector..... 46

Preface

Revision History

Revision	Description	Date
1.0	Initial release of document for Release 2021.1.0	September 2021

About this Guide

The guide introduces you to the features of the AppViewX Cloud Connector, the component that facilitates a SaaS deployment of AppViewX's flagship product, CERT+: the AppViewX Certificate Lifecycle Management as a Service (CLMaaS). The guide also includes steps for installing, configuring, managing, and troubleshooting your AppViewX Cloud Connector instance.

Audience

This guide is intended for AppViewX's customers looking to install the AppViewX Cloud Connector for the CLMaaS deployment.

Text Conventions

The following text conventions are used in this document:

Convention	Description
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`codeblock`	Indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Chapter 1: AppViewX CLM as a Service

- AppViewX CLM as a Service (CLMaaS)

AppViewX CLM as a Service (CLMaaS)

- Overview
- AppViewX CLM as a Service (CLMaaS)
- The AppViewX Cloud Connector

Overview

AppViewX's CERT+ product is now available as a Software as a Service (SaaS) offering for end-to-end Certificate Lifecycle Management (called CLMaaS).

AppViewX CLM as a Service (CLMaaS)

AppViewX is a next-gen Machine Identity Management and Automation Platform to simplify and enhance enterprise security with a unified approach for certificate lifecycle management.

Key highlights of the AppViewX CLMaaS:

- Discover, monitor, analyze, orchestrate and fully automate certificate lifecycle management and key management solutions
- Make a shift from reactive mode and be more proactive as you get a complete view of your entire certificate infrastructure.
- Manage certificates as a service with pre-built integrations and extensible APIs that plugin to your enterprise applications, web servers, microservices, and multi-cloud environments.
- Analyze certificates for crypto standards like key size, cipher strength, and allowed protocol versions
- Setup policies for enforcing high crypto standards
- Update certificates as per new policies
- Provision certificates for devices and applications
- Save resources, time, and effort of installation and maintenance.

Note: For information related to these functions, refer to the **AppViewX CLMaaS CERT+ User Guide**.

The AppViewX Cloud Connector

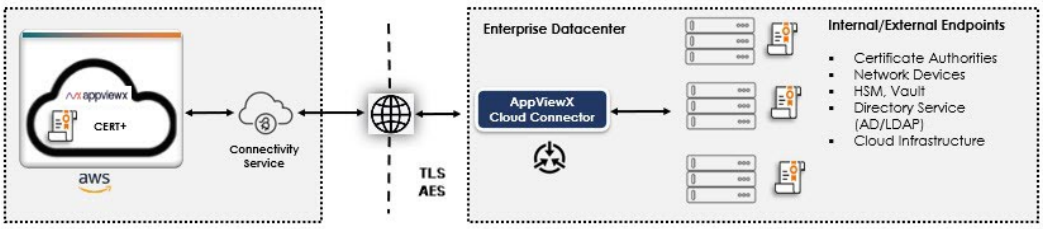
The AppViewX Cloud Connector faciliates the cloud-based deployment of CERT+. It is a secure proxy that is deployed within an enterprise's network to communicate with AppViewX CLMaaS.

The AppViewX Cloud Connector will be applicable in the following scenarios:

- Discovering certificates from an endpoint within the network (across all cloud and network endpoints in that network)
- Discovering certificates from public Certificate Authorities (CAs)

  In this case, AppViewX provides a default instance of the Cloud Connector called **cloud-dc**.

  To enable this, at the time of adding a new AppViewX Cloud Connector instance, from the **Data Center** dropdown list, select **DC Routing**.



The AppViewX Cloud Connector also comes with the following features:

- A self-serviceable, Linux-based lightweight setup
- Secure communication between the CLMaaS and the AppViewX Cloud Connector using TLS and AES encryption
- Connectivity from the CLMaaS to the enterprises' network endpoints
- No complex network setup (Inbound Firewall Whitelisting, VPN setup, and so on)

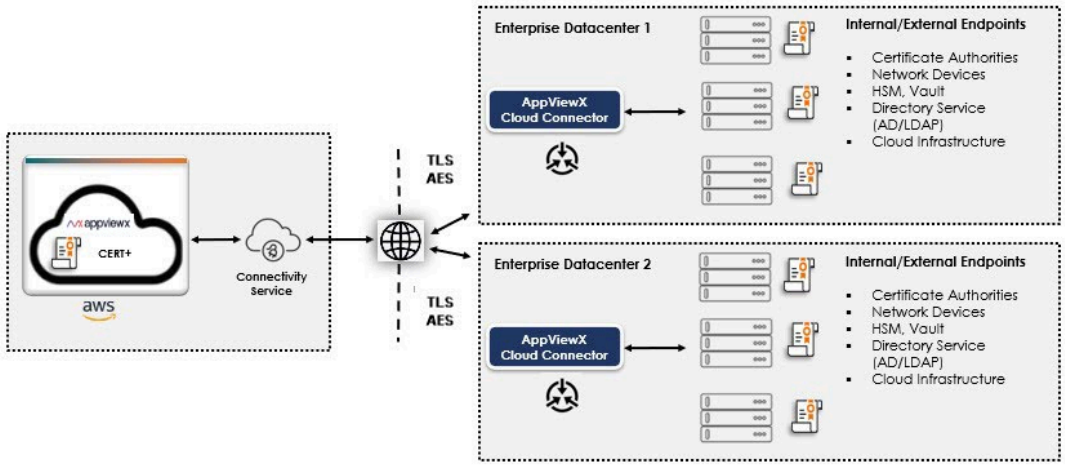
Chapter 2: Features of the AppViewX Cloud Connector

- Data Center-based Routing
- Cloud Connector High Availability
- Custom Certificates for Core Communication

Data Center-based Routing

The cloud connectors that need to connect to the network endpoints are deployed inside a specific data center (DC) in an enterprise's premises. At the time of adding a new AppViewX Cloud Connector instance, the user will be prompted to specify the DC name to which the cloud connector belongs. Based on the DC in which the cloud connector is added, the calls to manage the end points are routed to the specific cloud connector inside a DC.

Figure 1. Typical deployment of the AppViewX Cloud Connector across multiple data centers



AppViewX supports the following two types of data center routing:

- Non strict routing (Default)
- Strict routing

Non strict routing (Default)

In this mode of routing, when a user selects a specific DC when performing an action (like discovery, device addition, cert push etc), the specific action will be routed to the AppViewX Cloud Connector in the selected DC. However, when there are no healthy AppViewX Cloud Connector instances available in the selected DC, the request will be routed to the next available healthy instance in a different DC.

This is a preferred method of deployment when you do not have a restriction in communication across your data centers.

Strict routing

When you want the requests to an endpoint in a DC to be routed only to the AppViewX Cloud Connector instance in the same DC, enable strict routing. This method ensures that when there are no healthy AppViewX Cloud Connectors in the selected DC to perform the action, the request does not get routed to any other available AppViewX Cloud Connector instance in a different DC. This method is most suitable when you are trying to manage devices within restricted DMZ zones and high latency between DCs.

Note: To deploy strict routing, email the AppViewX Technical Support team at help@appviewx.com or call them at +1 (212) 390 1644.

Cloud Connector High Availability

To deploy AppViewX cloud connectors with high availability, it is recommended that you deploy:

- more than one cloud connector across all data centers, in the case of non-strict routing (default)
- more than one cloud connector per datacenter, in the case of strict routing

Custom Certificates for Core Communication

By default, you can provision existing AppViewX self-signed certificates for the communication between the AppViewX Cloud Connector and the AppViewX CLMaaS. In addition to this, you can also push your custom certificates created using external CAs. This is explained in detail in the adding a new AppViewX Cloud Connector instance section.

Chapter 3: Setting up the AppViewX Cloud Connector

- Step 1: Checking Prerequisites
- Step 2: Accessing the Setup Interface
- Step 3: Installing the AppViewX Cloud Connector

Step 1: Checking Prerequisites

The following sections list the minimum prerequisites for setting up and operating the AppViewX Cloud Connector, as well as the steps for checking if the prerequisite requirements have been fulfilled.

- Hardware
- Operating System
- Docker Prerequisites
- Server and Network Prerequisites
- Executing the Prerequisite Check Script

Hardware

Each AppViewX Cloud Connector instance requires the following minimum configuration:

- 4vCPU
- 8 GB memory
- 16 GB disk space
- x86 64 bit architecture

Operating System

- Ubuntu version 20.04
- CentOS version 7.7 and 7.9

Docker Prerequisites

- Docker version 20.10.5 or above installed with non-sudo access with basic read and write permissions

> **Note:** Support for rootless Docker is excluded.

For Docker installation instructions, refer to the links below:

- For installing the Docker Engine: https://docs.docker.com/engine/install/
- For post-installation steps for Linux: https://docs.docker.com/engine/install/linux-postinstall/

- Bash shell support in the node for the installation of the AppViewX Cloud Connector Connectivity Service

Server and Network Prerequisites

- Use dedicated machines for hosting the Cloud Connector and do not install any other components on these machines.
- Ensure the node on which the AppViewX Cloud Connector is installed has access to the enterprise's internal network devices.
- On the node on which the AppViewX Cloud Connector is installed, ensure that the node's clock is synchronized with the network time using NTP or PTP.

  To do this, execute the following sequence of commands:

```
yum install -y ntp

systemctl enable ntp

systemctl start ntp
```

- Ensure that the AppViewX Cloud Connector can establish connectivity with the AppViewX CLMaaS server endpoints over HTTPS (port 443).

> **Note:** In the instance a proxy being used, the proxy has to be configured as a pass-through.

> **Note:** The Cloud Connector URL to be whitelisted for connectivity can be obtained from the Cloud Connector Settings Page of your CLMaaS account. Example of the AppViewX Cloud Connector URL: `https://<example-tenant>-cc.appvx.com:443/`

> **Tip:** : To verify connectivity with the AppViewX SaaS servers, use the **cURL** utility as given below. When connectivity has been established successfully, the command will return the HTTP code **400**.

curl -kv **<<https://AppViewX CLMaaS server URL>>**/ 2>&1 | grep 400

Note: To install the curl utility on Ubuntu, use the command given below:

`apt-get install curl`

Note: To install the curl utility on CentOS, use the command given below:

`yum install curl`

- Disable the firewalld in the tenant's node (**Ubuntu**) where the AppViewX Cloud Connector is to be installed.

  To check the current status of firewalld, execute the command given below: `sudo ufw status`

  To permanently disable firewalld, execute the command given below: `sudo ufw disable`

- Disable the firewalld in the tenant's node (**CentOS** and **RedHat**) where the AppViewX Cloud Connector is to be installed.

  To check the current status of firewalld, execute the command given below: `sudo systemctl status firewalld --now`

  To permanently disable the firewalld, execute the command given below: `sudo systemctl disable firewalld --now`

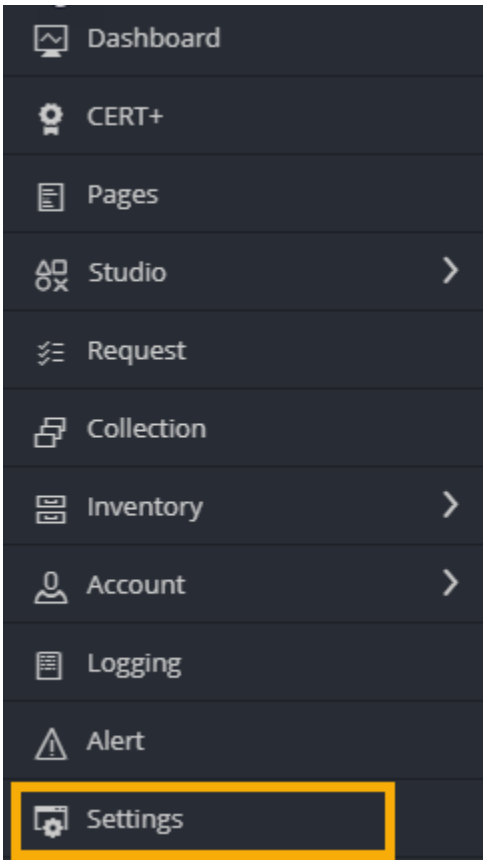
  To restrict other devices from enabling the firewalld, execute the command given below: `sudo systemctl mask firewalld --now`

Executing the Prerequisite Check Script

To simplify compliance to the AppViewX Cloud Connector installation prerequisites, you can execute a script to identify and rule out any deviations from the requirements.

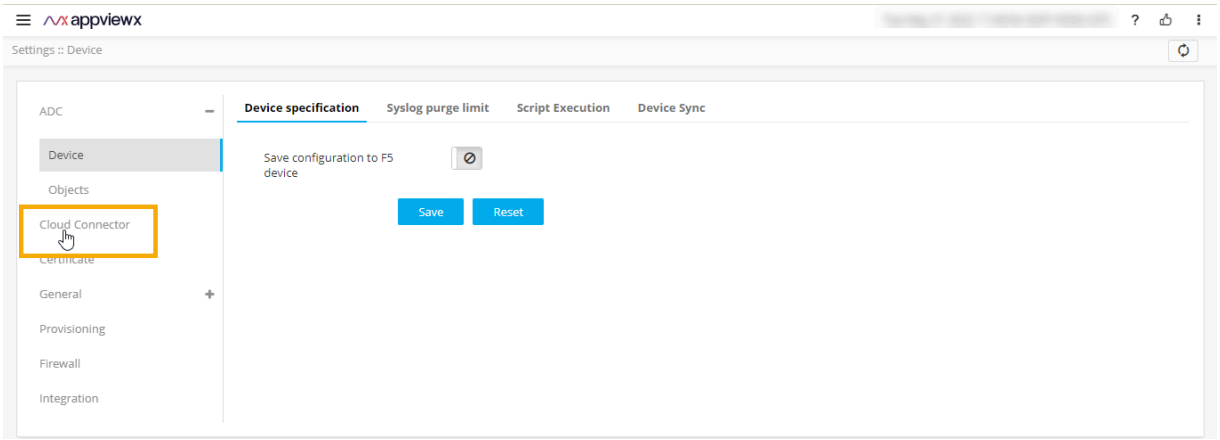
To perform a prerequisite check:

1. After successfully logging in to the AppViewX CLMaaS GUI, from the top left corner of the screen, click ☰.
2. From the menu displayed, click **Settings**.



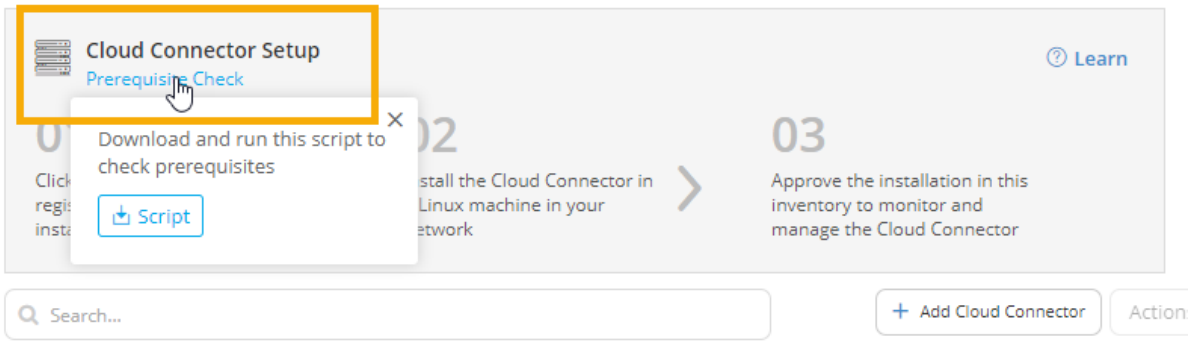
By default, the **Settings :: Device** page is displayed.

3. From the left navigation pane, select **AppViewX Cloud Connector**.

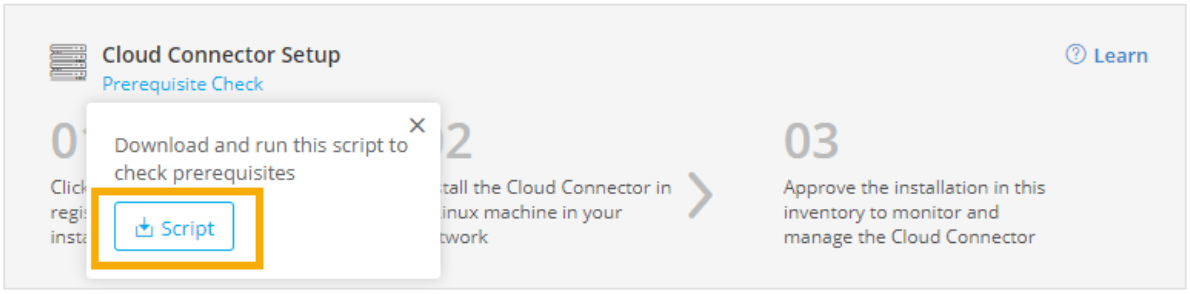


The **Settings :: AppViewX Cloud Connector** page is displayed.

4. On the **Settings :: AppViewX Cloud Connector** page, from the **AppViewX Cloud Connector Setup** banner, place the mouse pointer over **Prerequisite Check**.



5. From the dialog box displayed, click **Script**.

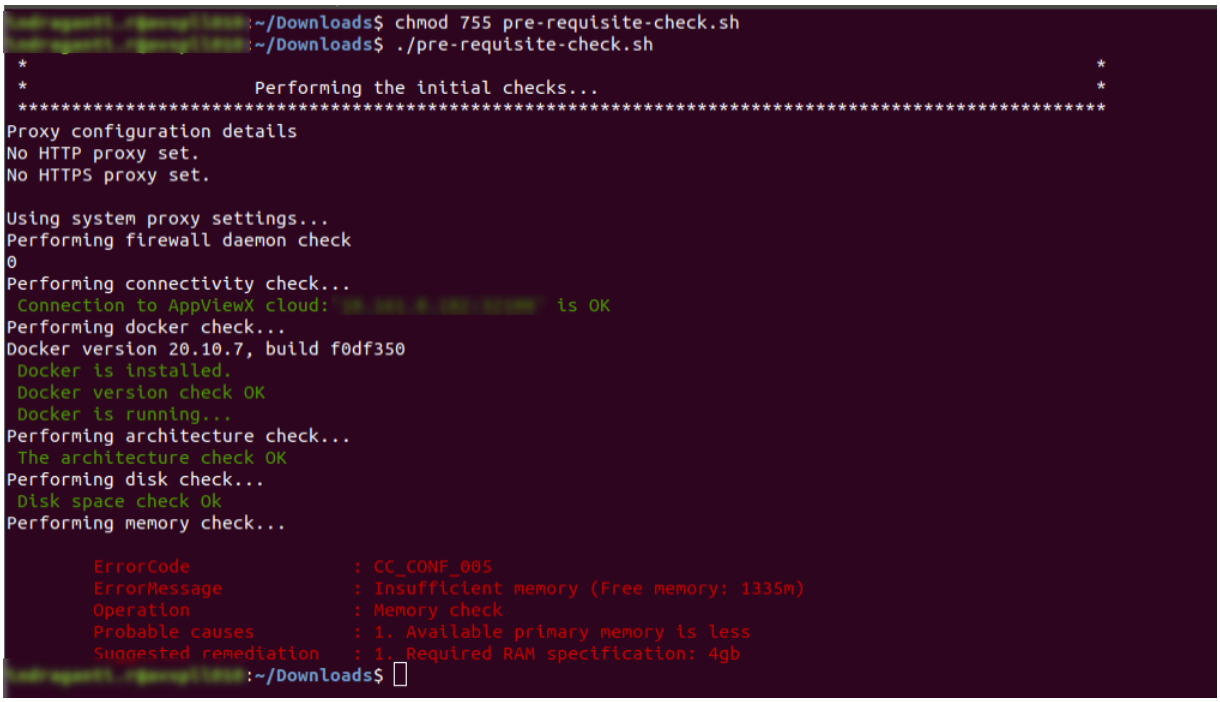


   The **pre-requisite-check.sh** script file is downloaded.

6. From the command line interface, navigate to the node where the AppViewX Cloud Connector will be installed.
7. Convert the downloaded script file into an executable file using the chmod command, as shown below:`chmod 755 pre-requisite-check.sh`
8. Execute the **.sh** prerequisite check script file.

   If the node does not meet the prerequisites for the AppViewX Cloud Connector installation, the output of the command returns an error code and the corresponding error message, causes, and fixes, if any.

   For example, as seen in the sample output in the image below, the prerequisite check for the memory requirement has failed.



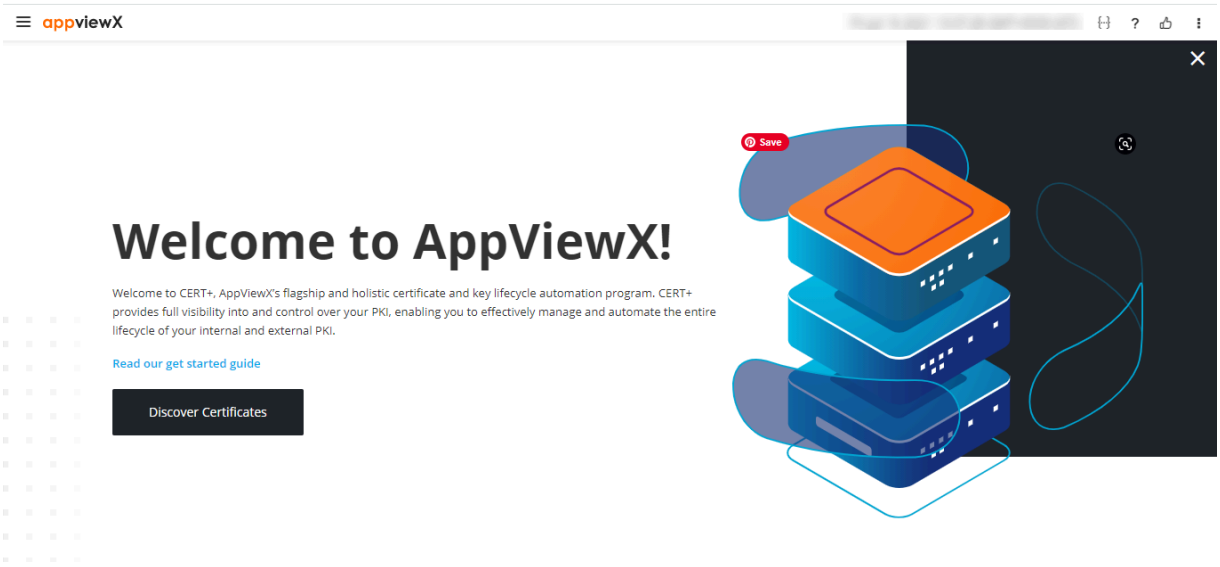
Note: For resolutions to the prerequisite check failure scenarios, click here.

Step 2: Accessing the Setup Interface

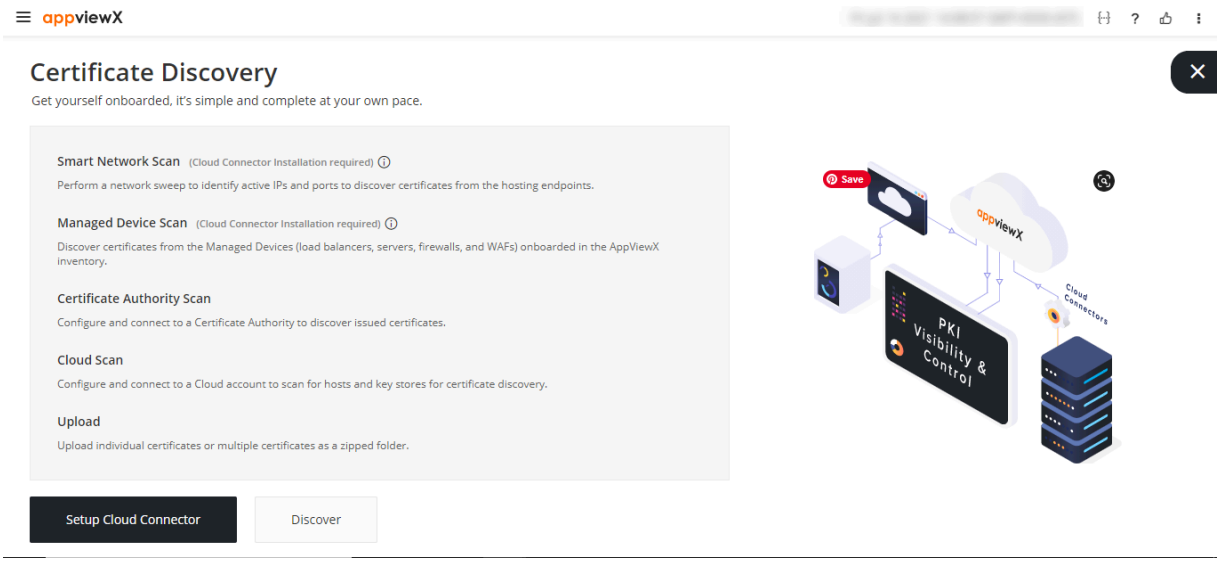
In order to set up the AppViewX Cloud Connector instance, you will need to login to the connectivity service's user interface. The following steps will outline the navigation and steps required to access the AppViewX Cloud Connector's setup interface.

Important: As an additional layer of security, AppViewX issues client certificates to access the AppViewX GUI. The client certificate will be made available as part of the onboarding process. Upload this client certificate to the browser to start accessing the product.

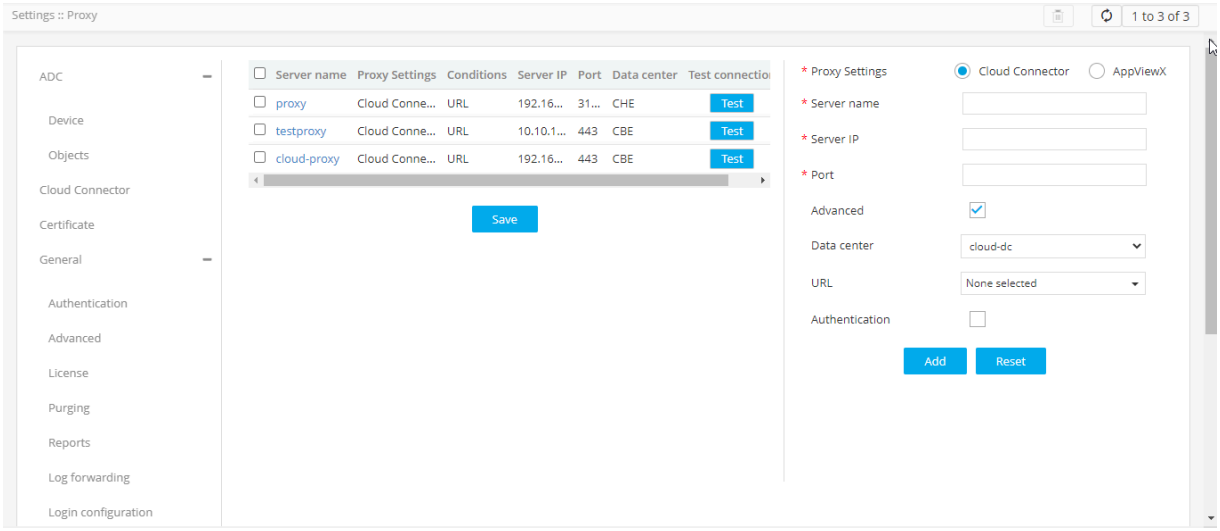
1. Enter your SaaS account URL (for example, **https://tenant-name.appvx.com/appviewx/login**) in the address bar of your browser.
2. From the landing page, click **Discover Certificates**.



3. From the **Certificate Discovery** page, click **Setup AppViewX Cloud Connector**.



The **Settings :: Proxy** page is displayed.



Step 3: Installing the AppViewX Cloud Connector

The AppViewX Cloud Connector authenticates and encrypts all communication between the AppViewX cloud and the enterprise network resources. Once installed, the AppViewX Cloud Connector establishes communication with the AppViewX cloud through an outbound connection. The process of deploying the AppViewX Cloud Connector involves the steps listed below. The substeps for each of these steps are outlined in the subsequent sections.

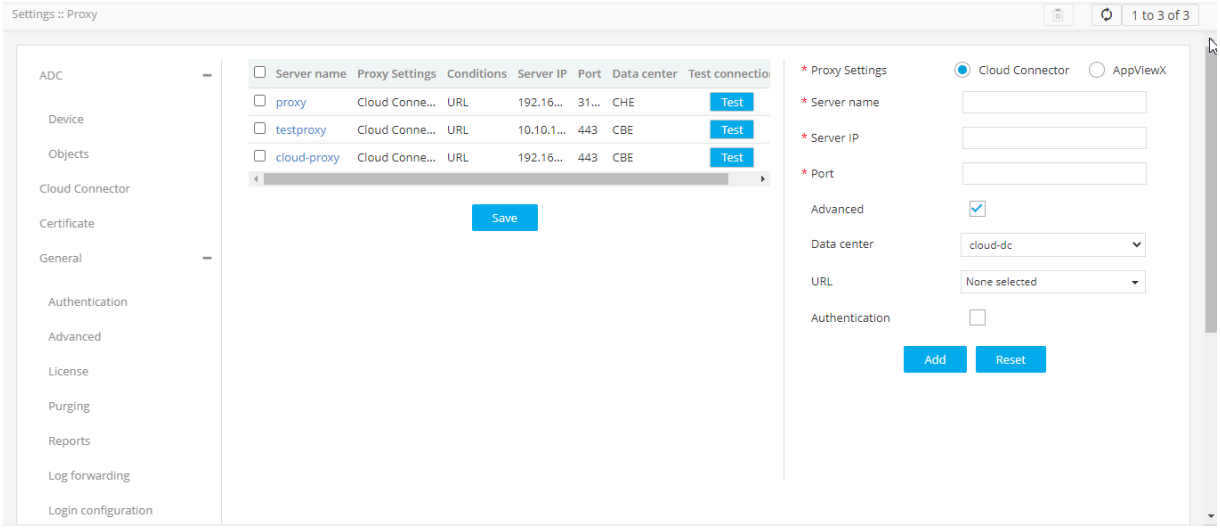
- Configuring Network Proxy Settings
- Downloading the Installer
- Installing the AppViewX Cloud Connector Agent
- Reviewing the Installation

Configuring Network Proxy Settings

If the Cloud Connector deployed on the machine requires communication to external or internal endpoints (Certificate Authority, SSL / TLS endpoints) via proxy, the follow the steps given here:

Important: This is an optional step.

1. To configure a proxy server, on the **Settings :: Proxy** page, enter the following details:



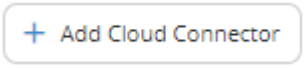
Field	Description
Proxy Settings*	• If you want to configure proxy settings for the AppViewX Cloud Connector, select **Cloud Connector**. • If you want to configure proxy settings for the AppViewX product holistically, select **AppViewX**.
Server name*	Name of the proxy server
Server IP*	IP address of the proxy server
Port*	Port number of the proxy server
Advanced	To enable advanced settings, select this checkbox.
Data center*	**Note:** This field is displayed only when the **Advanced** checkbox is selected. From the dropdown menu, select a data center.
URL	**Note:** This field is displayed only when the **Advanced** checkbox is selected. From the dropdown menu, select the URL that should be connected to via the proxy.

Field	Description
Authentication	To enable authentication for accessing the proxy, select this checkbox.
Username*	**Note:** This field is displayed only when the **Authentication** checkbox is selected. Enter the username configured for the proxy.
Password*	**Note:** This field is displayed only when the **Authentication** checkbox is selected. Enter the password required to access the proxy.

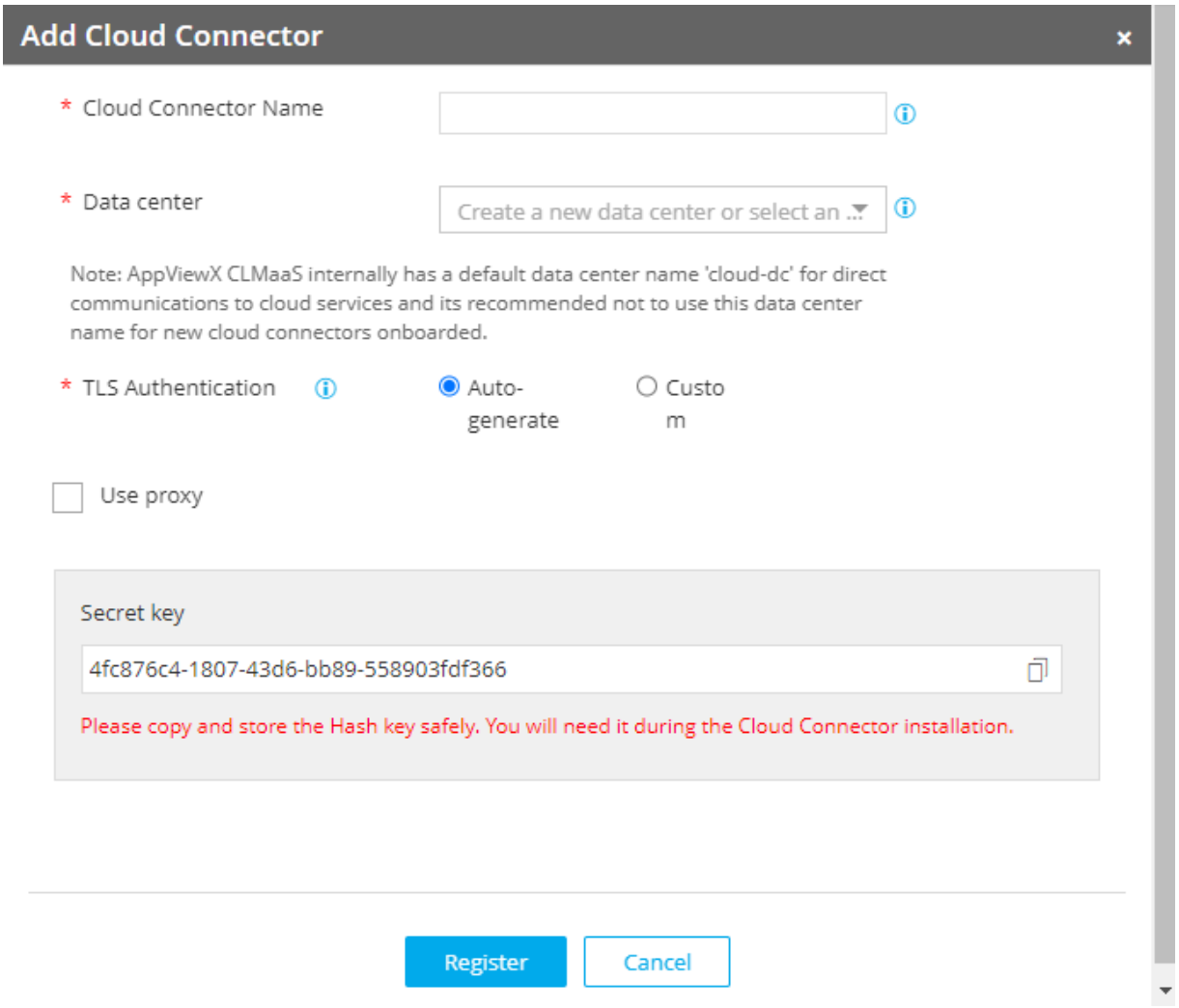
2. Click **Add**.
   Details of the added proxy server are displayed in the table on the left.

Downloading the Installer

To set up an instance of the AppViewX Cloud Connector, you are required to create an installer (for each AppViewX Cloud Connector you want to set up). The steps below outline how you can register and create a downloadable AppViewX Cloud Connector installer package.

1. From the **Settings :: Cloud Connector** page, click 

.
   The **Add Cloud Connector** action pane is displayed.



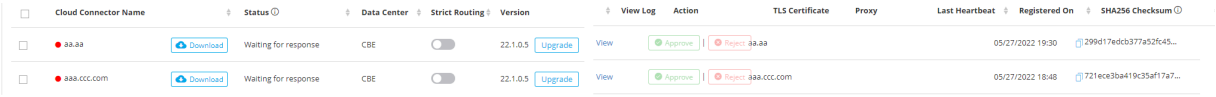
2. In the **Add Cloud Connector** action pane, enter the following details (sample values are shown in an image below the table):

Field	Description
Cloud Connector Name*	FQDN of the machine where the AppViewX Cloud Connector is to be installed
Data center*	Name of the data center where the AppViewX Cloud Connector is to be installed **Note:** The AppViewX CLMaaS, internally, has a default data center named **cloud-dc**, for direct communications to cloud services. It is recommended

Field	Description
	that you do not this data center name for new the AppViewX Cloud Connectors onboarded.
TLS Authentication	• To auto-generate a TLS certificate, select **Auto-generate** (default selection). Automatically, the certificate is generated using the AppViewX CA. **Note:** The created certificate is available in the certificate inventory. You can: • Assign this certificate to a certificate group • Configure a certificate expiry alert for this certificate group from the **Server Certificate** dashboard, using the **Certificate Summary Report** widget settings • To enter details of a custom TLS certificate, select **Custom**. The **TLS Certificate Password** and **Custom TLS Certificate** fields are displayed. The instructions for filling these fields are given below.
TLS Certificate Password*	**Note:** This field is displayed only if you have selected to enter details of a Custom TLS certificate in the **TLS Authentication** field. Password of the TLS certificate (that will be uploaded in the next step) **Note:** This is a mandatory field if a Custom TLS certificate is uploaded. AppViewX supports only password-protected Custom TLS certificates.
Custom TLS Certificate	**Note:** This field is displayed only if you have selected to enter details of a Custom TLS certificate in the **TLS Authentication** field. To upload a custom TLS certificate: a. To navigate to the location of the custom TLS certificate, click within the field. b. Select the certificate file. c. Click **Open**. d. To upload the custom TLS certificate selected, click **Upload**.

Field	Description
	Note: AppViewX supports only password-protected Custom TLS Certificates.
Use proxy	A proxy server is required if the AppViewX Cloud Connector is unable to connect to your endpoints available in the internet. To use a proxy server for the deployment: a. Select the **Use proxy** checkbox. b. To select a preconfigured proxy (for the selected data center), from the **Select Proxy** dropdown list, select a proxy server. **OR** a. To create a new proxy server setting: i. Use the Click here option shown below the **Select Proxy** dropdown list. ii. For steps to create a new proxy server setting, click here.
Secret key	A unique key for the AppViewX Cloud Connector installation Click to copy this key and save it in a safe place. **Note:** Ensure that the secret key is copied and stored safely before you click **Register**. If you don't, the key is lost forever. For security reasons, the key is not stored within the product.

3. To register the above AppViewX Cloud Connector configuration, click **Register**.
   Details of this AppViewX Cloud Connector are added in the inventory details table, which is explained in detail here.



   The AppViewX Cloud Connector's health is also analyzed and, accordingly, the health indicator is displayed before the **Cloud Connector Name**.

4. To download the installation package, for the AppViewX Cloud Connector, click Download.

5. In the **Confirmation Message** dialog box, click **Yes**.
   The download progress is shown using a progress bar.

Installing the AppViewX Cloud Connector Agent

1. Ensure that all the above mentioned prerequisites are fulfilled by the host machine.
2. Copy the installer from your local system to the AppViewX Cloud Connector host machine (using a SFTP server) as an appviewx user or the user that will be used to install the cloud connector
3. To extract the installer, from the downloaded package, extract the tar.gz file using the command given below: `tar -zxvf <filename>.tar.gz`
   For example: `tar - zxvf pesrv07-test-94-99-appviewx-appviewx-net-cloud-connector.tar.gz`
4. On the node where the AppViewX Cloud Connector agent will be installed, from the extracted installation package, run the **./install.sh** script.
   The script will run the prerequisites check once again.
5. On successful verification of the prerequisites, you will be prompted to enter the **Secret Key** (rendered during the Downloading the Installer process).
   On entering the Secret Key, the installation will proceed. Installation logs, according to the outcome of the installation, are displayed. A sample installation log is shown below.

```
Loaded image: rancher/k3s:v1.23.3-k3s1
Loaded image: rancher/k3d-tools:5.2.2
Loaded image: rancher/mirrored-pause:3.6
[36mINFO[0m[0000] [SimpleConfig] Hostnetwork selected - disabling injection of docker host into the cluster, server load balancer and setting the api port to
 the k3s default
[33mWARN[0m[0000] No node filter specified
[33mWARN[0m[0000] No node filter specified
[33mWARN[0m[0000] No node filter specified
[36mINFO[0m[0000] Prep: Network
[36mINFO[0m[0000] Re-using existing network 'host' (8bebb4ae61001f74487d0aa6b315396405d0127c938da1206614d113295ae139)
[36mINFO[0m[0000] Created volume 'k3d-cc-images'
[36mINFO[0m[0000] Starting new tools node...
[36mINFO[0m[0000] Starting Node 'k3d-cc-tools'
[36mINFO[0m[0001] Creating node 'k3d-cc-server-0'
[36mINFO[0m[0001] Using the k3d-tools node to gather environment information
[36mINFO[0m[0001] Starting cluster 'cc'
[36mINFO[0m[0001] Starting servers...
[36mINFO[0m[0001] Starting Node 'k3d-cc-server-0'
```

```
[36mINFO[0m[0033] All agents already running.
[36mINFO[0m[0033] All helpers already running.
[36mINFO[0m[0033] Cluster 'cc' created successfully!
[36mINFO[0m[0034] You can now use it like this:
kubectl cluster-info
Cluster setup is completed. Will start the deployment shortly...
Importing the required images...
[36mINFO[0m[0000] Importing image(s) into cluster 'cc'
[36mINFO[0m[0000] Importing images from 1 tarball(s)...
[36mINFO[0m[0000] Importing images '[/home/appviewx/CCTEST/deps/tools/mid-server-docker-image/avx-mid-server-base-22.1.0.0.tar]' into node
 'k3d-cc-server-0'...
[36mINFO[0m[0024] Successfully imported image(s)
[36mINFO[0m[0024] Successfully imported 1 image(s) into 1 cluster(s)
Import in progress...
[36mINFO[0m[0000] Importing image(s) into cluster 'cc'
[36mINFO[0m[0000] Importing images from 1 tarball(s)...
[36mINFO[0m[0000] Importing images '[/home/appviewx/CCTEST/deps/tools/mid-server-docker-image/k3d-tools-5.2.2.tar]' into node 'k3d-cc-server-0'...
[36mINFO[0m[0005] Successfully imported image(s)
[36mINFO[0m[0005] Successfully imported 1 image(s) into 1 cluster(s)
Import in progress...
[36mINFO[0m[0000] Importing image(s) into cluster 'cc'
[36mINFO[0m[0000] Importing images from 1 tarball(s)...
[36mINFO[0m[0000] Importing images '[/home/appviewx/CCTEST/deps/tools/mid-server-docker-image/rancher-mirrored-coredns-coredns-1.8.6.tar]' into
 node 'k3d-cc-server-0'...
[36mINFO[0m[0007] Successfully imported image(s)
[36mINFO[0m[0007] Successfully imported 1 image(s) into 1 cluster(s)
[36mINFO[0m[0000] Importing image(s) into cluster 'cc'
[36mINFO[0m[0000] Importing images from 1 tarball(s)...
[36mINFO[0m[0000] Importing images '[/home/appviewx/CCTEST/deps/tools/mid-server-docker-image/rancher-local-path-provisioner-v0.0.21.tar]' into node
 'k3d-cc-server-0'...
[36mINFO[0m[0004] Successfully imported image(s)
[36mINFO[0m[0004] Successfully imported 1 image(s) into 1 cluster(s)
[36mINFO[0m[0000] Importing image(s) into cluster 'cc'
[36mINFO[0m[0000] Importing images from 1 tarball(s)...
[36mINFO[0m[0000] Importing images '[/home/appviewx/CCTEST/deps/tools/mid-server-docker-image/rancher-mirrored-pause-3.6.tar]' into node
 'k3d-cc-server-0'...
[36mINFO[0m[0003] Successfully imported image(s)
```

```
[36mINFO[0m[0003] Successfully imported 1 image(s) into 1 cluster(s)
Deploying the Cloud Connector...
NAME: avx-mid-server-starter
LAST DEPLOYED: Mon May 30 15:51:13 2022
NAMESPACE: cc
STATUS: deployed
REVISION: 1
NOTES:
1. It may take a couple of minutes for the Cloud Connector to be up.
  kubectl get pod --namespace cc
**************************************************************************
*    Congratulations!!! The installation completed successfully.   *
*    Please wait till the Cloud Connector is up and running.      *
**************************************************************************
(1%) Cloud Connector status: Running
[32m Cloud Connector is up and running. (B[m
```

Troubleshooting: For installation errors, refer to the Troubleshooting section.

The AppViewX Cloud Connector consists of two important components—the starter plugin and the platform. The starter plugin component is installed along with the AppViewX Cloud Connector, in the same installation process.

When installed, the starter plugin is used to initiate the download of the platform component. The platform component is used to host business use cases related to the AppViewX Cloud Connector.

When the platform component download is in progress, it is indicated by the symbol prefixed to the platform component version number in the AppViewX Cloud Connector inventory details 

. A completed download/upgrade is indicated by the symbol in the same location 

.

Note: Based on the internet bandwidth and the number of cloud connectors being installed, the downloading of the cloud connector may vary between 5 to 15 minutes.

Note: To enable auto enrollment for the AppViewX Cloud Connector being installed, refer to the **Certificate Auto Enrollment Protocols Guide**.

Note: (For the AppViewX Support) As mentioned already, this version supports the CLMaaS deployment only for CERT+. Therefore, the complete parsing of F5 needs to be disabled, so that only CERT+ (and not ADC) is enabled for the tenant. To do this, after installing the Cloud Connector agent, execute the following query to enable the SKIP_ADC_PARSING property: `db.getCollection('avx_app_metadata').update({ "_id" : "SKIP_ADC_PARSING"},{$set: { "value" : "true"}});`

Reviewing the Installation

After the agent is successfully installed on the Linux machine, the agent can approve or reject the installation.

Note: The **Approve** and **Reject** buttons are displayed only after the AppViewX Cloud Connector agent has been downloaded.

To approve/reject the installation:

From the Action field, click 

.

If the installation has been approved, the AppViewX Cloud Connector is moved to the **Running** state. If the AppViewX Cloud Connector has been **Rejected**, the details of the AppViewX Cloud Connector are removed from the inventory.

Troubleshooting: If the AppViewX Cloud Connector instance has been approved but is not moved to the **Running** state, you can check the pod status and/or restart the pod(s), as required.

Chapter 4: Managing F5 BIG IP and A10 Devices

- Managing F5 BIG-IP and A10 Devices

Managing F5 BIG-IP and A10 Devices

This section lists the prerequisites for managing certificates on the F5 BIG IP and A10 devices using the AppViewX Cloud Connector.

- F5 BIG-IP Devices
- A10 Devices

F5 BIG-IP Devices

To manage certificates on F5 BIG-IP devices, follow the steps below:

1. Ensure you have a Licensed version of the iControl jar for F5 BIG-IP devices (Refer: https://devcentral.f5.com/s/articles/iControl-Library-For-Java-With-Source)
2. In the AppViewX Cloud Connector installation package, copy the iControl jar to the folder /deps/external_libs.

Note: Check with the Customer Support team that the iControl jar has been uploaded to the AWS instance. Without these two upload operations, F5 functionalities will fail with the following error message: **The pre-requisite library required for managing the F5 vendor is not configured. Please contact the system admin for more details.**

A10 Devices

To manage certificates on A10 devices, in **Authentication Settings**, set the **Node Password** to the password of the node where the AppViewX Cloud Connector instance is deployed.

For instructions on Configuring the Authentication Settings, refer to the **CLMaaS Platform User Guide**.

Chapter 5: Troubleshooting the AppViewX Cloud Connector

- Troubleshooting the AppViewX Cloud Connector

Troubleshooting the AppViewX Cloud Connector

This section includes steps for troubleshooting the AppViewX Cloud Connector at various points in its operation.

- Managing Certificates on F5 BIG-IP Devices
- AppViewX Cloud Connector Health
- Connectivity Checks
- Installation Errors
- Log Analysis
- Checking Pod Status
- Restarting Pods

Managing Certificates on F5 BIG-IP Devices

- In the event of not being able to manage certificates on an F5 BIG-IP device, ensure that the iControl jar is copied and restart the necessary services using the command given below:

```
./deps/tools/k3s kubectl rollout restart deployment avx-mid-server-platform -n cc
```

AppViewX Cloud Connector Health

Amber/red health indicator

- Check if the AppViewX Cloud Connector is up and running.
  - If no, check if there is a connectivity issue due to:
    - Firewall policies
    - Network configuration changes at the tenant's and/or AppViewX's end
  - If the AppViewX Cloud Connector is up and running, check the health indicators to determine if the traffic to the AppViewX Cloud Connector is configured correctly.

- If the health indicator is amber/red:
  - Check if the AppViewX Cloud Connector is up and running.
  - If yes, validate the connectivity from the AppViewX Cloud Connector node to the AppViewX CLMaaS.

Connectivity Checks

- Scenario 1: At the time of installation
  - Check if the AppViewX Cloud Connector is able to reach the AppViewX cloud.
- Scenario 2: After the package has been successfully installed
  - Check the AppViewX Cloud Connector's health indicator.
  - If the health indicator is amber/red:
    - Check if the AppViewX Cloud Connector is up and running in the network .
    - If yes, validate the connectivity from the AppViewX Cloud Connector node to the AppViewX CLMaaS.

Installation Errors

Prerequisite check failure

At the time of the package installation, check for the following prerequisites:

- Hardware
  - Check if the current hardware configuration is according to the prerequisites.
- Connectivity
  - Check the firewall policies, proxy settings, and network configuration settings. Refer to the firewall and network-related prerequisites.
- OS Version
  - Check if the current system configuration is according to the prerequisites.
- Docker installation
  - Check if the current configuration is according to the prerequisites.

SHA256 checksum failure

Cross check the SHA256 checksum in the AppViewX Cloud Connector inventory with the SHA256 checksum in the installer package.

To view the SHA256 checksum in the installer package, execute the command given below:

```
sha256sum <absolute path of the installer package file>
```

Installation Error Codes

Error Code	Error Message	Resolution
CC_CONF_001	Improper docker version (Docker version currently installed: <current version number>)	Ensure that the installed version of the Docker is 20.10.5 or above.
CC_CONF_002	Incompatible system architecture	Ensure the operating system on the node complies with the following prerequisites: • x86 64 bit
CC_CONF_003	Failed to establish connection to AppViewX cloud	Check the firewall policies and proxy settings in the tenant premises.
CC_CONF_004	Docker is not installed.	• Install Docker with non sudo access. • Required configuration: version 20.10.5 or higher • For instructions for installing the Docker Engine, click here. • For post-installation steps for Linux:, click here. • In the event of a VM reboot, the Docker needs to be restarted. To configure the Docker to restart on boot, follow the instructions given here.
CC_CONF_005	Insufficient memory (Free memory: <available memory>)	Required RAM specification: 8GB
CC_CONF_006	Disk space available is low: <available disk space in MB>	Minimum available disk space required: 16GB

Error Code	Error Message	Resolution
CC_CONF_007	Docker not running or not accessible for non sudoers	1. To check the Docker status, execute **one** of the following commands: • `service docker status` • `systemctl status docker` 2. To start the Docker, execute **one** of the following commands: • `service docker start` • `systemctl start docker` 3. Ensure that the Docker is accessible to non sudoers. • For post-installation steps for Linux: https://docs.docker.com/engine/install/linux-postinstall/ • In the event of a VM reboot, the Docker needs to be restarted. To configure the Docker to restart on boot, follow the instructions given here.
CC_CONF_008	Cluster already exists.	1. Uninstall the AppViewX Cloud Connector. 2. Reinstall the AppViewX Cloud Connector in the same/different node.

CC_CONF_009	firewalld is running	• Execute the following script to open the port in firewalld that requires sudo access:
		`./deps/utils/open-ips-ports-firewalld.sh`
		OR
		• Execute the following commands:
		`sudo firewall-cmd --permanent --add-port=22/tcp`
		`sudo firewall-cmd --permanent --add-source=10.42.0.0/16`
		`sudo firewall-cmd --permanent --add-source=10.43.0.0/16`
		`sudo firewall-cmd --direct --permanent --add-rule ipv4 filter FORWARD 1 -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT`
		`sudo firewall-cmd --permanent --add-forward-port=port=30020:proto=tcp:toport=30020:toaddr=`

Error Code	Error Message	Resolution
		sudo firewall-cmd --permanent --add-forward-port=port=30021:proto=tcp:toport=30021:toaddr= sudo firewall-cmd --permanent --add-forward-port=port=30022:proto=tcp:toport=30022:toaddr= sudo firewall-cmd --reload
CC_CONF_010	Not met cpu requirement:- No of available processors(vCPU): <number>	The required number of processors (vCPU) is 4.
CC_CONF_011	Docker running with Incompatible storage Driver	1. Update the storage driver to **overlay2**. For setup instructions, click here.
CC_CONF_012	A default route is not available in the tenant premises or the tenant is not connected to a network.	• Add a default route with an IP address. • Ensure that the network connection is up.
CC_PLATFORM_001	Failed to untar the upgrade dependencies	For SRE: 1. Retry the upgrade operation once. 2. If the upgrade operation fails, reinstall the Cloud Connector.
CC_PLATFORM_002	Upgrade operation failed	For SRE: Check user logs for cause of operation failure (insufficient disk and/or memory). • If the cause of failure is insufficient disk and/or memory space, direct the tenant to free disk and/or memory. • If not, reinstall the Cloud Connector.

Troubleshooting: If your error remains unresolved even after executing the above troubleshooting steps, email the AppViewX Technical Support team at help@appviewx.com or call them at +1 (212) 390 1644.

Log Analysis

For troubleshooting common error scenarios, the AppViewX Cloud Connector's logs can be analyzed to identify the cause and the solution required, therefore.

Within the installation directory, the AppViewX Cloud Connector logs can be accessed at **./deps/logs/cloud-connector.log**.

AppViewX comes with a set of commands required for troubleshooting based on log analysis.

To troubleshoot based on log analysis, you can use the `kubectl` commands as usual. You can also use the k3s available in the **./deps/tools folder**. Using k3s, you can fire `kubectl` commands in the form k3s kubectl.

Syntax:

```
<kubectl-command-parameters>
```

- To list all the pods and their status, use the following syntax:

  ```
  ./deps/tools/k3s kubectl get pods -A
  ```

- To delete a pod and then restart it, use the following syntax:

  ```
  ./deps/tools/k3s kubectl delete pods <pod-id> -n <name-space>
  ```

- To describe the pods, use the following syntax:

  ```
  ./deps/tools/k3s kubectl describe pods <pod-id> -n <name-space>
  ```

- To check the logs via `kubectl` command for k3s related pods, use the following syntax:

  ```
  ./deps/tools/k3s kubectl logs <pod-id> -n kube-system
  ```

Note: The k3s cluster created by the AppViewX Cloud Connector is called cc.

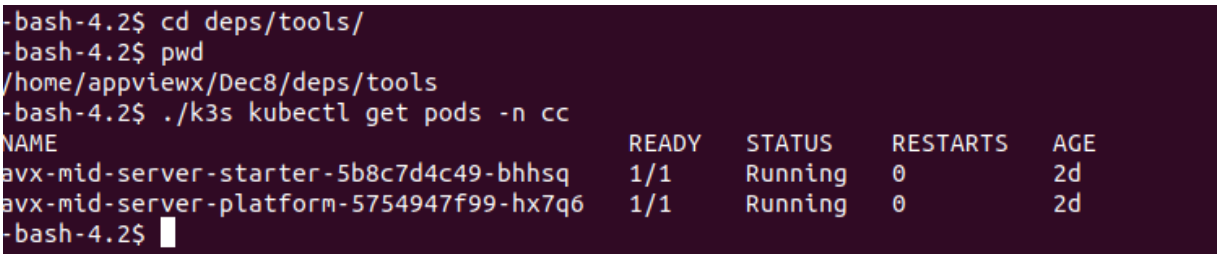
Troubleshooting: If your error remains unresolved even after executing the above troubleshooting steps, email the AppViewX Technical Support team at help@appviewx.com or call them at +1 (212) 390 1644.

Checking Pod Status

To check the pod status:

1. From the terminal, navigate to the **deps/tools** folder.
2. As shown in the image below, execute the following command:

```
./k3s kubectl get pods -n cc
```



The output should return the details of two pods: **avx-mid-server-started** and **avx-mid-server-platform** (described below). The status of both pods should be **Running**.

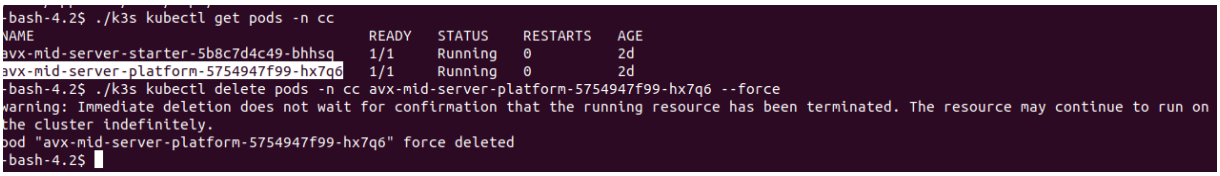
- **avx-mid-server-starter**: This pod is responsible for starting the AppViewX Cloud Connector. It downloads the artifacts required to start the AppViewX Cloud Connector from the AppViewX servers to the nodes. Restarting this pod downloads the artifacts once again and applies the latest upgrades to the AppViewX Cloud Connector from ther server.
- **avx-mid-server-platform**: This pod is responsible for all device communication. It checks the AppViewX SaaS servers for the CLM actions to be performed (for example, **Discovery**, **Device Addition**, **Certificate push**, and so on). The pod executes the required command(s) on the end device and sends the response to the AppViewX servers.

Restarting Pods

Restarting a Specific Pod

1. From the terminal, navigate to the **deps/tools** folder.
2. As shown in the image below, execute the following command:

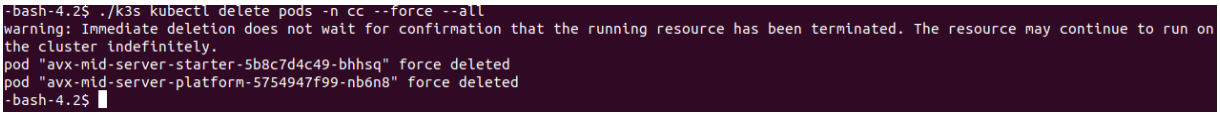
```
./k3s kubectl delete pods -n cc <podname> --force
```



Restarting both (starter and platform) Pods

1. From the terminal, navigate to the **deps/tools** folder.
2. As shown in the image below, execute the following command:

```
./k3s kubectl delete pods -n cc --force --all
```



Chapter 6: Managing the AppViewX Cloud Connector

- Managing the AppViewX Cloud Connector

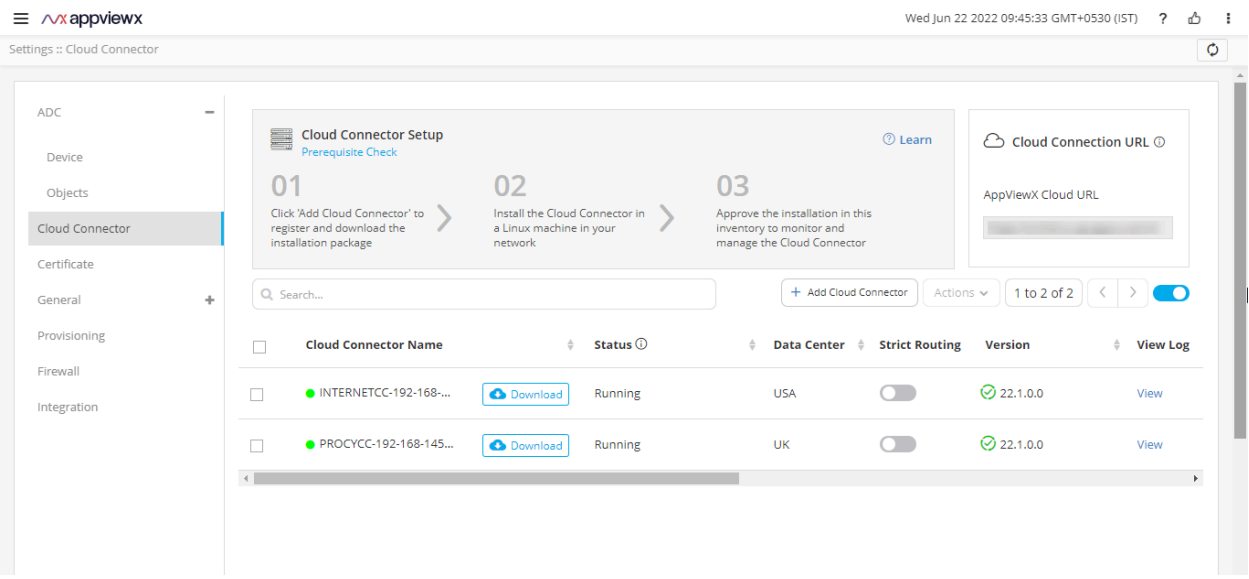
Managing the AppViewX Cloud Connector

To help you work with and manage the AppViewX Cloud Connector, this section introduces you to the Cloud Connector inventory and outlines the steps for performing the various AppViewX Cloud Connector actions, uninstalling the AppViewX Cloud Connector, as well as monitoring its health.

- Understanding the AppViewX Cloud Connector Inventory
- AppViewX Cloud Connector Actions
- Monitoring the Health of the AppViewX Cloud Connector

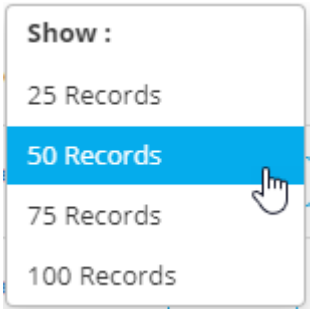
Understanding the AppViewX Cloud Connector Inventory

The AppViewX Cloud Connector inventory details page displays important information pertaining to the AppViewX Cloud Connector. The page provides easy access to functions that let you add a new AppViewX Cloud Connector and perform configuration actions like starting, pausing, and deleting the AppViewX Cloud Connector instance and so on.



The elements and fields that constitute the AppViewX Cloud Connector inventory details are explained in the table below:

Element/Field	Description
AppViewX Cloud Connector Setup banner	The banner summarizes the process of setting up the AppViewX Cloud Connector.
Cloud Connection URL	AppViewX cloud URL of the server (internal) that hosts the AppViewX instance of the AppViewX Cloud Connector
Search	 To search for a AppViewX Cloud Connector entry: 1. In the **Search** field, enter the value you want to filter the records for. 2. Press **Enter**. The **Settings :: Cloud Connector** page is updated to show details of only those records that match the search criteria.
Add AppViewX Cloud Connector	To add a new AppViewX Cloud Connector, click  . **Note:** For detailed steps for adding a new AppViewX Cloud Connector, refer to the Setting Up the AppViewX Cloud Connector section.
Actions	**Note:** This button is enabled only when one or multiple AppViewX Cloud Connectors are selected. AppViewX lets you perform the following actions on a AppViewX Cloud Connector: • Start • Pause • Upgrade

Element/Field	Description
	• Update config • Delete To perform these actions, click Actions . **Note:** For detailed steps to perform each of the above listed actions, refer to the AppViewX Cloud Connector Actions .
Record Count Preference	For easier viewing of records, AppViewX lets you set the record count preference, which is the number of records that will be displayed on one page. To set the record count preference: 1. Click 1 to 9 of 9 . 2. From the **Show** menu displayed, select your record count preference (for example, 50 records).  The **Settings :: Cloud Connector** page is updated according to the record count preference selected. The message, **Record count preference saved successfully**, is displayed. The UI control is also updated to display the current selection.
Auto Refresh	If enabled, the **Auto Refresh** feature automatically refreshes the AppViewX Cloud Connector inventory details every 5 seconds.

<table>
<tr><th>Element/Field</th><th>Description</th></tr>
<tr><td></td><td>To enable this feature, use the 

key.

Tip: Enabling the Auto Refresh feature gives you a real-time status update of the AppViewX Cloud Connector's health, therefore facilitating for timely troubleshooting in the event that it is required.</td></tr>
<tr><td>AppViewX Cloud Connector Name</td><td>This field displays the following two details:

• Name assigned to the AppViewX Cloud Connector when it is added
• Health status of the AppViewX Cloud Connector</td></tr>
<tr><td>Status</td><td>This field has the following values:

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Waiting for response</td><td>After the AppViewX Cloud Connector is registered, the package must be downloaded and installed.

This status indicates that this installation is pending.</td></tr>
<tr><td>Waiting for approval</td><td>After the AppViewX Cloud Connector is installed, the admin must approve/reject the installation by clicking the 

/ 

buttons from the Action field.

This status indicates that the admin's response to the installation is pending.</td></tr>
<tr><td>Running</td><td>The AppViewX Cloud Connector has been approved by the admin and is running.</td></tr>
<tr><td>Paused</td><td>The AppViewX Cloud Connector has been approved by the admin but is paused.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Element/Field</th><th>Description</th></tr>
<tr><td></td><td><table><tr><th>Value</th><th>Description</th></tr></table>
Note: The AppViewX Cloud Connector is paused when it has to undergo maintenance and/or troubleshooting.</td></tr>
<tr><td>Data Center</td><td>Physical location where the AppViewX Cloud Connector system is hosted</td></tr>
<tr><td>Strict Routing</td><td>To enable strict data center-based routing, turn on the toggle under Strict Routing.</td></tr>
<tr><td>Version</td><td>Version of the AppViewX Cloud Connector platform component

Note: If a new version of the AppViewX Cloud Connector platform component is available, the Upgrade button is displayed for that AppViewX Cloud Connector.</td></tr>
<tr><td>View Log</td><td>To view the activity log for a AppViewX Cloud Connector, click View for that AppViewX Cloud Connector.</td></tr>
<tr><td>Action</td><td>This field lets you perform the following actions for a AppViewX Cloud Connector

• Pause a running AppViewX Cloud Connector
• Start a paused AppViewX Cloud Connector
• Approve a AppViewX Cloud Connector
• Reject a AppViewX Cloud Connector

This field displays an action that can be performed for the AppViewX Cloud Connector, depending on the current status of the AppViewX Cloud Connector.
For example, if the connector is running, this field shows the Pause button.</td></tr>
</table>

Element/Field	Description
	Remember: Only the admin user, with the modify permission, can approve/reject a AppViewX Cloud Connector.
TLS Certificate	If a custom TLS certificate has been uploaded at the time of adding the AppViewX Cloud Connector, this field displays the common name and other details (for example, validity) of the custom TLS certificate.
Proxy	Details of the proxy server, if it has been used for the deployment
Last Heartbeat	Timestamp of the latest health analysis of the AppViewX Cloud Connector
Registered On	Timestamp of the AppViewX Cloud Connector installation
SHA256 Checksum	Details of the SHA256 token It ensures that the downloaded AppViewX Cloud Connector package is the same as the checksum displayed in the AppViewX Cloud Connector inventory page.

AppViewX Cloud Connector Actions

Key actions that can be performed include:

- Start
- Pause
- Upgrade
- Updating the Certificate Configuration
- Deleting an AppViewX Cloud Connector Instance
- Uninstalling an AppViewX Cloud Connector Instance

Start

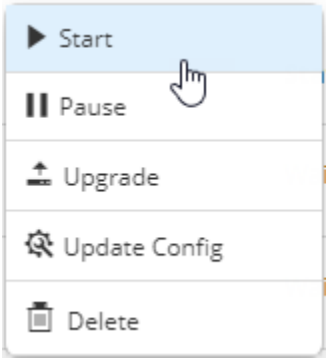
After the admin user has approved its installation, you need to ‘start’ the AppViewX Cloud Connector—you need to enable the AppViewX Cloud Connector to route traffic between the internal network and the AppViewX cloud. There are two ways you can start the AppViewX Cloud Connector.

• Starting the AppViewX Cloud Connector

Starting the AppViewX Cloud Connector

To start a AppViewX Cloud Connector after it has been approved or paused:

1. Navigate to the AppViewX Cloud Connector inventory.
2. Select the checkbox for the AppViewX Cloud Connector you want to start.
3. Click Actions.
4. From the menu displayed, select **Start**.



The AppViewX Cloud Connector Status is set to **Running**.

Pause

The AppViewX Cloud Connector can be paused for regular maintenance or troubleshooting. Pausing the AppViewX Cloud Connector will pause all activities that have to be performed in the network premises—for example, discovering and scanning certificates, accessing endpoints within the network, and so on. There are two ways you can pause the AppViewX Cloud Connector.

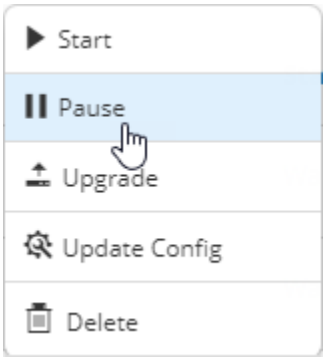
• Pausing the AppViewX Cloud Connector

Pausing the AppViewX Cloud Connector

To pause the AppViewX Cloud Connector:

1. Navigate to the AppViewX Cloud Connector inventory.
2. Select the checkbox for the AppViewX Cloud Connector you want to pause.

3. Click Actions.
4. From the menu displayed, select **Pause**.



The AppViewX Cloud Connector Status is set to **Paused**.

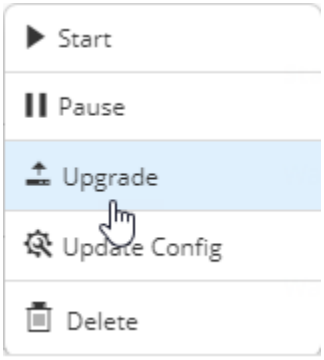
Upgrade

AppViewX provides a seamless CI/CD pipeline to capture the AppViewX Cloud Connector versioning and upgrades on the release portal. If a new version of the AppViewX Cloud Connector component is available, the Upgrade button is displayed for that AppViewX Cloud Connector. If a version upgrade is available, there are two ways in which you can upgrade the AppViewX Cloud Connector.

- Upgrading the AppViewX Cloud Connector Version

Upgrading the AppViewX Cloud Connector Version

1. Navigate to the AppViewX Cloud Connector inventory.
2. Select the checkbox for the AppViewX Cloud Connector you want to upgrade.
3. Click Actions.
4. From the menu displayed, select **Upgrade**.



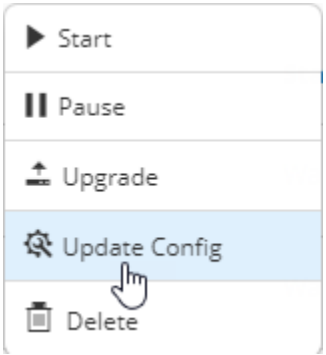
The AppViewX Cloud Connector version is upgraded, as indicated by the symbol prefixed to the version number.

Note: Based on the internet bandwidth and the number of cloud connectors being installed, the downloading of the cloud connector may vary between 5 to 15 minutes.

Updating the Certificate Configuration

If a new certificate has been pushed to the AppViewX Cloud Connector, you scan the selected AppViewX Cloud Connectors to display the updated details in the AppViewX Cloud Connector inventory. To update the certificate configuration for a AppViewX Cloud Connector:

1. Select the checkbox for the required AppViewX Cloud Connector.
2. Click Actions.
3. From the menu displayed, select **Update Config**.



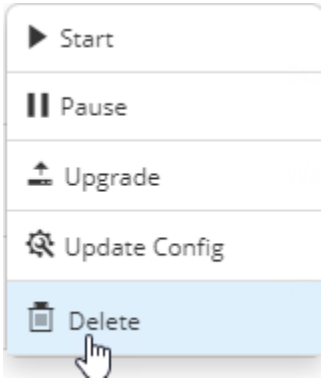
On successful update of the certificate configuration, the message **Update config triggered successfully** is displayed.

Deleting an AppViewX Cloud Connector Instance

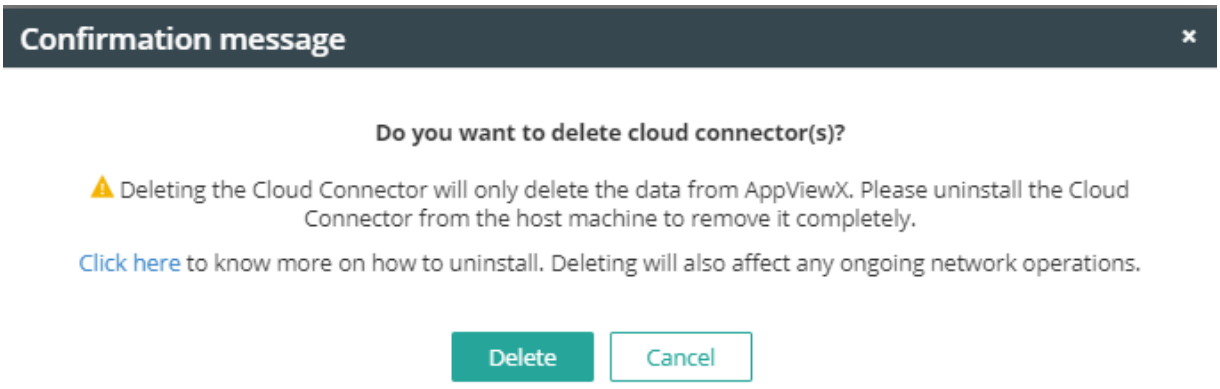
The AppViewX Cloud Connector instance may have to be deleted in events like if there is a fault with the system on which the instance is installed or if it is a faulty installation.

> **Warning:** Deleting a AppViewX Cloud Connector instance without having a backup node will result in traffic blockage.

1. Select the checkbox for the AppViewX Cloud Connector you want to delete.
2. Click Actions.
3. From the menu displayed, select **Delete**.



4. In the **Confirmation message** dialog box, click **Delete**.



The selected AppViewX Cloud Connector is deleted.

> **Attention:** As mentioned in the image above, deleting the AppViewX Cloud Connector will only delete the data from AppViewX. To remove it from the host machine, you will have to uninstall the AppViewX Cloud Connector.

Uninstalling an AppViewX Cloud Connector Instance

To uninstall a AppViewX Cloud Connector instance:

On the node where the AppViewX Cloud Connector agent is installed, run the **uninstall.sh** script (located in the AppViewX Cloud Connector agent's download package).
The corresponding AppViewX Cloud Connector instance is deleted.

Monitoring the Health of the AppViewX Cloud Connector

As a precautionary measure, to ensure in-time troubleshooting in the event of a failure, AppViewX enables runtime health analysis of the AppViewX Cloud Connector Connectivity Service. Accordingly, a color-coded health indicator is displayed for each AppViewX Cloud Connector.

Color of the health indicator	Description
🟢	The AppViewX Cloud Connector is working as expected.
🟠	Although there are no current problems with routing traffic to and from the AppViewX Cloud Connector, the AppViewX Cloud Connector's health needs to be checked. To resolve, refer to the Troubleshooting section.
🔴	The AppViewX Cloud Connector is not receiving traffic. The AppViewX Cloud Connector's health is analyzed for 3 to 5 minutes before it is declared to be down. To resolve, refer to the Troubleshooting section.

For details on how the health indicators are displayed, refer to the Understanding the AppViewX Cloud Connector Inventory page.