



# AppViewX Cloud Connector User Guide

---

Version: 2022.1.0

# Copyright AppViewX, Inc.

## **Copyright © 2022 AppViewX, Inc. All Rights Reserved.**

This document may not be copied, disclosed, transferred, or modified without the prior written consent of AppViewX, Inc. While all content is believed to be correct at the time of publication, it is provided as general-purpose information. The content is subject to change without notice and is provided “as is” and with no expressed or implied warranties whatsoever, including, but not limited to, a warranty for accuracy made by AppViewX. The software described in this document is provided under written license only, contains valuable trade secrets and proprietary information, and is protected by the copyright laws of the United States and other countries. Unauthorized use of software or its documentation can result in civil damages and criminal prosecution.

## **Trademarks**

The trademarks, logos, and service marks displayed in this manual are the property of AppViewX or other third parties. Users are not permitted to use these marks without the prior written consent of AppViewX or such third party which may own the mark.

## **External Reference Links**

This product includes software developed by the CentOS Project ([www.centos.org](http://www.centos.org)).

This product includes software developed by Red Hat, Inc. ([www.redhat.com](http://www.redhat.com)).

This product includes software developed by VMware, Inc. ([www.vmware.com](http://www.vmware.com)).

All other trademarks mentioned in this document are the property of their respective owners.

## **Contact Information**

AppViewX, Inc.

222 Broadway, FL 19

New York, NY 10038

Email: [info@appviewx.com](mailto:info@appviewx.com)

Web: [www.appviewx.com](http://www.appviewx.com)

# Contents

Preface.....	6
Revision History.....	6
About this Guide .....	6
Audience.....	6
Text Conventions.....	6
<b>Chapter 1. AppViewX Software as a Service.....</b>	<b>7</b>
Overview.....	7
Key Highlights of AppViewX Software as a Service.....	7
The AppViewX Cloud Connector.....	8
<b>Chapter 2. Features of the AppViewX Cloud Connector.....</b>	<b>10</b>
AppViewX Cloud Connector DataCenter Significance.....	10
Data Center-based Routing.....	10
Cloud Connector High Availability.....	12
Custom Certificates for Core Communication.....	12
Communication Authentication and Encryption.....	12
Auto Enrollment with the AppViewX Cloud Connector.....	12
Enabling Proxy for End Point Communication.....	12
Configuring Network Proxy Settings .....	13
<b>Chapter 3. System Requirements for Setting up the AppViewX Cloud Connector.....</b>	<b>15</b>
Overview.....	15
Hardware.....	15
Operating System.....	16
Server and Network Prerequisites.....	16
Docker Prerequisites.....	18
<b>Chapter 4. Setting Up the AppViewX Cloud Connector.....</b>	<b>20</b>
Methods to Set up the AppViewX Cloud Connector.....	20
Setting up the AppViewX Cloud Connector via a Virtual Image.....	20

Step 1: Deploying the AppViewX OVA.....	23
Step 2: Accessing the Setup Interface.....	35
Step 3: Executing the Prerequisite Check Script.....	37
Step 4: Installing the AppViewX Cloud Connector.....	40
Setting up the AppViewX Cloud Connector via the Native OS.....	48
Step 1: Configuring the Native OS for Cloud Connector Installation.....	50
Step 2: Accessing the Setup Interface.....	54
Step 3: Executing the Prerequisite Check Script.....	55
Step 4: Installing the AppViewX Cloud Connector.....	58
<b>Chapter 5. Prerequisites for Managing ADC Devices.....</b>	<b>67</b>
Managing F5 and A10 Devices through the AppViewX Cloud Connector.....	67
F5 BIG-IP Devices.....	67
A10 Devices.....	67
<b>Chapter 6. Installing the AppViewX Windows Gateway.....</b>	<b>68</b>
<b>Chapter 7. Troubleshooting the AppViewX Cloud Connector.....</b>	<b>69</b>
Managing Certificates on F5 BIG-IP Devices.....	69
AppViewX Cloud Connector Health.....	69
Connectivity Checks.....	70
Installation Errors.....	70
Log Analysis.....	74
Steps to check pod status.....	75
Expected result:.....	75
Pod Description:.....	75
Steps to restart pods.....	76
Restart specific pod:.....	76
Restart both starter and platform pods:.....	76
<b>Chapter 8. Managing the AppViewX Cloud Connector.....</b>	<b>77</b>
Overview.....	77
Understanding the AppViewX Cloud Connector Inventory.....	77

AppViewX Cloud Connector Actions.....	82
Starting the AppViewX Cloud Connector.....	83
Pausing the AppViewX Cloud Connector.....	83
Upgrading the AppViewX Cloud Connector Version.....	84
Updating the Certificate Configuration.....	85
Deleting an AppViewX Cloud Connector Instance.....	86
Uninstalling an AppViewX Cloud Connector Instance.....	87
Monitoring the Health of the AppViewX Cloud Connector.....	87
<b>Chapter 9. Frequently Asked Questions.....</b>	<b>89</b>
Disabling firewall.....	89
Docker Prerequisites.....	89
Monitoring the Health of the AppViewX Cloud Connector.....	90
Steps to check pod status.....	91
Expected result:.....	91
Pod Description:.....	91
Steps to restart pods.....	91
Restart specific pod:.....	91
Restart both starter and platform pods:.....	92
Deploying the AppViewX OVA.....	92
Updating the AppViewX Virtual Image from the AppViewX Repository.....	92
Prerequisites for Updating the Virtual Image from the Repository.....	92
Updating the Virtual Image from the Repository.....	93
Synchronizing the Node Clock with the Network Time.....	93
Validating the SHA256 Checksum.....	93

# Preface

## Revision History

Revision	Description	Date
3.0	Updated release of document for the release 2022.1.0 FP2	December 2022
2.0	Updated release of document for release 2022.1.0 FP2 Beta	November 2022
1.0	Initial release of document for release 2022.1.0	June 2022

## 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 products: CERT+ SaaS, ADC+ SaaS, PKI+.

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 AppViewX SaaS deployment.

## Text Conventions

The following text conventions are used in this document:

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

# Chapter 1: AppViewX Software as a Service

- [Overview](#)
- [Key Highlights of AppViewX Software as a Service](#)
- [The AppViewX Cloud Connector](#)

## Overview

AppViewX's CERT+ and ADC+ products are now available as a Software as a Service (SaaS) offering for end-to-end Certificate Lifecycle Management and Device Management.

## Key Highlights of AppViewX Software as a Service

The AppViewX Security Automation and Orchestration Platform is a centralized control plane to automate tasks, orchestrate workflows and gain visibility to manage identities at scale, reduce security and compliance risk and ensure secure application availability

The AppViewX SaaS platform offers the following three products:

- CERT+ SaaS, which lets you:
  - 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.

For details, refer the [CERT+ SaaS User Guide](#)

- ADC+ SaaS, which lets you
  - Efficiently distribute network load or client requests across servers.
  - Send requests to the available servers, ensuring high application availability.
  - Scale the number of servers (up or down) based on the traffic.

For details, refer the [ADC+ SaaS User Guide](#)

- PKI+, which lets you:
  - Create root CAs and subordinate CAs and enroll them to the AppViewX PKIaaS certificate authority.
  - Onboard custodians to add root CAs and subordinate CAs to the PKI+ system.
  - Manage custodians for approving PKI+-related actions.

For details, refer the [PKI+ User Guide](#).

## The AppViewX Cloud Connector

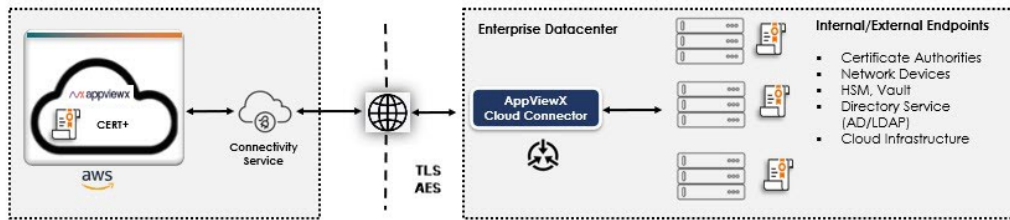
AppViewX Cloud Connector is a lightweight plug-in that establishes connectivity between AppViewX Cloud and the Enterprise Network. The cloud connector serves as a secure channel for communication between AppViewX SaaS and your enterprise network without requiring any complex network or infrastructure configuration

Services that require the AppViewX Cloud Connector for using the AppViewX products (examples):

- **CERT+**
  - Discovering certificates from an endpoint within the enterprise network via Smart Network Scan and Managed Device Scan.
  - Discovering certificates from Certificate Authorities (CAs) that are internal to the enterprise. For example : EJBCA.
  - Discovering certificates from public Certificate Authorities (CAs)

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

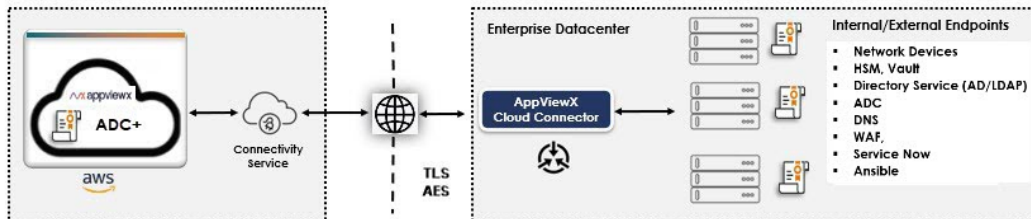
To enable this, at the time of [adding a new AppViewX Cloud Connector instance](#), from the **Data Center** dropdown list, select **DC Routing**.



- **ADC+:**

- Communicating with ADC devices and discover the Application Services from the ADC infrastructure
- Gain Visibility and to fetch the real time state/status of the Applications discovered
- Self Service the Applications to allow/deny traffic
- Backup the configuration of the ADC devices
- Restore the configuration of the ADC devices
- Automate and Orchestrate the ADC configuration within and across devices

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



Key features of the AppViewX Cloud Connector:

- A self-serviceable, Linux-based lightweight setup
- Secure communication between the AppViewX SaaS and the AppViewX Cloud Connector using TLS and AES encryption
- Connectivity from the AppViewX SaaS 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

- [AppViewX Cloud Connector DataCenter Significance](#)
- [Cloud Connector High Availability](#)
- [Custom Certificates for Core Communication](#)
- [Communication Authentication and Encryption](#)
- [Auto Enrollment with the AppViewX Cloud Connector](#)
- [Enabling Proxy for End Point Communication](#)

## AppViewX Cloud Connector DataCenter Significance

AppViewX provides a **default** Cloud Connector DataCenter called **cloud-dc** within the AppViewX SaaS infrastructure for connectivity to the public endpoints such as Cloud Accounts (Cloud service providers - AWS, Azure, Google Cloud) and External Certificate Authorities (Digicert, Entrust, Commodo etc) directly thereby eliminating complex configurations.

Alternatively, users can also set up **dedicated** cloud connectors within the enterprises's cloud infrastructure should there be any source connection restrictions or a need for dedicated communication to the respective public endpoints - Cloud Accounts (Cloud service providers - AWS, Azure, Google Cloud) and External Certificate Authorities (Digicert, Entrust, Commodo, and so on). The cloud connectors can be mapped to a unique DataCenter name.

You can map the DataCenter to the AppViewX Cloud Connector instance at the time of adding the cloud connector.

You can also choose to have a dedicated cloud connector instance for specific endpoints. To do this, at the time of onboarding the endpoint (Managed Devices, Network Scan, Certificate Authority, and so on), select the AppViewX Cloud Connector DataCenter Name from the **DataCenter** field in each of the endpoint onboarding pages.

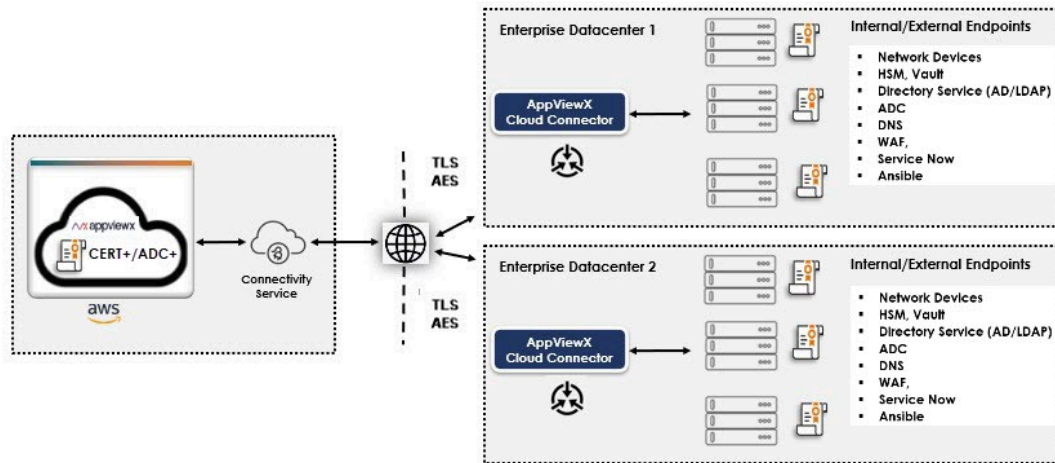
- [Data Center-based Routing](#)

## Data Center-based Routing

The AppViewX Cloud Connector instances that need to connect to the network endpoints are deployed inside a specific DataCenter in an enterprise's premises. Based on the DataCenter in which the cloud

connector is added, the calls to manage the end points are routed to the specific cloud connector inside a DataCenter.

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.

## 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 SaaS. 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.

## Communication Authentication and Encryption

The Cloud Connector authenticates and encrypts all communications between the AppViewX SaaS and the DataCenter where the cloud connector is deployed. All connections are established from the Cloud Connector to the AppViewX SaaS using the standard HTTPS port (443) and the TCP protocol.

## Auto Enrollment with the AppViewX Cloud Connector

The AppViewX Cloud Connector supports Auto Enrollment Protocols like EST, SCEP and ACME. IoT devices should be able to send auto enrollment requests to the AppViewX Cloud Connector using EST, SCEP, and ACME protocols. The cloud connector will be able to capture such information and route it to CERT+.

A gateway service will be running as a part of the AppViewX Cloud Connector to support Auto Enrollment Protocols. Ports that are exposed from this gateway are listed below:

- 30021 - port that will receive EST request
- 30022 - port that will receive SCEP request
- 30020 - port that will receive ACME request

## Enabling Proxy for End Point Communication

You can configure network proxy settings if the machine on which the Cloud Connector is deployed requires communication to external or internal endpoints (Certificate Authority, SSL / TLS endpoints).



**Note:** This is an optional enablement and should be performed only if a proxy setup is required.

• [Configuring Network Proxy Settings](#)

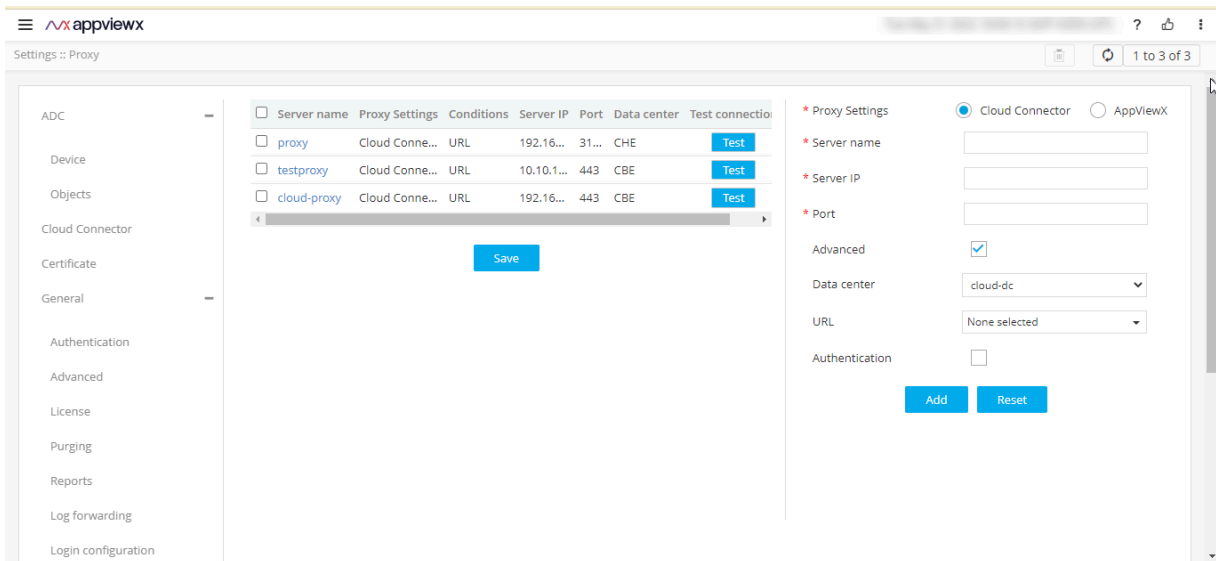
## Configuring Network Proxy Settings

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
<b>Proxy Settings*</b>	<ul style="list-style-type: none"> <li>• If you want to configure proxy settings for the AppViewX Cloud Connector, select <b>Cloud Connector</b>.</li> <li>• If you want to configure proxy settings for the AppViewX product holistically, select <b>AppViewX</b>.</li> </ul>
<b>Server name*</b>	Name of the proxy server
<b>Server IP*</b>	IP address of the proxy server

Field	Description
<b>Port*</b>	Port number of the proxy server
<b>Advanced</b>	To enable advanced settings, select this checkbox.
<b>Data center*</b>	<div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-bottom: 10px;">  <b>Note:</b> This field is displayed only when the <b>Advanced</b> checkbox is selected.         </div> <p>From the dropdown menu, select a data center.</p>
<b>URL</b>	<div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-bottom: 10px;">  <b>Note:</b> This field is displayed only when the <b>Advanced</b> checkbox is selected.         </div> <p>From the dropdown menu, select the URL that should be connected to via the proxy.</p>
<b>Authentication</b>	To enable authentication for accessing the proxy, select this checkbox.
<b>Username*</b>	<div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-bottom: 10px;">  <b>Note:</b> This field is displayed only when the <b>Authentication</b> checkbox is selected.         </div> <p>Enter the username configured for the proxy.</p>
<b>Password*</b>	<div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-bottom: 10px;">  <b>Note:</b> This field is displayed only when the <b>Authentication</b> checkbox is selected.         </div> <p>Enter the password required to access the proxy.</p>

2. Click **Add**.

Details of the added proxy server are displayed in the table on the left.

# Chapter 3: System Requirements for Setting up the AppViewX Cloud Connector

- [Overview](#)
- [Hardware](#)
- [Operating System](#)
- [Server and Network Prerequisites](#)
- [Docker Prerequisites](#)

## Overview

The following sections list the system requirements that are minimum prerequisites for setting up and operating the AppViewX Cloud Connector.



**Note:** If the host machine on which you want to set up the AppViewX Cloud Connector does not/cannot fulfill the operating system, network, and Docker prerequisites (listed below), you can set up the AppViewX Cloud Connector via the AppViewX SaaS OVA, which is a virtual, remotely-accessible setup bundled with the OS, system, and Docker prerequisites for the AppViewX Cloud Connector.

To know more about the OVA and for instructions on setting up the AppViewX Cloud Connector using the AppViewX SaaS OVA, click [here](#).

## 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

## 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/PTP.

For the **ntpd** package (for **CentOS**), execute the following sequence of commands:

```
sudo yum install -y ntp
sudo systemctl enable ntpd
sudo systemctl start ntpd
```

For the **chronyd** package , execute the following sequence of commands:

- For **CentOS**

```
sudo yum install -y chrony
sudo systemctl enable chronyd
sudo systemctl start chronyd
```

- For **Ubuntu**

```
sudo dnf install -y chrony
sudo systemctl enable chronyd
sudo systemctl start chronyd
```

- Ensure that the AppViewX Cloud Connector can establish connectivity with the AppViewX SaaS 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 SaaS 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.

1. Install the **cURL** utility.

- On **Ubuntu**: `sudo apt-get install curl`
- On **CentOS**: `sudo yum install curl`

2. To check connectivity, execute the following command:

- If proxy is not enabled:

```
curl -k --max-time 20 --connect-timeout 20 -s -o /dev/null -w "%{http_code}" "<<https://AppViewX SaaS server URL>>/socket.io/?EIO=3&transport=polling&t=O11wka_"
```

- If proxy is enabled:

```
curl -k --proxy "<<http://proxyhost:proxyport>>" --max-time 20 --connect-timeout 20 -s -o /dev/null -w "%{http_code}" "<<https://AppViewX SaaS server URL>>/socket.io/?EIO=3&transport=polling&t=O11wka_"
```

If connectivity has been established successfully, the command will return the HTTP code **200**. If the command returns any other code, it indicates that connectivity is not established.

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

- For **CentOs**

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
```

- For **Ubuntu**

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 where the AppViewX Cloud Connector is to be installed.

- For **CentOs**

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
```

- For **Ubuntu**

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 **nftables** service.

- For **CentOS**

```
sudo yum remove -y nftables
```

- For **Ubuntu**

```
sudo apt purge -y nftables
```

## 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/>



**Important:** 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](#).

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

# Chapter 4: Setting Up the AppViewX Cloud Connector

- [Methods to Set up the AppViewX Cloud Connector](#)
- [Setting up the AppViewX Cloud Connector via a Virtual Image](#)
- [Setting up the AppViewX Cloud Connector via the Native OS](#)

## Methods to Set up the AppViewX Cloud Connector

The AppViewX Cloud Connector can be set up in two ways:

- **Via a Virtual Image**

The AppViewX Open Virtual Appliance (OVA) is a virtual, remotely-accessible setup that is bundled with the [software](#), [network](#), and [Docker](#) prerequisites for installing the AppViewX Cloud Connector without altering the OS configuration on their systems. (The .ova file that can be downloaded from [here](#)).

- **Via the Native OS**

The host machine on which the AppViewX Cloud Connector has to be installed must fulfil a certain set of [prerequisites](#) across the following categories: [hardware](#), [operating system](#), [Docker](#), and [server and network](#). If all prerequisites are met, you can [install the AppViewX Cloud Connector via the Native OS](#).

## Setting up the AppViewX Cloud Connector via a Virtual Image

The AppViewX Virtual Image is an Open Virtual Appliance (OVA) that is bundled with the [operating system](#), [server and network](#), and [Docker](#) prerequisites for installing the AppViewX Cloud Connector without altering the OS configuration on their systems.



**Note:** The AppViewX SaaS OVA is CIS benchmarked.

The AppViewX SaaS OVA offers the following advantages:

- Built with CentOS version 7.9
- Docker 20.10.5 pre installed with all required permissions
- Hardened OVA with all security issues addressed

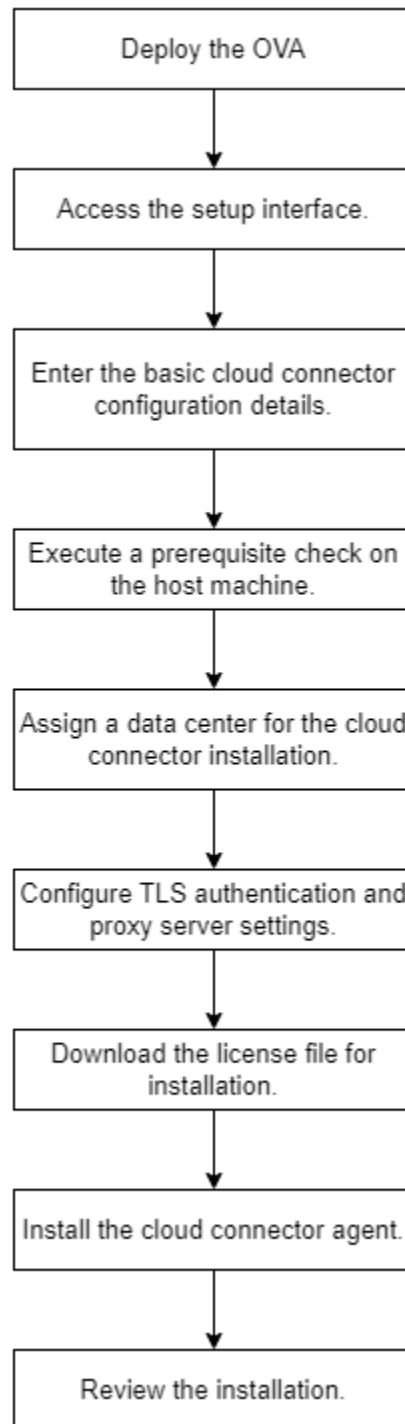


**Note:** Detailed instructions for updating the AppViewX virtual image from the AppViewX repository are documented [here](#).



**Note:** If this AppViewX Cloud Connector installation requires configuring a proxy server, click [here](#) for instructions.

The process of setting up the AppViewX Cloud Connector via the virtual image comprises of the following steps (explained in the subsequent sections):



- [Step 1: Deploying the AppViewX OVA](#)
- [Step 2: Accessing the Setup Interface](#)

- [Step 3: Executing the Prerequisite Check Script](#)
- [Step 4: Installing the AppViewX Cloud Connector](#)

## Step 1: Deploying the AppViewX OVA

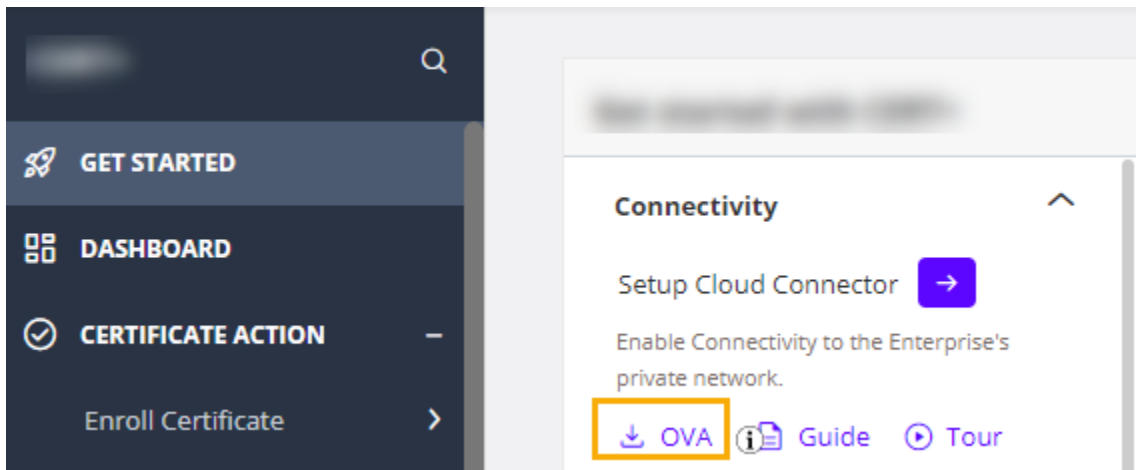


**Note:** If the node meets all the software, network, and Docker prerequisites, **skip this step.**

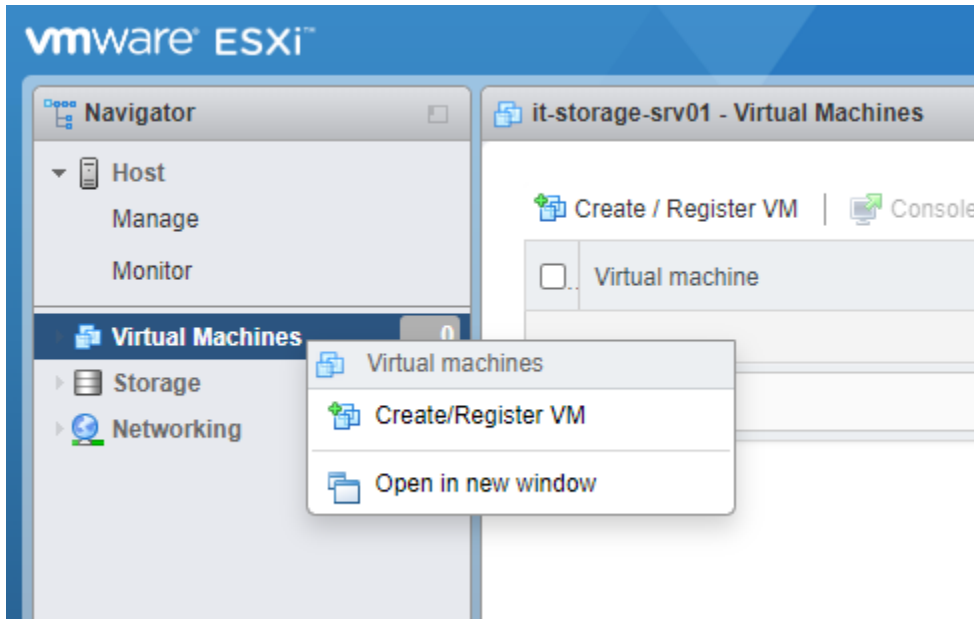
- [Deploying the AppViewX Virtual Machine for an On-prem Deployment](#)
- [Deploying the AppViewX Virtual Machine for AWS](#)
- [Deploying the AppViewX Virtual Machine for Azure](#)
- [Deploying the AppViewX Virtual Machine for GCP](#)

## Deploying the AppViewX Virtual Machine for an On-prem Deployment

1. To download the release package in the OVA format, from the respective AppViewX's product line landing page, under **GET STARTED** menu > **Connectivity** section, click [↓ OVA](#).

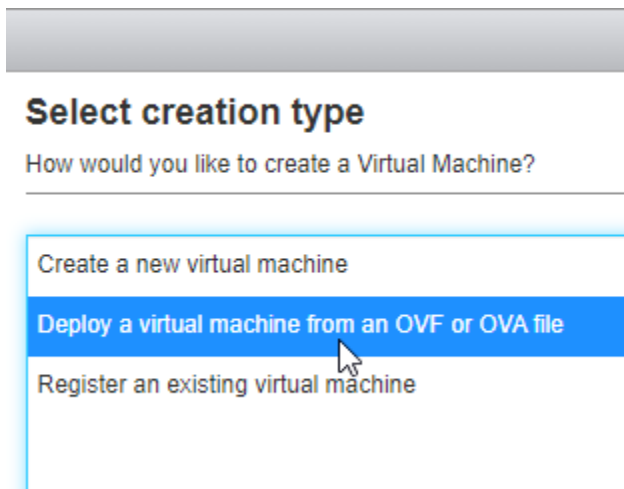


2. Log in to the **vmware** client.
3. From the **Navigation** pane on the left, right click **Virtual Machines**.
4. Click **Create/Register VM**.



The **New Virtual machine** window is displayed.

5. From the navigation pane in the left, select **Select creation type**.
6. In the **Select creation type** window, select the **Deploy a virtual machine from an OVA or OVF file** option.



7. Click **Next**.
8. In the **Select OVF and VMDK files** window:

Deployment

### Select OVF and VMDK files

Select the OVF and VMDK files or OVA for the VM you would like to deploy

---

Enter a name for the virtual machine.

Virtual machine names can contain up to 80 characters and they must be unique within each ESXi instance.

Click to select files or drag/drop

- a. Enter a name for the virtual machine.  
For the purpose of this document, we will name it **test-cc-deployment**.
  - b. In the **Click to select files or drag/drop** area, click and, from the file explorer, navigate to the location of the file, select the file, and click **Open**.
9. Click **Next**.
10. In the **Select storage** window, from the available options, select a datastore for storing the virtual machine's files and all of its virtual disks.

### Select storage

Select the storage type and datastore

---

**Standard** Persistent Memory

Select a datastore for the virtual machine's configuration files and all of its' virtual disks.

Name	Capacity	Free	Type	Thin pro...	Access
SSD	244.5 GB	238.85 GB	VMFS6	Supported	Single
VMStore	931.25 GB	921.21 GB	VMFS5	Supported	Single

2 items

11. Click **Next**.
12. In the **Deployment options** window:

### Deployment options

Select deployment options

Network mappings	VM Network_192.168.31.x <div style="border: 1px solid #ccc; padding: 2px; width: 150px;">HS data</div>
Disk provisioning	<input checked="" type="radio"/> Thin <input type="radio"/> Thick
Power on automatically	<input checked="" type="checkbox"/>

- a. Select the network mapping.
  - b. Select the disk provisioning required.
  - c. Select the **Power on automatically** checkbox.
13. Click **Next**.
  14. In the **Ready to complete** window, review your settings.

### Ready to complete

Review your settings selection before finishing the wizard

Product	CENTOS_CC_BASE_VM
VM Name	test-cc-deployment
Files	centos_cis_compliant_cc-disk1.vmdk
Datastore	VMStore
Provisioning type	Thin
Network mappings	VM Network_192.168.31.x: HS data
Guest OS Name	Unknown



Do not refresh your browser while this VM is being deployed.

15. Click **Finish**.
  - The progress of the OVA deployment is shown in the **Recent Tasks** section.

Task	Target	Initiator	Queued	Started	Result	Completed
Upload disk - centos_cis_compliant_cc-dis...	test-cc-deployment	root	04/23/2022 02:43:44	04/23/2022 02:43:44	<div style="border: 1px solid #ccc; width: 100px; height: 15px; background-color: #ccc;"></div>	Running... 0 %
Create VM	vm	root	04/23/2022 02:42:43	04/23/2022 02:42:43	Completed successfully	04/23/2022 02:42:45
Import VApp	Resources	root	04/23/2022 02:42:43	04/23/2022 02:42:43	<div style="border: 1px solid #ccc; width: 100px; height: 15px; background-color: #ccc;"></div>	Running... 0 %

Task	Target	Initiator	Queued	Started	Result	Completed
Upload disk - centos_cis_compliant_co-dis...	test-cc-deployment	root	04/23/2022 02:43:44	04/23/2022 02:43:44	Completed successfully	04/23/2022 03:21:40
Import VApp	Resources	root	04/23/2022 02:42:43	04/23/2022 02:42:43	Completed successfully	04/23/2022 03:20:48
Create VM	vm	root	04/23/2022 02:42:43	04/23/2022 02:42:43	Completed successfully	04/23/2022 02:42:45
Power On VM	test-cc-deployment	root	04/23/2022 03:20:48	04/23/2022 03:20:48	Completed successfully	04/23/2022 03:20:52

- On successful completion of the OVA deployment, the new virtual machine is displayed in the **Virtual Machines** inventory. For each virtual machine in the inventory, the following details are displayed:

Virtual machine	Status	Used space	Guest OS	Host name	Host CPU	Host memory
test-cc-deployment	Normal	10.57 GB	CentOS 7 (64-bit)	Unknown	0 MHz	0 MB

16. From the **Virtual Machines** inventory, click the virtual machine just added.

The terminal window for the virtual machine is displayed. The script for configuring the network IP is executed automatically.

17. To configure the IP address, when prompted, enter the required values for the following requested parameters:

```

=====
IPADDR = XXX.XXX.XXX.XXX

NETMASK = XXX.XXX.XXX.XXX

GATEWAY = XXX.XXX.XXX.XXX

=====
    
```

For example, refer to the sample screenshot below:

```

CC_NW_TESTING
#####
## Network Configuration
#####
Provide the required informations for [redacted]
Enter ip address
[redacted]
Enter netmask
[redacted]
Enter gateway IP
[redacted]

Information Provided
#####
# IPADDR=[redacted]
# NETMASK=[redacted]
# GATEWAY=[redacted]
#####
Proceed [Y/N]
Y
Device [redacted] successfully disconnected.
Connection successfully activated (D-Bus active path: [redacted])

Could not set property: Connection timed out
Need to configure Hostname [Y/N]
Y
Enter hostname fqdn
[redacted]

```

18. To configure the hostname and the DNS, when prompted, press **Y**. If you prefer to configure the hostname and DNS manually, to skip this step, press **N**.
19. To configure an NTP server(s):
  - a. When prompted, **Do you want to configure ntpd server (default public server)** press **Y**.

```

N
resolv.conf configuration is skipped..
Do you want to configure ntpd server (default public server) [Y/N]
Y
Enter the number of servers :
1
Enter server 1 ip :
[redacted]


```

- b. Enter the number of NTP servers to be configured.
  - c. For the number of servers entered above, enter the IP address of each NTP server on a new line.
  - d. To update the **ntp.conf** file with the IP addresses provided above, press **Y**.
20. After the script is executed, when prompted, login to the VM as the **appviewx** user, using the following credentials:
  - Username: **appviewx**
  - Password: **de393\$1w21KR**



**Note:** Root user access is required for maintaining the OS configuration and for patching security updates. Since direct root access is not provided, you can:


- a. Login as the **appviewx** user.
- b. Switch to the root user by executing the command `sudo -i`.

 **Note:** It is recommended that, after the first login, change the default credentials.

21. To check if the Docker is up and running, execute the command: `systemctl status docker`.

If the Docker status is **active (running)**, as shown in the screenshot below, it means that the OVA has been deployed successfully.

```
[appviewx@ccnode ~]$ systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2022-06-14 06:08:50 EDT; 6 days ago
     Docs: https://docs.docker.com
   Main PID: 1540 (dockerd)
    CGroup: /system.slice/docker.service
            └─1540 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

 **Note:** To check if the Docker is accessible to the appviewx user, execute the following command:

```
docker image ls
```


If the command does not return an error, it means that the Docker is accessible to the appviewx user:

22. For secure access and management of the remote system, on the command line interface, execute the following command: `ssh appviewx@<hostname>`

23. When prompted **Are you sure you want to continue connecting**, press **y** and then press **Enter**.

```
~$ ssh appviewx@
The authenticity of host ' [redacted] ' can't be established.
ECDSA key fingerprint is SHA256:fNQG4/+CfA9BC/bBQpSnf90cke2JvUxWJP5rrPIfHww.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added ' [redacted] ' (ECDSA) to the list of known hosts.
Authorized uses only. All activity may be monitored and reported.
appviewx@ [redacted] 's password:
```

## Deploying the AppViewX Virtual Machine for AWS

 **Note:** For the AWS AMI, the following two operating systems are supported: **Ubuntu** and **Amazon Linux 2**.

### Prerequisites

- Relay your requirements to your assigned AppViewX Solution Architect and finalize a deployment model.
- Share your AWS account number and region with your Solution Architect. The Solution Architect will use these details to create a custom AMI based on your account and region.

When the AppViewX AMI is successfully shared with your customer account, AppViewX will notify you of this development via email.

- From your AppViewX Onboarding Engineer, get the default password for the **appviewx** user.

To install the AppViewX Cloud Connector on AWS, you will need a virtual machine that is preconfigured for the operating system and software stack prerequisites. AWS uses AMI to create pre-configured EC2 instances as per AppViewX standards and requirements.

To create an EC2 instance using the AppViewX AMI:

1. Login to the AWS Management Console and go to **EC2 > Images > AMIs**.

The **Amazon Machine Images (AMIs)** page is displayed.

2. On the **Amazon Machine Images (AMIs)** page, from the **Owned by me** dropdown list, select **Private images**.

All AMIs with visibility set to private are listed. This list will also have the AMI that is created and shared by AppViewX for your requirements.

3. From this list, select the checkbox for the AMI shared by AppViewX.
4. Click **Launch instance from AMI**.

The **EC2 > Instances > Launch an instance** page is displayed.

5. Enter the **Name and tags** to be associated with this EC2 instance.



**Note:** The **Application and OS Images (Amazon Machine Images)** section will show the configuration details of the AppViewX AMI.

6. For the master node, select the following hardware configuration:

#### Instance type

**c5.xlarge**  
 Family: c5    4 vCPU    8 GiB Memory  
 On-Demand Linux pricing: 0.17 USD per Hour  
 On-Demand Windows pricing: 0.354 USD per Hour

7. To securely connect to the EC2 instance, in the **Key pair (login)** section:

- a. To use an existing key pair, from the **Key pair name** dropdown list, select the key pair you want to use.

OR

- a. To create a new key pair, Click **Create new key pair**.
8. In the **Network settings** section, under **Firewall (security groups)**, as required, create a new security group or select an existing security group.
9. If you select **Select existing security group** in the previous step, from the **Common security groups** dropdown list, select the required security group.
10. From the bottom-right corner of the screen, click **Launch instance**.  
The **Launching instance** page is displayed, which shows you the progress of the launch. As soon as the launch is initiated, you will get a success message.
11. Under **Success**, click **Launch log** to review the instance details.
12. From the page name (**EC2 > Instances > Launch an instance**), click **Instances** to go back to the previous page.
13. From the list of instances, select the AWS instance just created.
14. To login to this AWS instance using the key pair .pem file, execute the following command:

- For Ubuntu:

```
ssh -i newkey.pem ubuntu@<public ipaddress of the aws instance>
```

- For Amazon Linux 2:

```
ssh -i newkey.pem ec2-user<public ipaddress of the aws instance>
```

15. To switch to the **sudo** user, execute the following command:

```
sudo -i
```

16. To switch to the **appviewx** user, execute the following command:

```
sudo su - appviewx
```

17. In order to successfully execute the installation, AppViewX needs to run a script for which authentication via the **.pem** file needs to be bypassed. To do this, execute the following commands:

```
sudo sed -i 's/.*PasswordAuthentication.*/PasswordAuthentication yes/g' /etc/ssh/sshd_config
sudo systemctl restart sshd
```

## Deploying the AppViewX Virtual Machine for Azure

To install the AppViewX Cloud Connector in Azure, you need to create a virtual machine (VM) using the Azure Virtual Hard Disk (VHD). Microsoft Azure uses the Azure VHD file format to store the virtual machine (VM) disk images that are containers preloaded with the operating system, network, applications, and data requirements for setting up a virtual machine.

To deploy the AppViewX virtual machine for Azure:

1. Go to <https://release.appviewx.com/Login> and, from **Overview**, navigate to **2022.1.0**.
2. Scroll down to Production Images and download the latest artifact of Azure CC VHD, **AppViewX-2022.1.3-FP3-CC-Ubuntu-Azure-ddmmmyyy-vhd.tar.gz**.
3. Untar the downloaded artifact.

```
tar -xvf AppViewX-2022.1.3-FP3-CC-Ubuntu-Azure-08Jun2023-vhd.tar.gz
```

4. Download the **Azure Storage Explorer** from [here](#).

The **Azure Storage Explorer** is a desktop application that provides you with a GUI for easily managing your Azure resources.



**Important:** Install the Azure Storage Explorer at the same location as the downloaded Azure CC VHD artifact.

5. Using the Azure Storage Explorer, login to the Azure account for which the VM has to be created.
6. On successful login, go to the **disks** section and select the resource group.  
The resource group page is displayed.
7. Click **Upload**.
8. In the pop-up window displayed, enter/select the resource details.

#### Descriptions for the resources and their corresponding values

Resource	Value
Source VHD	<Disk file location>
Disk name	<Name of the disk>
OS type	<b>Linux</b>
Location	<region in which the VM is to be created>
Availability Zone	<zone name>
Account type	<b>Premium SSD</b>
Hyper-V Generation	<b>V1</b>
Architecture	<b>x64</b>



**Note:** All the values in bold, in the above table, are actual values and have to be assigned as is; values enclosed in angle brackets (<>) have to be assigned as per your specific configuration.

9. Click **Create**.
10. Once the disk is successfully uploaded to the Azure cloud, from the Azure portal, select the disk and click **+Create VM**.
11. On the **Create a virtual machine** page, configure the VM configurations based on your organization's standards and requirements.
12. Once the VM is successfully created, use the **.pem** file shared by AppViewX's SRE/TS team to login to the node (on which the cloud connector is to be installed). To do this, on the Command Line Terminal or Powershell, execute the following command:

```
ssh -i Azure-CC_key.pem azureuser@<ip_of_the_CC_VM>
```

**Output:**

```
~/Downloads$ ssh -i Azure-CC_key.pem azureuser@10.96.0.4
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-1038-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Wed May 31 12:16:59 UTC 2023

System load:  0.09521484375   Processes:            146
Usage of /:   5.0% of 28.89GB   Users logged in:     0
Memory usage: 1%              IPv4 address for eth0: 10.96.0.4
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Wed May 31 10:45:57 2023 from 10.109.0.4
```

13. To switch to the root user, execute the following command:

```
sudo -i
```

**OR**

To switch to the appviewx user, execute the following command:

```
su - appviewx
```

14. [Optional] To enable login without the **.pem** file:

- a. Enable password authentication in the Azure node in which the AppViewX Cloud Connector is installed by executing the following commands:

```
sudo sed -i 's/^(PasswordAuthentication \).*\1yes/ /etc/ssh/sshd_config
sudo sed -i 's/^(PasswordAuthentication \).*\1yes/ /etc/ssh/sshd_config.d/50-cloud-init.conf
```

- b. Enable root password authentication by executing the following commands:

```
sudo sed -i '/^#PermitRootLogin/s/^#/' /etc/ssh/sshd_config
sudo sed -i 's/^PermitRootLogin.*/PermitRootLogin yes/ /etc/ssh/sshd_config
```

15. The SSH (Secure Shell) protocol is used for secure administration (login, command execution) over remote networks. For SSH changes to take effect, restart the SSH service. To do this, execute the following command:

```
sudo service ssh restart
```

16. To login without the **.pem** file, execute the following commands:

```
ssh appviewx@<ip/hostname>
ssh root@<ip/hostname>
```

17. Update the **/etc/hosts** file for the IP and the hostname of the VM created, using the following commands:

```
vi /etc/hosts
hostnamectl set-hostname "hostname-of-the-vm"
```

18. To validate the update to the **/etc/hosts** file, execute the following commands:

```
hostname -i
hostname -f
hostname
```

**What to do next:** Go to the setup interface to begin setting up the AppViewX Cloud Connector via the virtual machine.

## Deploying the AppViewX Virtual Machine for GCP

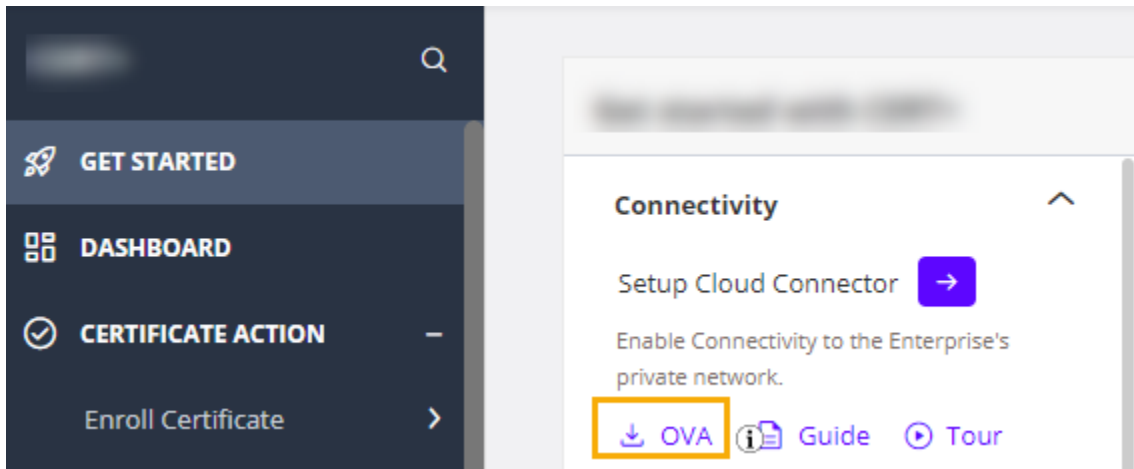
1. Log in to the GCP console and start a **Cloud Shell** instance.
2. From the command line terminal, create a bucket with the default settings.

```
gsutil mb gs://my-virtual-appliances-bucket
```

3. Download the GCP OVA from the [AppViewX release portal](#).

Alternatively, you can Download the release package in the OVA format, from the respective AppViewX's product line landing page, under **GET STARTED** menu > **Connectivity** section, click

[↓ OVA](#)



4. Upload it to the Cloud Shell. To upload/copy the OVA to the GCP bucket, execute the following command:

```
gsutil cp ~/path-to-file/local
gs://my-virtual-appliances-bucket/my-va-file.ova
```

5. Create a virtual instance from the OVA.

```
gcloud compute instances import <my-instance> \
--source-uri=gs://my-virtual-appliances-bucket/my-va-file.ova \
--zone southamerica-east1-a \
--os=ubuntu-1804
```

**What to do next:** Go to the setup interface to begin setting up the AppViewX Cloud Connector via the virtual machine.

## 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.

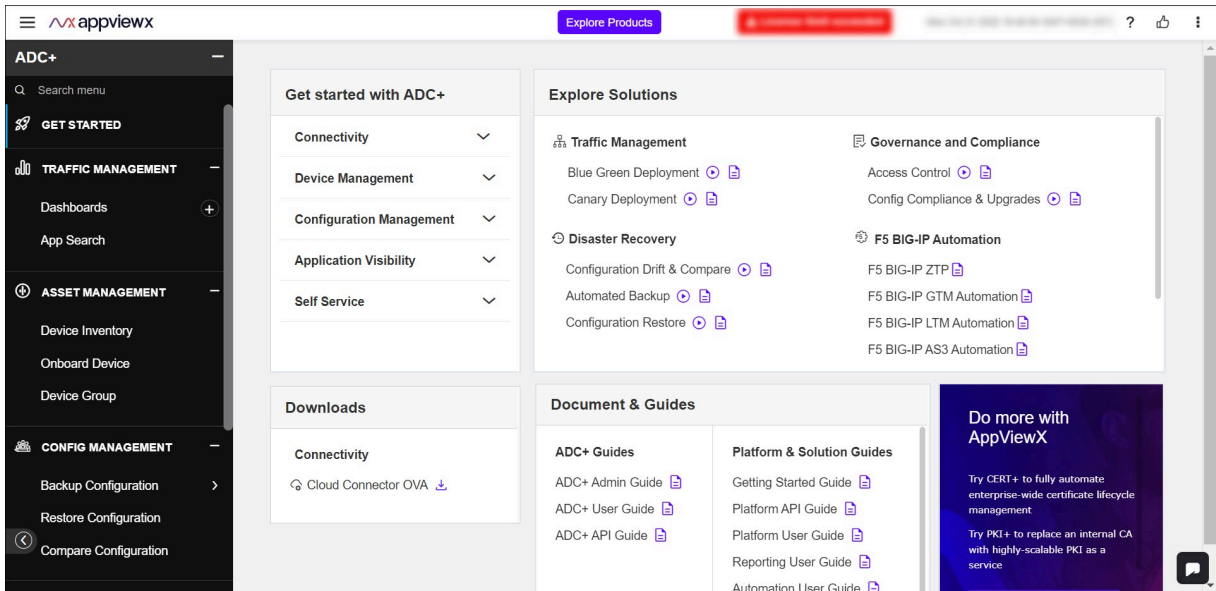
1. Enter your SaaS account URL (for example, <https://tenant-name.appvx.com/appviewx/login>) in the address bar of your browser.

The AppViewX SaaS login page is displayed.

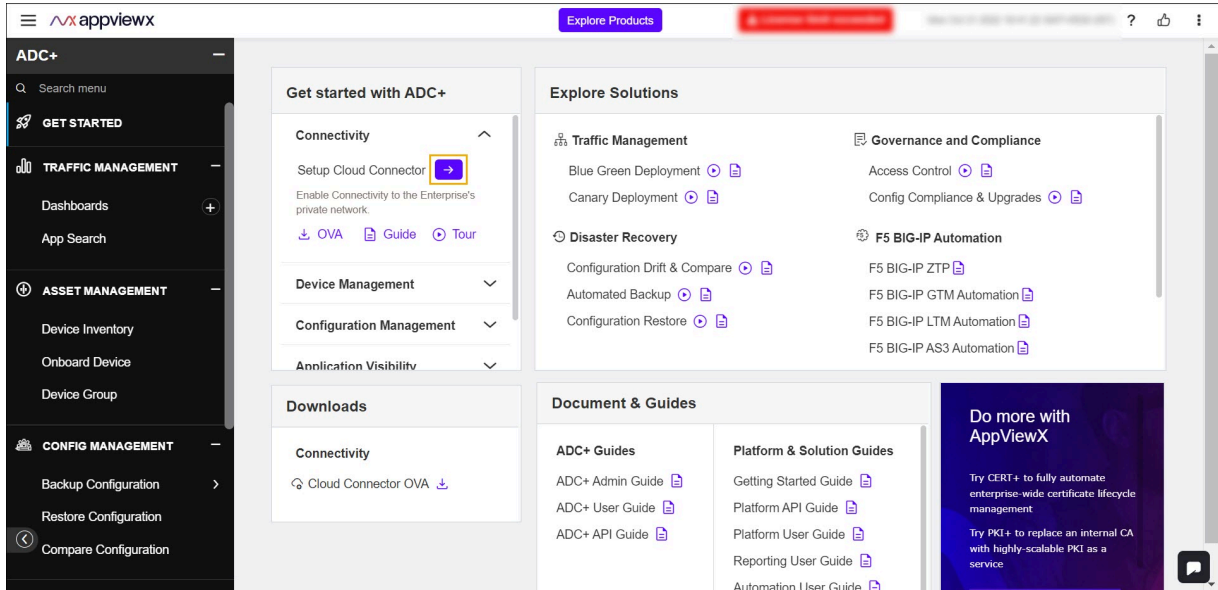
2. Using the credentials sent as part of the Welcome email, login to the AppViewX SaaS.

The AppViewX SaaS landing page is displayed.

**Note:** The landing page differs based on a selected AppViewX SaaS product. The following image shows ADC+ landing page.



3. Click  located beside the SetUp Cloud Connector under the Connectivity section.



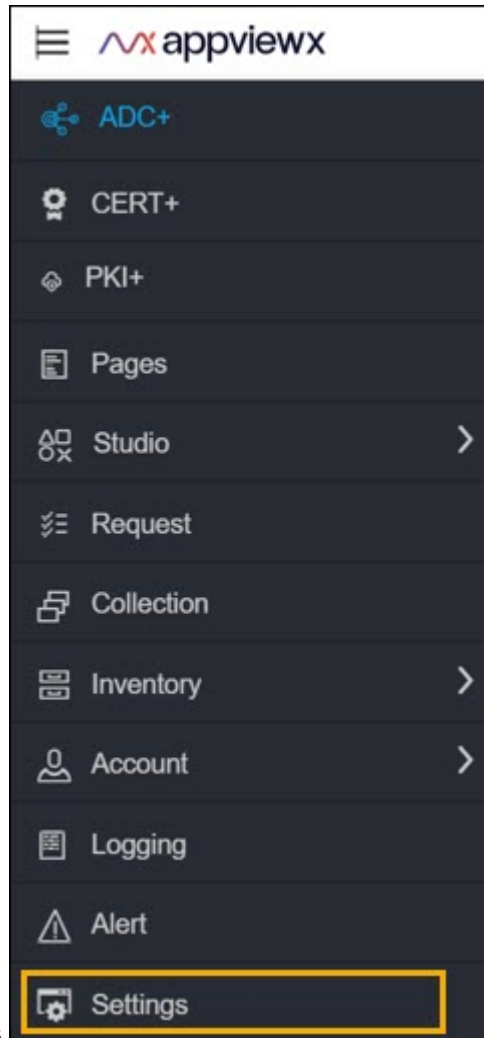
Redirected to the Settings :: Cloud Connector page.

### Step 3: 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:

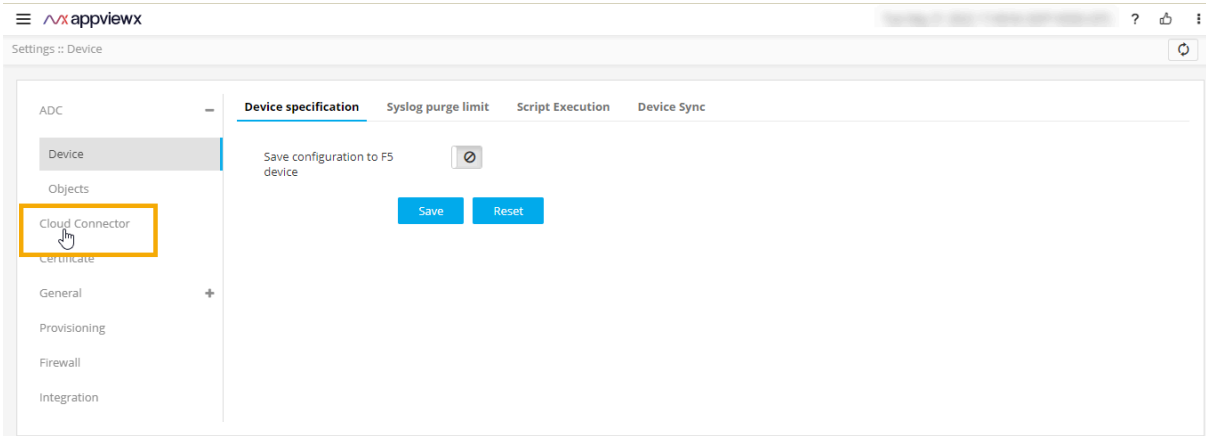
1. After successfully logging in to the AppViewX SaaS GUI, from the top left corner of the screen, click



2. From the menu displayed, click **Settings**.

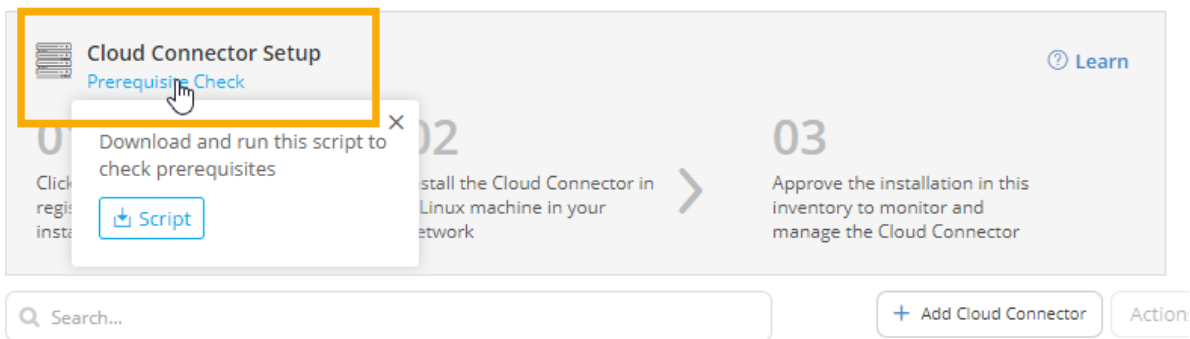
By default, the **Settings :: Device** page is displayed.

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

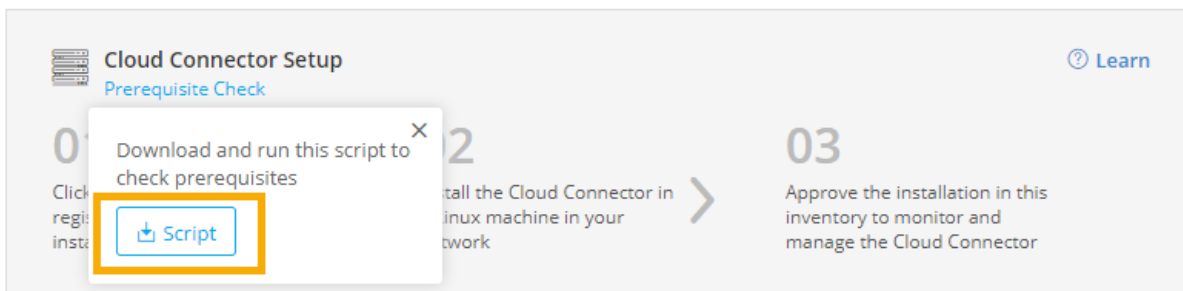


The **Settings :: AppViewX Cloud Connector** page is displayed.

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



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



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

- 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.

```

root@server: ~# ./pre-requisite-check.sh
root@server: ~# ./pre-requisite-check.sh
*
*                               *
*       Performing the initial checks...                               *
*                               *
*****
Proxy configuration details
No HTTP proxy set.
No HTTPS proxy set.

Using system proxy settings...
Performing firewall daemon check
0
Performing connectivity check...
Connection to AppViewX cloud: 20.10.7.100:443 is OK
Performing docker check...
Docker version 20.10.7, build f0df350
Docker is installed.
Docker version check OK
Docker is running...
Performing architecture check...
The architecture check OK
Performing disk check...
Disk space check Ok
Performing memory check...

      ErrorCode      : CC_CONF_005
      ErrorMessage    : Insufficient memory (Free memory: 1335m)
      Operation       : Memory check
      Probable causes : 1. Available primary memory is less
      Suggested remediation : 1. Required RAM specification: 4gb
root@server: ~#

```



**Note:** For resolutions to the prerequisite check failure scenarios, click [here](#).

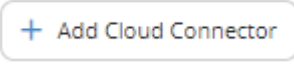
## Step 4: Installing the AppViewX Cloud Connector

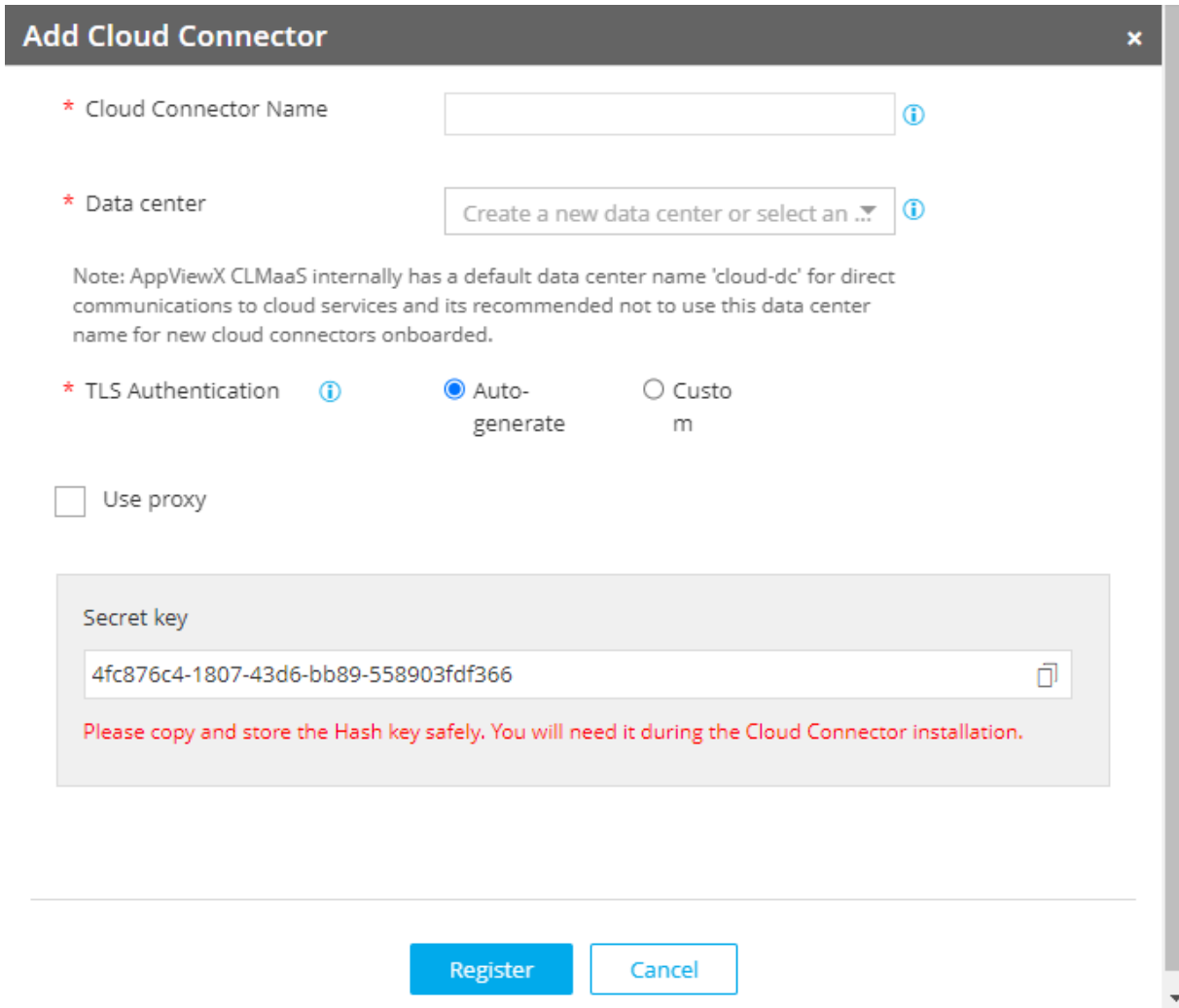
The process of deploying the AppViewX Cloud Connector involves the steps listed below. The substeps for each of these steps are outlined in the subsequent sections.

- [Downloading the Installer](#)
- [Installing the AppViewX Cloud Connector Agent](#)
- [Reviewing the Installation](#)

## 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.



**Add Cloud Connector** ×

\* Cloud Connector Name  ⓘ

\* Data center  ⓘ

Note: AppViewX CLMaaS internally has a default data center name 'cloud-dc' for direct communications to cloud services and its recommended not to use this data center name for new cloud connectors onboarded.

\* TLS Authentication ⓘ  Auto-generate  Custom




Use proxy





Secret key


ⓘ

Please copy and store the Hash key safely. You will need it during the Cloud Connector installation.

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

Field	Description
<b>Cloud Connector Name*</b>	FQDN of the machine where the AppViewX Cloud Connector is to be installed
<b>Data center*</b>	<p>Name of the data center where the AppViewX Cloud Connector is to be installed</p> <div data-bbox="440 495 1419 716" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;">  <b>Note:</b> The AppViewX SaaS, internally, has a default data center named <b>cloud-dc</b>, for direct communications to cloud services. It is recommended that you do not this data center name for new the AppViewX Cloud Connectors onboarded. </div>
<b>Authentication Type</b>	<ul style="list-style-type: none"> <li>• Select the authentication type from the following options: <ul style="list-style-type: none"> <li>• <b>TLS</b></li> <li>• <b>MTLS</b></li> </ul> </li> <li>• To auto-generate a TLS certificate, select <b>Auto-generate</b> (default selection). Automatically, the certificate is generated using the AppViewX CA.</li> </ul> <div data-bbox="461 1045 1419 1350" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;">  <b>Note:</b> The created certificate is available in the certificate inventory. You can: <ul style="list-style-type: none"> <li>• Assign this certificate to a certificate group</li> <li>• Configure a certificate expiry alert for this certificate group from the <b>Server Certificate</b> dashboard, using the <b>Certificate Summary Report</b> widget settings</li> </ul> </div> <ul style="list-style-type: none"> <li>• To enter details of a custom TLS certificate, select <b>Custom</b>. The <b>TLS Certificate Password</b> and <b>Custom TLS Certificate</b> fields are displayed. The instructions for filling these fields are given below.</li> </ul>
<b>TLS Certificate Password*</b>	<div data-bbox="440 1587 1419 1717" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;">  <b>Note:</b> This field is displayed only if you have selected to enter details of a Custom TLS certificate in the <b>TLS Authentication</b> field. </div> <p>Password of the TLS certificate (that will be uploaded in the next step)</p>

Field	Description
	<p> <b>Note:</b> This is a mandatory field if a Custom TLS certificate is uploaded. AppViewX supports only password-protected Custom TLS certificates.</p>
<p><b>Custom TLS Certificate</b></p>	<p> <b>Note:</b> This field is displayed only if you have selected to enter details of a Custom TLS certificate in the <b>TLS Authentication</b> field.</p> <p>To upload a custom TLS certificate:</p> <ol style="list-style-type: none"> <li>To navigate to the location of the custom TLS certificate, click within the field.</li> <li>Select the certificate file.</li> <li>Click <b>Open</b>.</li> <li>To upload the custom TLS certificate selected, click <b>Upload</b>.</li> </ol> <p> <b>Note:</b> AppViewX supports only password-protected Custom TLS Certificates.</p>
<p><b>Use proxy</b></p>	<p>To use a proxy server for the deployment:</p> <ol style="list-style-type: none"> <li>Select the <b>Use proxy</b> checkbox.</li> <li>To select a preconfigured proxy (for the selected data center), from the <b>Select Proxy</b> dropdown list, select a proxy server.</li> </ol> <p><b>OR</b></p> <p>To create a new proxy server setting:</p> <ol style="list-style-type: none"> <li>Use the <a href="#">Click here</a> option shown below the <b>Select Proxy</b> dropdown list.</li> <li>For steps to create a new proxy server setting, click <a href="#">here</a>.</li> </ol>
<p><b>Secret key</b></p>	<p>A unique key for the AppViewX Cloud Connector installation</p> <p>Click  to copy this key and save it in a safe place.</p>


Field	Description
	 <b>Note:</b> Ensure that the secret key is copied and stored safely before you click <b>Register</b> . If you don't, the key is lost forever. For security reasons, the key is not stored within the product.

- 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](#).


Cloud Connector Name	Status	Data Center	Strict Routing	Version	View Log	Action	TLS Certificate	Proxy	Last Heartbeat	Registered On	SHA256 Checksum
aa-aa	Waiting for response	CBE	<input type="checkbox"/>	22.1.0.5	Upgrade	View	Approve	Reject aa-aa	05/27/2022 19:30	299d17edcb377a52fc45...	
aaa.ccc.com	Waiting for response	CBE	<input type="checkbox"/>	22.1.0.5	Upgrade	View	Approve	Reject aaa-ccc.com	05/27/2022 18:48	721ecc30a419c35af17a7...	

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

- To download the installation package, for the AppViewX Cloud Connector, click .
- In the **Confirmation Message** dialog box, click **Yes**.

The download progress is shown using a progress bar.

## Installing the AppViewX Cloud Connector Agent

 **Note:** The following steps assume that:

- All system prerequisites are fulfilled by the host machine.
- The AppViewX Cloud Connector installer (downloaded in the above step) is securely copied via SCP/SFTP to the host machine where the AppViewX Cloud Connector is to be installed.
- The Secret Key received is kept handy for installation.

- 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`
- 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.

3. 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:
kubect! 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.

kubect! 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

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

 21.1.0.1 .



**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.

## 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.

## Setting up the AppViewX Cloud Connector via the Native OS

You can install the AppViewX Cloud Connector interactively on the command line of the machine by following the below instructions.



**Note:** The installation proceeds with the privileges of the user who begins the installation.



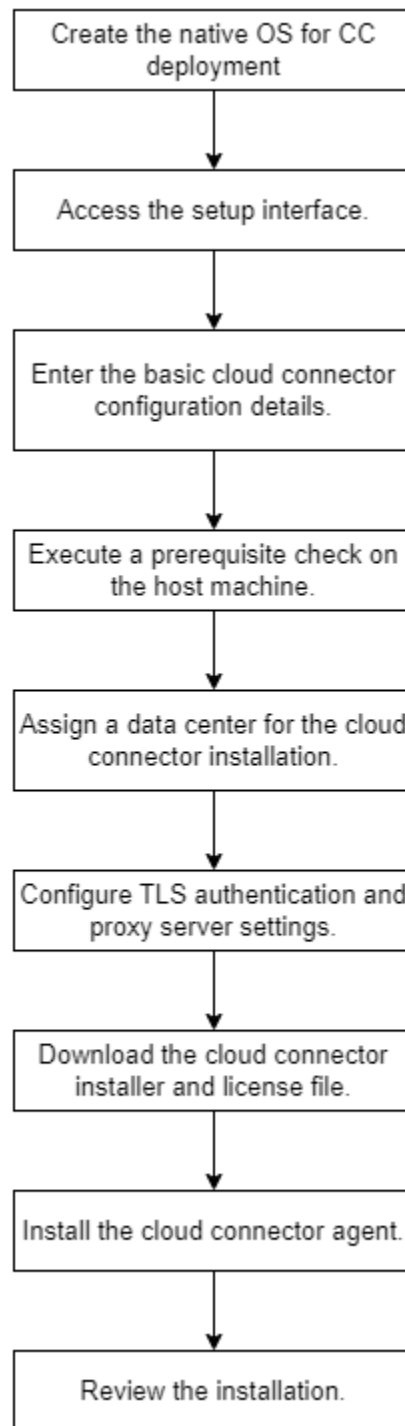
**Note:** The steps for installing the AppViewX Cloud Connector via the native OS assume that you have gone through the [system requirements](#) across the following categories: [hardware](#), [operating system](#), [Docker](#), and [server and network](#).

If the host machine does not/cannot fulfill the installation prerequisites, you can set up the AppViewX Cloud Connector via the AppViewX SaaS OVA. To know more about the OVA and for instructions on setting up the AppViewX Cloud Connector using the AppViewX SaaS OVA, click [here](#).



**Note:** If this AppViewX Cloud Connector installation requires configuring a proxy server, click [here](#) for instructions.

The process of setting up the AppViewX Cloud Connector via the native OS comprises of the following steps (explained in the subsequent sections):



- [Step 1: Configuring the Native OS for Cloud Connector Installation](#)
- [Step 2: Accessing the Setup Interface](#)

- [Step 3: Executing the Prerequisite Check Script](#)
- [Step 4: Installing the AppViewX Cloud Connector](#)

## Step 1: Configuring the Native OS for Cloud Connector Installation



**Note:** Before executing the following instructions, ensure that you have root user permissions.

1. Create the **appviewx** user and directory.

```
useradd appviewx && mkdir /home/appviewx/ && chown appviewx:appviewx /home/appviewx && chmod 700 /home/appviewx
```

2. Uninstall the previous version of Docker.

- **For CentOS:**

```
sudo yum remove docker \
docker-client \
docker-client-latest \
docker-common \
docker-latest \
docker-latest-logrotate \
docker-logrotate \
Docker-engine
```

- **For Ubuntu**

```
for pkg in docker.io docker-doc docker-compose podman-docker containerd runc; do sudo apt-get remove $pkg; done
```

3. Configure the hostname and the nameserver.

- a. Configure the hostname. `sudo hostnamectl set-hostname "hostname"`

- b. Add the host IP address at the end of the **hosts** file.

```
sudo vi /etc/hosts
<ip> <hostname>
```

- c. To replace the default nameserver IP address:

```
sudo systemctl stop systemd-resolved.service
sudo systemctl disable systemd-resolved.service
sudo rm /etc/resolv.conf
sudo vi /etc/hosts/
<ip> <hostname>
```

d. To configure nameserver, add the IP address of the nameserver to beginning of the resolv.conf

```
sudo vi /etc/resolv.conf
nameserver <nameserver ip>
```

4. Add Docker repo and install Docker.

- **For CentOS**

```
sudo yum install -y yum-utils && sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
sudo yum install -y docker-ce docker-ce-cli containerd.io
sudo systemctl start docker
sudo systemctl enable docker
sudo systemctl status docker
```

- **For Ubuntu**

```
sudo apt-get update
sudo apt-get install ca-certificates curl gnupg
#Add Docker's official GPG key for Ubuntu:
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
sudo chmod a+r /etc/apt/keyrings/docker.gpg
#Use the following command to set up the repository:
echo \
"deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \
"${(. /etc/os-release && echo "$VERSION_CODENAME")}" stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

To install Docker for Ubuntu

```
sudo apt-get update && sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
sudo systemctl start docker
sudo systemctl enable docker
sudo systemctl status docker
```

5. Add the **appviewx** user in the Docker group.

```
sudo groupadd docker
sudo usermod -aG docker appviewx
```

6. Install NTP.

- **For CentOS**

```
sudo yum update -y
sudo yum install ntp -y
sudo systemctl restart ntpd
sudo systemctl enable ntpd
ntpq -np
```

- **For Ubuntu**

```
sudo apt update -y
sudo apt install ntp -y
sudo systemctl restart ntp
sudo systemctl enable ntp
ntpq -np
```

7. Install the additional packages required for the AppViewX Cloud Connector to run.

- **For CentOS**

```
sudo yum install bind-utils net-tools telnet tcpdump curl -y
```

- **For Ubuntu**

```
sudo apt install bind9-utils net-tools telnet tcpdump curl -y
```

8. Limit the maximum number of processes that the **appviewx** user can create.

```
sudo vi /etc/security/limit.conf

appviewx soft nproc 65536
appviewx hard nproc 65536
appviewx soft nofile 65536
appviewx hard nofile 65536
```

9. Allow the **appviewx** user to run **sudo root**.

```
sudo vi /etc/sudoers

appviewx ALL=(ALL) ALL
```

10. To download the required packages, enable the AppViewX repository.



**Note:** This step is not applicable for Amazon Linux 2.



**Note:** If access to the internet is restricted, whitelist the AppViewX repository to perform OS patching.

#### • For CentOS

```
mv /etc/yum.repos.d /etc/yum.repos.d_backup
mkdir -p /etc/yum.repos.d
mv appviewx.repo /etc/yum.repos.d/appviewx.repo
yum clean all
yum update
```

#### • For Ubuntu

- a. Create a file, **appviewx\_repo.list**, and open it for editing.

```
vi /etc/apt/sources.list.d/appviewx_repo.list
```

- b. Add the following to the **appviewx\_repo.list** file.

```
#appviewx apt repo
deb https://repos.appviewx.com/repository/ubuntu focal universe
deb https://repos.appviewx.com/repository/ubuntu focal-updates universe
deb https://repos.appviewx.com/repository/ubuntu focal multiverse
deb https://repos.appviewx.com/repository/ubuntu focal-updates multiverse
deb https://repos.appviewx.com/repository/ubuntu focal-backports main restricted universe multiverse
deb https://repos.appviewx.com/repository/ubuntu focal-security main restricted
deb https://repos.appviewx.com/repository/ubuntu focal-security universe
deb https://repos.appviewx.com/repository/ubuntu focal-security multiverse
```

- c. Create a file, **auth.conf** and open it for editing.

```
vi /etc/apt/auth.conf
```

- d. Add the following to the **auth.conf** file.

```
machine repos.appviewx.com
login ereq2aYRsD4rR5KjD9H2wN4CBuwC7uVpkFaMq
password zstGbbn7wBvHTWkREuTYpeL8fVgJEZxkJtsJM
```

11. Create a new directory at the location **/home/appviewx** and name it **cc-installer**.
12. Ensure that the AppViewX Cloud Connector can establish connectivity with the AppViewX SaaS server endpoints over HTTPS (port 443).

- a. To verify connectivity with the AppViewX SaaS servers, use the **cURL** utility.

```
curl -k --max-time 20 --connect-timeout 20 -s -o /dev/null -w "%{http_code}" "<https://AppViewX SaaS server
URL>>/socket.io/?EIO=3&transport=polling&t=O11wka_"
```

If connectivity has been established successfully, the command will return the HTTP code **200**. If the command returns any other code, it indicates that connectivity is not established.

## 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.

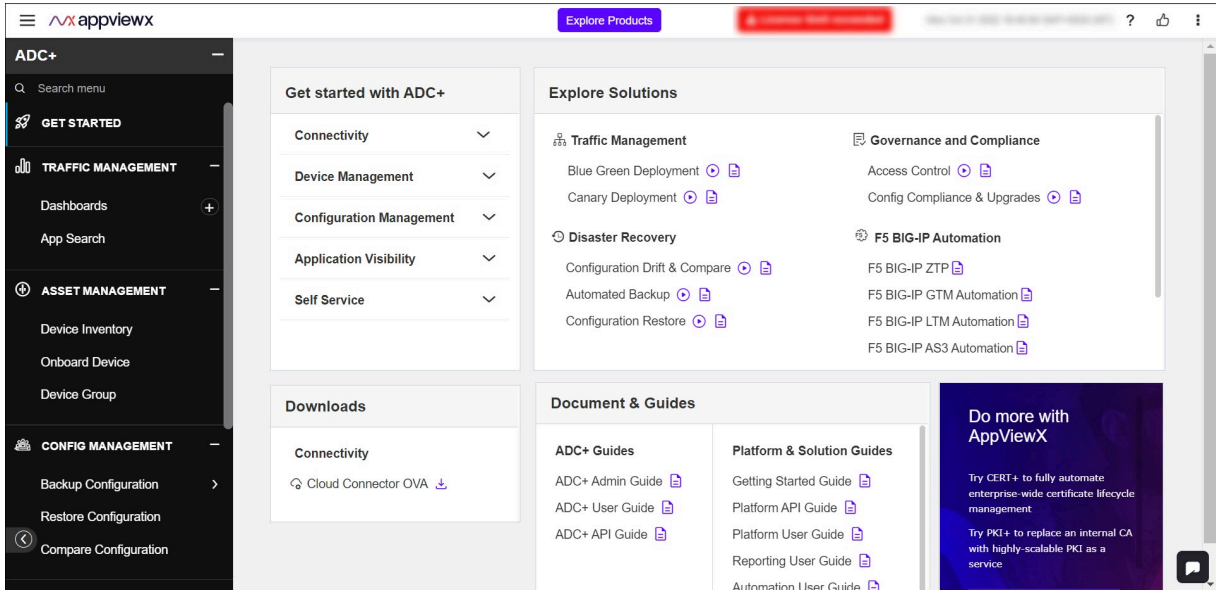


**Important:** As an additional layer of security, AppViewX issues client certificates to access the AppViewX GUI. The client certificate will be made available as part of the onboarding process. Upload this client certificate to the browser to start accessing the product.

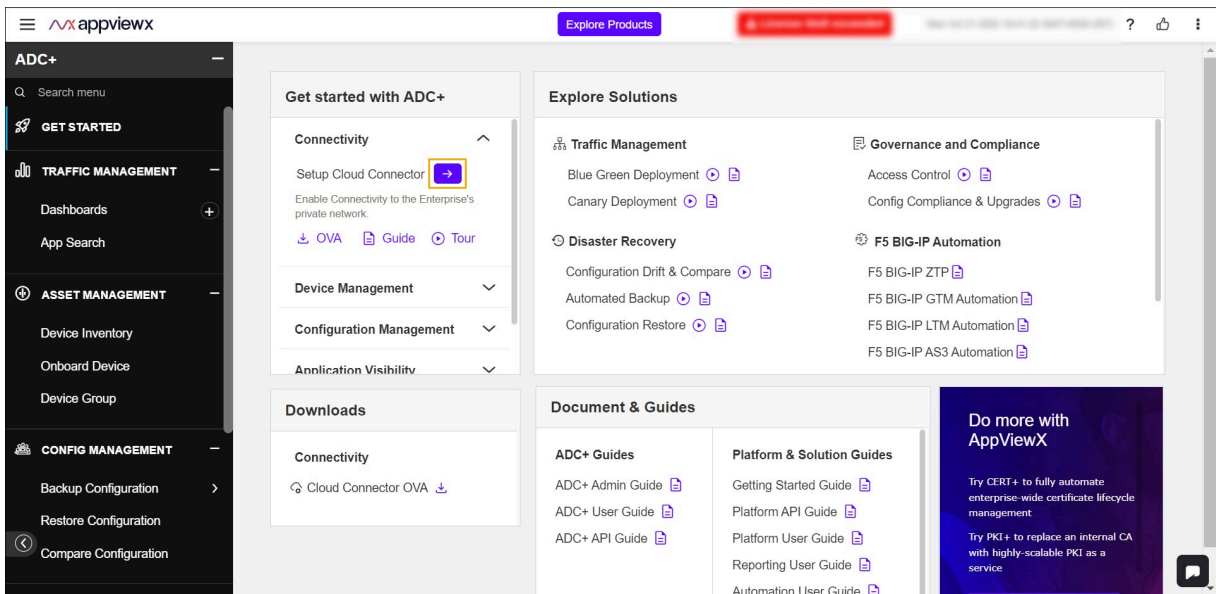
1. Enter your SaaS account URL (for example, <https://tenant-name.appvx.com/appviewx/login>) in the address bar of your browser.  
The AppViewX SaaS login page is displayed.
2. Using the credentials sent as part of the Welcome email, login to the AppViewX SaaS.  
The AppViewX SaaS landing page is displayed.



**Note:** The landing page differs based on a selected AppViewX SaaS product. The following image shows ADC+ landing page.



3. Click  located beside the SetUp Cloud Connector under the Connectivity section.



Redirected to the Settings :: Cloud Connector page.

### Step 3: Executing the Prerequisite Check Script

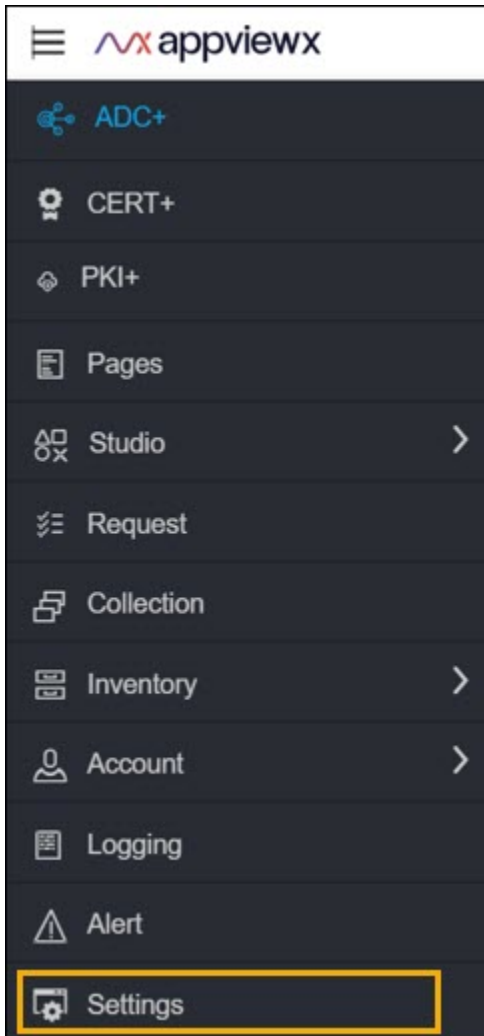
To simplify compliance to the AppViewX Cloud Connector installation prerequisites, you can execute a script to identify and rule out any deviations from the prerequisites.

To perform a prerequisite check:

1. After successfully logging in to the AppViewX SaaS GUI, from the top left corner of the screen, click

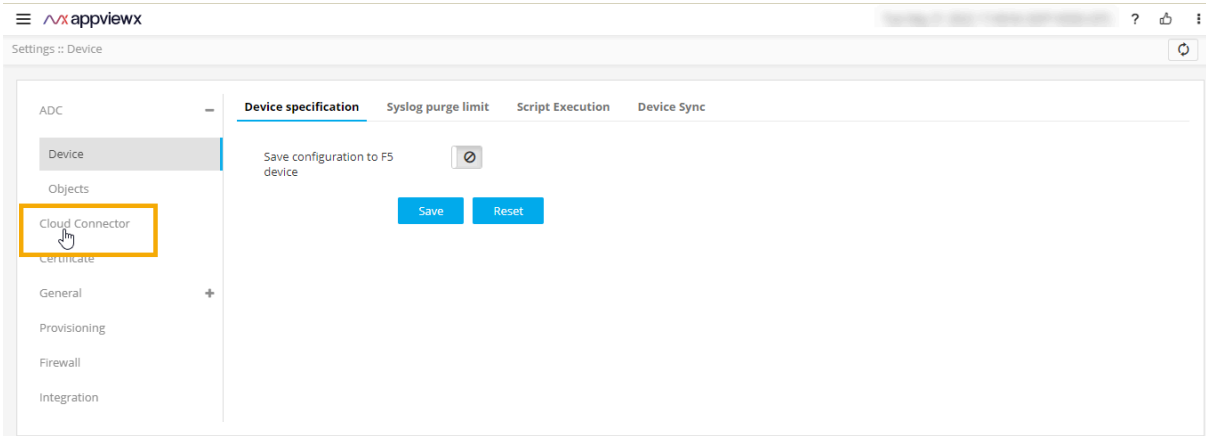


2. From the menu displayed, click **Settings**.



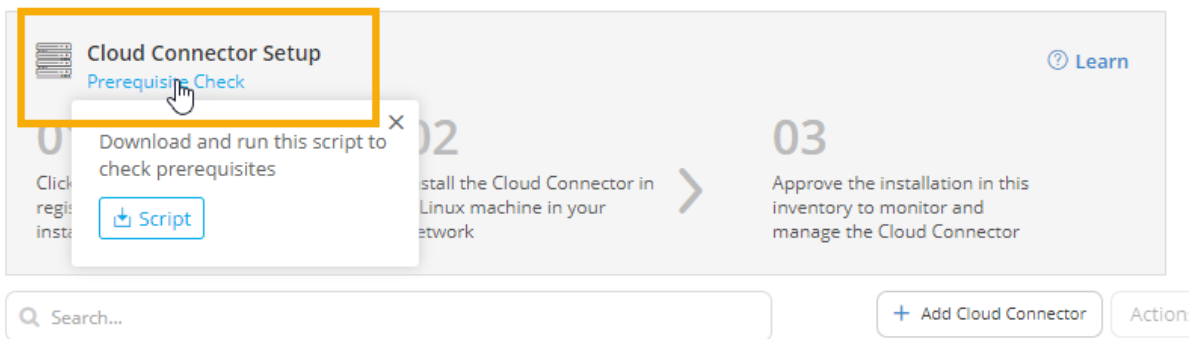
By default, the **Settings :: Device** page is displayed.

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

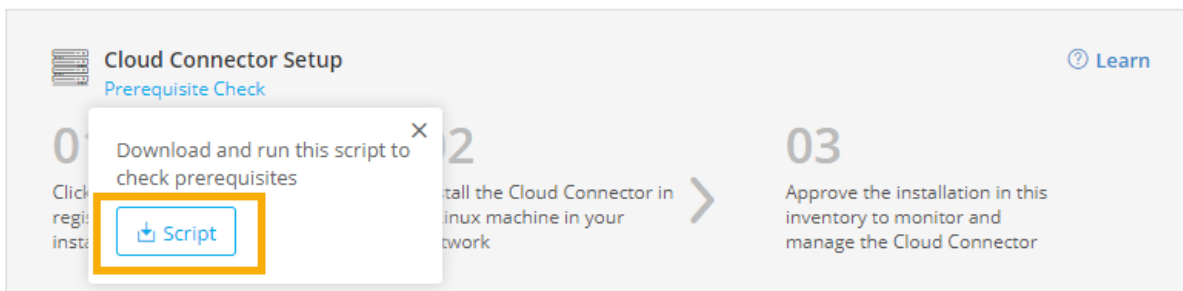


The **Settings :: AppViewX Cloud Connector** page is displayed.

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



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



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

- Securely copy the **pre-requisite-check.sh** via SCP/SFTP to the host machine where the AppViewX Cloud Connector is to 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.

```

root@server1: ~# ./pre-requisite-check.sh
root@server1: ~# ./pre-requisite-check.sh
*
*                               Performing the initial checks...                               *
*****
Proxy configuration details
No HTTP proxy set.
No HTTPS proxy set.

Using system proxy settings...
Performing firewall daemon check
0
Performing connectivity check...
Connection to AppViewX cloud: 20.10.7.100:443 is OK
Performing docker check...
Docker version 20.10.7, build f0df350
Docker is installed.
Docker version check OK
Docker is running...
Performing architecture check...
The architecture check OK
Performing disk check...
Disk space check Ok
Performing memory check...

      ErrorCode      : CC_CONF_005
      ErrorMessage    : Insufficient memory (Free memory: 1335m)
      Operation       : Memory check
      Probable causes : 1. Available primary memory is less
      Suggested remediation : 1. Required RAM specification: 4gb
root@server1: ~#

```



**Note:** For resolutions to the prerequisite check failure scenarios, click [here](#).

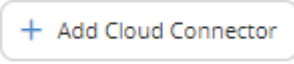
## Step 4: Installing the AppViewX Cloud Connector

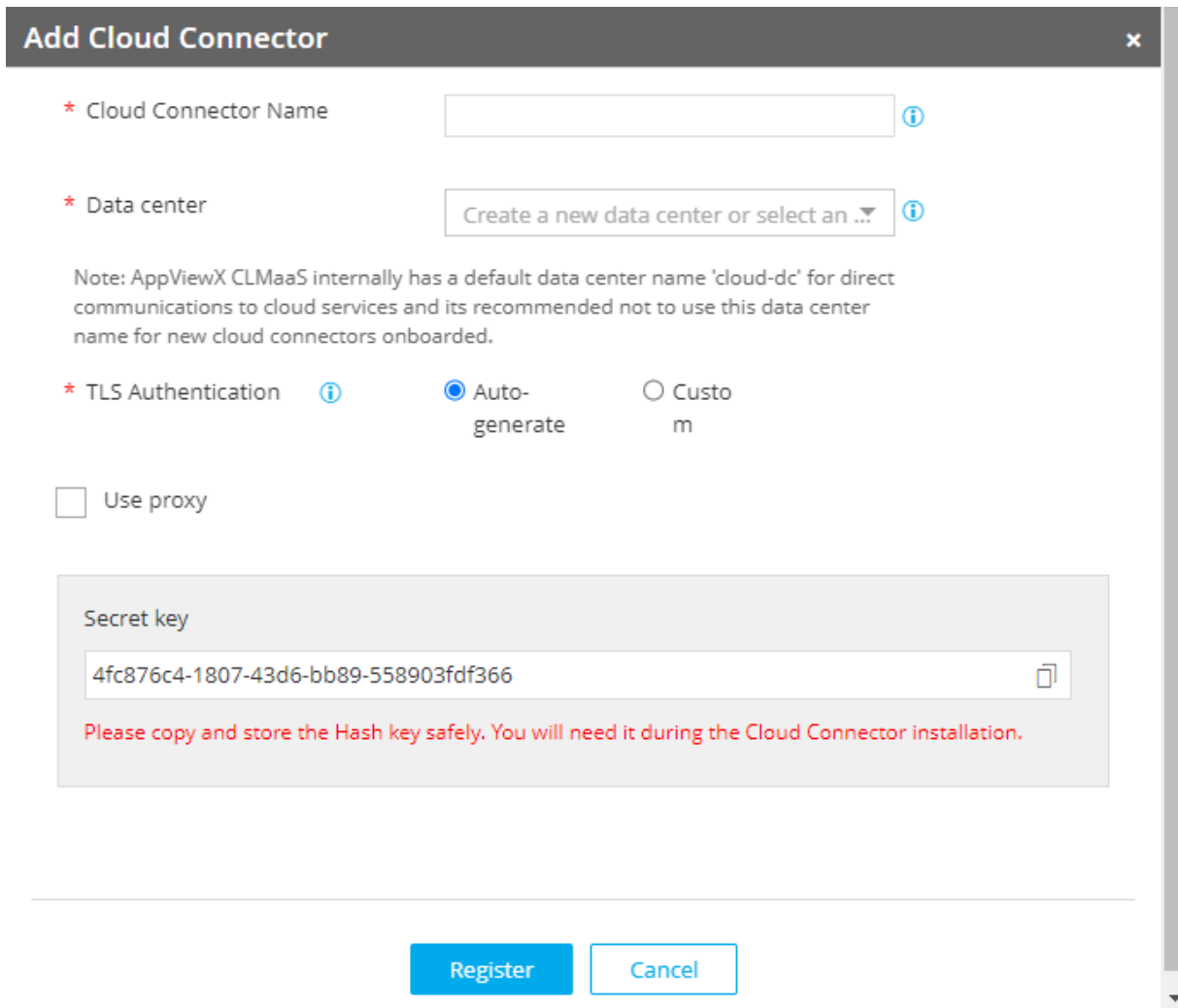
The process of deploying the AppViewX Cloud Connector involves the steps listed below. The substeps for each of these steps are outlined in the subsequent sections.

- [Downloading the Installer](#)
- [Installing the AppViewX Cloud Connector Agent](#)
- [Reviewing the Installation](#)

## 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.



**Add Cloud Connector** ×

\* Cloud Connector Name  ⓘ


\* Data center  ⓘ

Note: AppViewX CLMaaS internally has a default data center name 'cloud-dc' for direct communications to cloud services and its recommended not to use this data center name for new cloud connectors onboarded.

\* TLS Authentication ⓘ  Auto-generate  Custom





Use proxy




Secret key




Please copy and store the Hash key safely. You will need it during the Cloud Connector installation.

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

Field	Description
<b>Cloud Connector Name*</b>	FQDN of the machine where the AppViewX Cloud Connector is to be installed
<b>Data center*</b>	<p>Name of the data center where the AppViewX Cloud Connector is to be installed</p> <div data-bbox="440 495 1419 716" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px;"> <p> <b>Note:</b> The AppViewX SaaS, internally, has a default data center named <b>cloud-dc</b>, for direct communications to cloud services. It is recommended that you do not this data center name for new the AppViewX Cloud Connectors onboarded.</p> </div>
<b>TLS Authentication</b>	<ul style="list-style-type: none"> <li>To auto-generate a TLS certificate, select <b>Auto-generate</b> (default selection). Automatically, the certificate is generated using the AppViewX CA.</li> </ul> <div data-bbox="461 884 1419 1188" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px;"> <p> <b>Note:</b> The created certificate is available in the certificate inventory. You can:</p> <ul style="list-style-type: none"> <li>Assign this certificate to a certificate group</li> <li>Configure a certificate expiry alert for this certificate group from the <b>Server Certificate</b> dashboard, using the <b>Certificate Summary Report</b> widget settings</li> </ul> </div> <ul style="list-style-type: none"> <li>To enter details of a custom TLS certificate, select <b>Custom</b>. The <b>TLS Certificate Password</b> and <b>Custom TLS Certificate</b> fields are displayed. The instructions for filling these fields are given below.</li> </ul>
<b>TLS Certificate Password*</b>	<div data-bbox="440 1430 1419 1556" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px;"> <p> <b>Note:</b> This field is displayed only if you have selected to enter details of a Custom TLS certificate in the <b>TLS Authentication</b> field.</p> </div> <p>Password of the TLS certificate (that will be uploaded in the next step)</p> <div data-bbox="440 1661 1419 1787" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px;"> <p> <b>Note:</b> This is a mandatory field if a Custom TLS certificate is uploaded. AppViewX supports only password-protected Custom TLS certificates.</p> </div>

Field	Description
<b>Custom TLS Certificate</b>	<div data-bbox="440 296 1419 430" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-bottom: 10px;">  <b>Note:</b> This field is displayed only if you have selected to enter details of a Custom TLS certificate in the <b>TLS Authentication</b> field. </div> <p>To upload a custom TLS certificate:</p> <ol style="list-style-type: none"> <li>a. To navigate to the location of the custom TLS certificate, click within the field.</li> <li>b. Select the certificate file.</li> <li>c. Click <b>Open</b>.</li> <li>d. To upload the custom TLS certificate selected, click <b>Upload</b>.</li> </ol> <div data-bbox="440 814 1419 949" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;">  <b>Note:</b> AppViewX supports only password-protected Custom TLS Certificates. </div>
<b>Use proxy</b>	<p>A proxy server is required if the AppViewX Cloud Connector is unable to connect to your endpoints available in the internet.</p> <p>To use a proxy server for the deployment:</p> <ol style="list-style-type: none"> <li>a. Select the <b>Use proxy</b> checkbox.</li> <li>b. To select a preconfigured proxy (for the selected data center), from the <b>Select Proxy</b> dropdown list, select a proxy server.</li> </ol> <p><b>OR</b></p> <p>To create a new proxy server setting:</p> <ol style="list-style-type: none"> <li>a. Use the <a href="#">Click here</a> option shown below the <b>Select Proxy</b> dropdown list.</li> <li>b. For steps to create a new proxy server setting, click <a href="#">here</a>.</li> </ol>
<b>Secret key</b>	<p>A unique key for the AppViewX Cloud Connector installation</p> <p>Click  to copy this key and save it in a safe place.</p>

Field	Description
	 <b>Note:</b> Ensure that the secret key is copied and stored safely before you click <b>Register</b> . If you don't, the key is lost forever. For security reasons, the key is not stored within the product.

- 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](#).

Cloud Connector Name	Status	Data Center	Strict Routing	Version	View Log	Action	TLS Certificate	Proxy	Last Heartbeat	Registered On	SHA256 Checksum
aa-aa	Waiting for response	CBE	<input type="checkbox"/>	22.1.0.5	Upgrade	View	Approve	Reject aa-aa	05/27/2022 19:30	299d17edcb377a52fc45...	
aaa.ccc.com	Waiting for response	CBE	<input type="checkbox"/>	22.1.0.5	Upgrade	View	Approve	Reject aaa-ccc.com	05/27/2022 18:48	721ecc30a419c35af17a7...	

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

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

- In the **Confirmation Message** dialog box, click **Yes**.

The download progress is shown using a progress bar.

## Installing the AppViewX Cloud Connector Agent



**Note:** The following steps assume that:

- All system prerequisites are fulfilled by the host machine.
- The AppViewX Cloud Connector installer (downloaded in the above step) is securely copied via SCP/SFTP to the host machine where the AppViewX Cloud Connector is to be installed.
- The Secret Key received is kept handy for installation.

- 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`

- 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.

3. 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:
kubectf 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

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

 21.1.0.1 .



**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.

## 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 5: Prerequisites for Managing ADC Devices

- [Managing F5 and A10 Devices through the AppViewX Cloud Connector](#)
- [F5 BIG-IP Devices](#)
- [A10 Devices](#)

## Managing F5 and A10 Devices through the AppViewX Cloud Connector

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

### 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 **Platform User Guide**.

# Chapter 6: Installing the AppViewX Windows Gateway

To integrate Microsoft IIS servers and Microsoft CAs with the AppViewX SaaS, you will need to install the AppViewX Windows Gateway.

To do this, from the AppViewX CERT+ SaaS landing page, click:

- **Download AppViewX Windows Gateway** to download the gateway setup
- **AppViewX Windows Gateway Setup Guide** to install and manage the gateway

# Chapter 7: Troubleshooting the AppViewX Cloud Connector

- [Managing Certificates on F5 BIG-IP Devices](#)
- [AppViewX Cloud Connector Health](#)
- [Connectivity Checks](#)
- [Installation Errors](#)
- [Log Analysis](#)
- [Steps to check pod status](#)
- [Steps to restart 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 SaaS.

## 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 SaaS.

## 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: <ul style="list-style-type: none"> <li>• x86 64 bit</li> </ul>
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.	<ul style="list-style-type: none"> <li>• Install Docker with non sudo access.</li> <li>• Required configuration: version 20.10.5 or higher</li> <li>• For instructions for installing the Docker Engine, click <a href="#">here</a>.</li> <li>• For post-installation steps for Linux:, click <a href="#">here</a>.</li> <li>• 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 <a href="#">here</a>.</li> </ul>
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	<ol style="list-style-type: none"> <li>To check the Docker status, execute <b>one</b> of the following commands: <ul style="list-style-type: none"> <li><code>service docker status</code></li> <li><code>systemctl status docker</code></li> </ul> </li> <li>To start the Docker, execute <b>one</b> of the following commands: <ul style="list-style-type: none"> <li><code>service docker start</code></li> <li><code>systemctl start docker</code></li> </ul> </li> <li>Ensure that the Docker is accessible to non sudoers. <ul style="list-style-type: none"> <li>For post-installation steps for Linux: <a href="https://docs.docker.com/engine/install/linux-postinstall/">https://docs.docker.com/engine/install/linux-postinstall/</a></li> <li>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 <a href="#">here</a>.</li> </ul> </li> </ol>
CC_CONF_008	Cluster already exists.	<ol style="list-style-type: none"> <li>Uninstall the AppViewX Cloud Connector.</li> <li>Reinstall the AppViewX Cloud Connector in the same/different node.</li> </ol>
CC_CONF_009	firewalld is running	<ul style="list-style-type: none"> <li>Execute the following script to open the port in firewalld that requires sudo access: <pre>./deps/utils/open-ips-ports-firewalld.sh</pre> </li> </ul> <p><b>OR</b></p> <ul style="list-style-type: none"> <li>Execute the following commands: <pre>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</pre> </li> </ul>

Error Code	Error Message	Resolution
		<pre> 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= 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 </pre>
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	Update the storage driver to <b>overlay2</b> . For setup instructions, click <a href="#">here</a> .
CC_CONF_012	A default route is not available in the tenant premises or the tenant is not connected to a network.	<ul style="list-style-type: none"> <li>• Add a default route with an IP address.</li> <li>• Ensure that the network connection is up.</li> </ul>
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).

Error Code	Error Message	Resolution
		<ul style="list-style-type: none"> <li>• If the cause of failure is insufficient disk and/or memory space, direct the tenant to free disk and/or memory.</li> <li>• If not, reinstall the Cloud Connector.</li> </ul>



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

## Steps to check pod status

- Navigate to `deps/tools/`
- Execute the command `./k3s kubectl get pods -n cc`

```
-bash-4.2$ cd deps/tools/
-bash-4.2$ pwd
/home/appviewx/Dec8/deps/tools
-bash-4.2$ ./k3s kubectl get pods -n cc
NAME                                READY   STATUS    RESTARTS   AGE
avx-mid-server-starter-5b8c7d4c49-bhhsq  1/1     Running   0           2d
avx-mid-server-platform-5754947f99-hx7q6  1/1     Running   0           2d
-bash-4.2$
```

### Expected result:

There should be 2 pods `avx-mid-server-starter` and `avx-mid-server-platform` and both should be in running state.

### Pod Description:

`avx-mid-server-starter`: Responsible for startup of the CC, this downloads the required artifacts from AppViewX servers to CC nodes during startup. Post successful startup, this pod does not play any significance. Restarting the pod would download the artifacts once again and upgrade to the latest cloud connector changes from the server.

`avx-mid-server-platform`: Responsible for all device communication, it checks with AppViewX SaaS servers, and if there are any actions that need to be performed and if it finds that actions have to be performed (for example **Discovery**, **Device Addition**, **Certificate push**, and so on), then the commands will be executed on the end device from this pod and response will be sent back to AppViewX servers.

## Steps to restart pods

### Restart specific pod:

1. Navigate to `deps/tools/`
2. Execute the command `./k3s kubectl delete pods -n cc <podname> --force`

```
-bash-4.2$ ./k3s kubectl get pods -n cc
NAME                                READY   STATUS    RESTARTS   AGE
avx-mid-server-starter-5b8c7d4c49-bhhsq  1/1     Running   0           2d
avx-mid-server-platform-5754947f99-hx7q6  1/1     Running   0           2d
-bash-4.2$ ./k3s kubectl delete pods -n cc avx-mid-server-platform-5754947f99-hx7q6 --force
warning: Immediate deletion does not wait for confirmation that the running resource has been terminated. The resource may continue to run on the cluster indefinitely.
pod "avx-mid-server-platform-5754947f99-hx7q6" force deleted
-bash-4.2$
```

### Restart both starter and platform pods:

1. Navigate to `deps/tools/`
2. Execute the command `./k3s kubectl delete pods -n cc --force --all`

```
-bash-4.2$ ./k3s kubectl delete pods -n cc --force --all
warning: Immediate deletion does not wait for confirmation that the running resource has been terminated. The resource may continue to run on the cluster indefinitely.
pod "avx-mid-server-starter-5b8c7d4c49-bhhsq" force deleted
pod "avx-mid-server-platform-5754947f99-nb6n8" force deleted
-bash-4.2$
```

# Chapter 8: Managing the AppViewX Cloud Connector

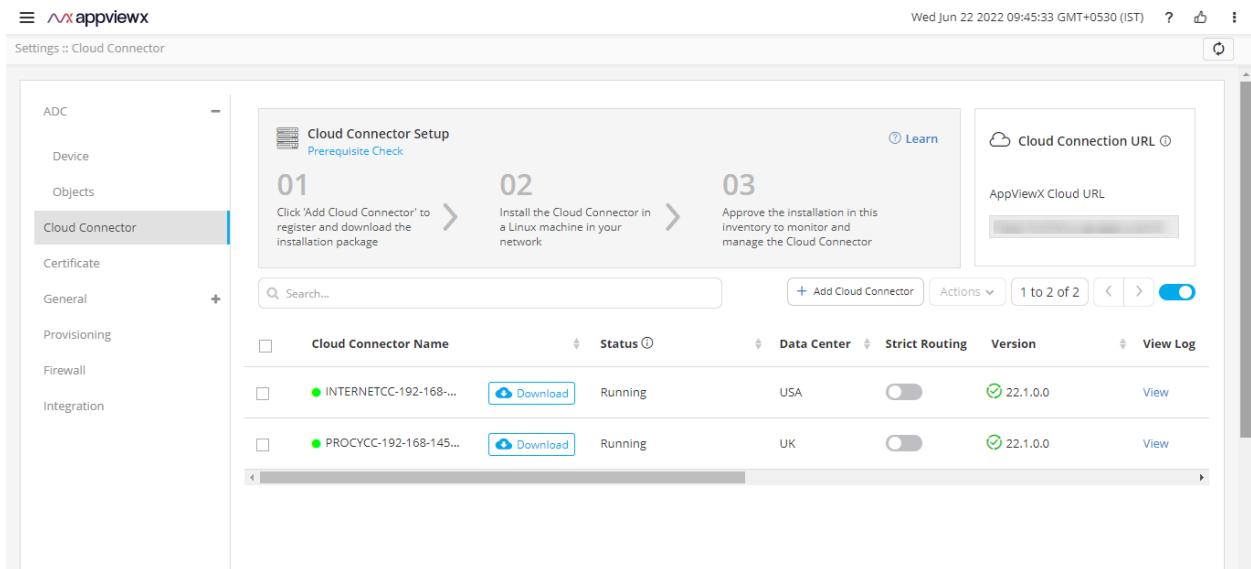
- [Overview](#)
- [Understanding the AppViewX Cloud Connector Inventory](#)
- [AppViewX Cloud Connector Actions](#)
- [Monitoring the Health of the AppViewX Cloud Connector](#)

## Overview





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.



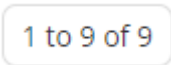

## Understanding the AppViewX Cloud Connector Inventory








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
<p><b>AppViewX Cloud Connector Setup banner</b></p>	<p>The banner summarizes the process of setting up the AppViewX Cloud Connector.</p>
<p><b>Cloud Connection URL</b></p>	<p>AppViewX cloud URL of the server (internal) that hosts the AppViewX instance of the AppViewX Cloud Connector</p>
<p><b>Search</b></p>	<div data-bbox="737 525 1510 588" style="border: 1px solid #ccc; border-radius: 5px; padding: 5px; margin-bottom: 10px;">  Search ...         </div> <p>To search for a AppViewX Cloud Connector entry:</p> <ol style="list-style-type: none"> <li>1. In the <b>Search</b> field, enter the value you want to filter the records for.</li> <li>2. Press <b>Enter</b>.</li> </ol> <p>The <b>Settings :: Cloud Connector</b> page is updated to show details of only those records that match the search criteria.</p>
<p><b>Add AppViewX Cloud Connector</b></p>	<p>To add a new AppViewX Cloud Connector, click</p> <div data-bbox="737 1108 1036 1171" style="border: 1px solid #ccc; border-radius: 5px; padding: 5px; margin-bottom: 10px;">  Add Cloud Connector         </div> <div data-bbox="737 1213 1510 1390" style="border: 1px solid #00a0c0; border-radius: 10px; padding: 10px; background-color: #e6f2f8;">  <b>Note:</b> For detailed steps for adding a new AppViewX Cloud Connector, refer to the <a href="#">Setting Up the AppViewX Cloud Connector</a> section.         </div>
<p><b>Actions</b></p>	<div data-bbox="737 1465 1510 1596" style="border: 1px solid #00a0c0; border-radius: 10px; padding: 10px; background-color: #e6f2f8;">  <b>Note:</b> This button is enabled only when one or multiple AppViewX Cloud Connectors are selected.         </div> <p>AppViewX lets you perform the following actions on a AppViewX Cloud Connector:</p> <ul style="list-style-type: none"> <li>• Start</li> <li>• Pause</li> </ul>

Element/Field	Description
	<ul style="list-style-type: none"> <li>• Upgrade</li> <li>• Update config</li> <li>• Delete</li> </ul> <p>To perform these actions, click .</p> <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> <b>Note:</b> For detailed steps to perform each of the above listed actions, refer to the <a href="#">AppViewX Cloud Connector Actions</a>.</p> </div>
<p><b>Record Count Preference</b></p>	<p>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.</p> <p>To set the record count preference:</p> <ol style="list-style-type: none"> <li>1. Click .</li> <li>2. From the <b>Show</b> menu displayed, select your record count preference (for example, 50 records).</li> </ol> <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <p><b>Show :</b></p> <p>25 Records</p> <p style="background-color: #0070C0; color: white; padding: 2px;">50 Records </p> <p>75 Records</p> <p>100 Records</p> </div> <p>The <b>Settings :: Cloud Connector</b> page is updated according to the record count preference selected. The message, <b>Record count preference saved successfully</b>, is displayed. The UI control is also updated to display the current selection.</p>

Element/Field	Description						
<p><b>Auto Refresh</b></p>	<p>If enabled, the <b>Auto Refresh</b> feature automatically refreshes the AppViewX Cloud Connector inventory details every 5 seconds.</p>  <p>To enable this feature, use the <b>Auto Refresh</b> key.</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p><b>i</b> <b>Tip:</b> 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.</p> </div>						
<p><b>AppViewX Cloud Connector Name</b></p>	<p>This field displays the following two details:</p> <ul style="list-style-type: none"> <li>• Name assigned to the AppViewX Cloud Connector when it is added</li> <li>• <a href="#">Health status of the AppViewX Cloud Connector</a></li> </ul>						
<p><b>Status</b></p>	<p>This field has the following values:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%;">Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td style="vertical-align: top;"><b>Waiting for response</b></td> <td> <p>After the AppViewX Cloud Connector is registered, the package must be downloaded and installed.</p> <p>This status indicates that this installation is pending.</p> </td> </tr> <tr> <td style="vertical-align: top;"><b>Waiting for approval</b></td> <td> <p>After the AppViewX Cloud Connector is installed, the admin must approve/reject the installation by clicking the  /  buttons from the <b>Action</b> field.</p> </td> </tr> </tbody> </table>	Value	Description	<b>Waiting for response</b>	<p>After the AppViewX Cloud Connector is registered, the package must be downloaded and installed.</p> <p>This status indicates that this installation is pending.</p>	<b>Waiting for approval</b>	<p>After the AppViewX Cloud Connector is installed, the admin must approve/reject the installation by clicking the  /  buttons from the <b>Action</b> field.</p>
Value	Description						
<b>Waiting for response</b>	<p>After the AppViewX Cloud Connector is registered, the package must be downloaded and installed.</p> <p>This status indicates that this installation is pending.</p>						
<b>Waiting for approval</b>	<p>After the AppViewX Cloud Connector is installed, the admin must approve/reject the installation by clicking the  /  buttons from the <b>Action</b> field.</p>						

Element/Field	Description								
	<table border="1"> <thead> <tr> <th data-bbox="730 264 867 317">Value</th> <th data-bbox="867 264 1516 317">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="730 317 867 436"></td> <td data-bbox="867 317 1516 436">This status indicates that the admin's response to the installation is pending.</td> </tr> <tr> <td data-bbox="730 436 867 541"><b>Running</b></td> <td data-bbox="867 436 1516 541">The AppViewX Cloud Connector has been approved by the admin and is running.</td> </tr> <tr> <td data-bbox="730 541 867 907"><b>Paused</b></td> <td data-bbox="867 541 1516 907">The AppViewX Cloud Connector has been approved by the admin but is paused.</td> </tr> </tbody> </table> <div data-bbox="873 688 1507 869" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;">  <b>Note:</b> The AppViewX Cloud Connector is paused when it has to undergo maintenance and/or troubleshooting.                 </div>	Value	Description		This status indicates that the admin's response to the installation is pending.	<b>Running</b>	The AppViewX Cloud Connector has been approved by the admin and is running.	<b>Paused</b>	The AppViewX Cloud Connector has been approved by the admin but is paused.
Value	Description								
	This status indicates that the admin's response to the installation is pending.								
<b>Running</b>	The AppViewX Cloud Connector has been approved by the admin and is running.								
<b>Paused</b>	The AppViewX Cloud Connector has been approved by the admin but is paused.								
<b>Data Center</b>	Physical location where the AppViewX Cloud Connector system is hosted								
<b>Strict Routing</b>	To enable <a href="#">strict data center-based routing</a> , turn on the toggle under <b>Strict Routing</b> .								
<b>Version</b>	Version of the AppViewX Cloud Connector platform component <div data-bbox="734 1234 1513 1444" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;">  <b>Note:</b> If a new version of the AppViewX Cloud Connector platform component is available, the  button is displayed for that AppViewX Cloud Connector.                     </div>								
<b>View Log</b>	To view the activity log for a AppViewX Cloud Connector, click <b>View</b> for that AppViewX Cloud Connector.								
<b>Action</b>	This field lets you perform the following actions for a AppViewX Cloud Connector <ul style="list-style-type: none"> <li>• Pause a running AppViewX Cloud Connector</li> <li>• Start a paused AppViewX Cloud Connector</li> </ul>								

Element/Field	Description
	<ul style="list-style-type: none"> <li>• Approve a AppViewX Cloud Connector</li> <li>• Reject a AppViewX Cloud Connector</li> </ul> <p>This field displays an action that can be performed for the AppViewX Cloud Connector, depending on the current status of the AppViewX Cloud Connector.</p> <p>For example, if the connector is running, this field shows the  button.</p> <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> <b>Remember:</b> Only the admin user, with the modify permission, can approve/reject a AppViewX Cloud Connector.</p> </div>
<b>TLS Certificate</b>	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.
<b>Proxy</b>	Details of the proxy server, if it has been used for the deployment
<b>Last Heartbeat</b>	Timestamp of the latest health analysis of the AppViewX Cloud Connector
<b>Registered On</b>	Timestamp of the AppViewX Cloud Connector installation
<b>SHA256 Checksum</b>	<p>Details of the SHA256 token</p> <p>It ensures that the downloaded AppViewX Cloud Connector package is the same as the checksum displayed in the AppViewX Cloud Connector inventory page.</p>

## AppViewX Cloud Connector Actions

Key actions that can be performed include:

- [Starting the AppViewX Cloud Connector](#)
- [Pausing the AppViewX Cloud Connector](#)
- [Upgrading the AppViewX Cloud Connector Version](#)
- [Updating the Certificate Configuration](#)
- [Deleting an AppViewX Cloud Connector Instance](#)
- [Uninstalling an AppViewX Cloud Connector Instance](#)

## Starting the AppViewX Cloud Connector

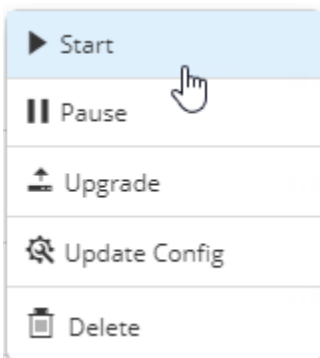
After the admin user has approved its installation, you need to ‘start’ the AppViewX Cloud Connector—you need to enable the AppViewX Cloud Connector to route traffic between the internal network and the AppViewX cloud.

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 .

4. From the menu displayed, select **Start**.



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

## Pausing the AppViewX Cloud Connector

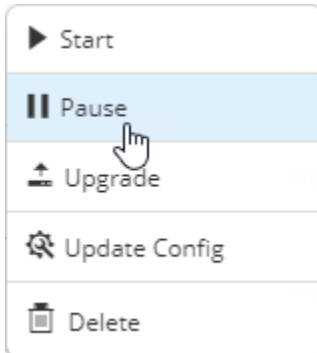
The AppViewX Cloud Connector can be paused for regular maintenance or troubleshooting. Pausing the AppViewX Cloud Connector will pause all activities that have to be performed in the network premises—for example, discovering and scanning certificates, accessing endpoints within the network, and so on.

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  .

4. From the menu displayed, select **Pause**.



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

## Upgrading the AppViewX Cloud Connector Version


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  button is displayed for that AppViewX Cloud Connector.

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

3. Click  .

4. From the menu displayed, select **Upgrade**.



- When the upgrade starts, the **Upgrade in progress** status is displayed.
- When the AppViewX Cloud Connector version is upgraded, it is indicated by the  symbol prefixed to the version number.
- If the upgrade fails:
  - The **Upgrade** button is displayed again.
    - When an upgrade fails, internally, AppViewX attempts to rollback to the previous successful version of the AppViewX Cloud Connector. When the rollback is successful, the **Upgrade** button is displayed again so that you can retry the version upgrade.

**OR**


- The status is updated to **Upgrade failed**.
  - The upgrade is marked as failed if the rollback is unsuccessful. In this case, please reach out to [saashelp@appviewx.com](mailto:saashelp@appviewx.com) for further assistance.

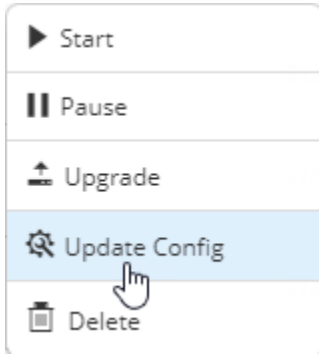


**Note:** Based on the internet bandwidth and the number of cloud connectors being installed, the downloading of the cloud connector may vary between 5 to 15 minutes.

## 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  .
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

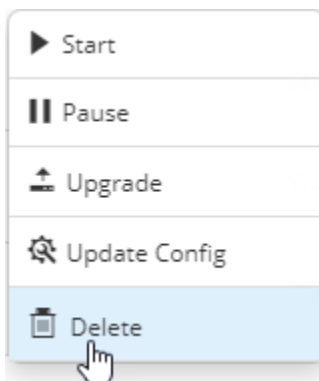
The AppViewX Cloud Connector instance may have to be deleted in events like if there is a fault with the system on which the instance is installed or if it is a faulty installation.

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

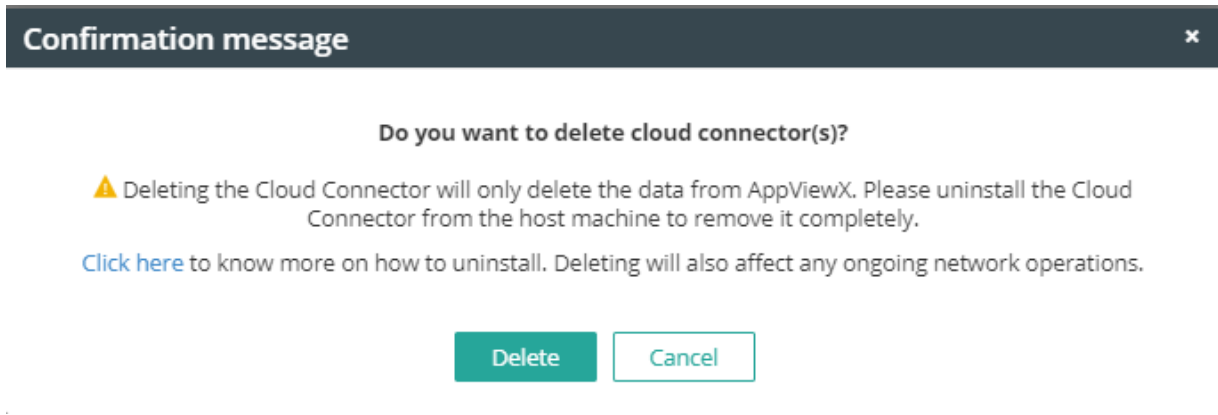
1. Select the checkbox for the AppViewX Cloud Connector you want to delete.

2. Click .

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.

Color of the health indicator	Description
	<p>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.</p> <p>To resolve, refer to the <a href="#">Troubleshooting</a> section.</p>
	<p>The AppViewX Cloud Connector is not receiving traffic.</p> <p>The AppViewX Cloud Connector's health is analyzed for 3 to 5 minutes before it is declared to be down.</p> <p>To resolve, refer to the <a href="#">Troubleshooting</a> section.</p>

For details on how the health indicators are displayed, refer to the [Understanding the AppViewX Cloud Connector Inventory](#) page.

# Chapter 9: Frequently Asked Questions

- [Disabling firewalld](#)
- [Docker Prerequisites](#)
- [Monitoring the Health of the AppViewX Cloud Connector](#)
- [Steps to check pod status](#)
- [Steps to restart pods](#)
- [Deploying the AppViewX OVA](#)
- [Updating the AppViewX Virtual Image from the AppViewX Repository](#)
- [Synchronizing the Node Clock with the Network Time](#)
- [Validating the SHA256 Checksum](#)

## Disabling firewalld

- 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`

## 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/>






**Important:** 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](#).

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

## 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 <a href="#">Troubleshooting</a> 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 <a href="#">Troubleshooting</a> section.

For details on how the health indicators are displayed, refer to the [Understanding the AppViewX Cloud Connector Inventory](#) page.

## Steps to check pod status

- Navigate to `deps/tools/`
- Execute the command `./k3s kubectl get pods -n cc`

```
-bash-4.2$ cd deps/tools/
-bash-4.2$ pwd
/home/appviewx/Dec8/deps/tools
-bash-4.2$ ./k3s kubectl get pods -n cc
NAME                                READY   STATUS    RESTARTS   AGE
avx-mid-server-starter-5b8c7d4c49-bhhsq  1/1     Running   0           2d
avx-mid-server-platform-5754947f99-hx7q6  1/1     Running   0           2d
-bash-4.2$
```

### Expected result:

There should be 2 pods `avx-mid-server-starter` and `avx-mid-server-platform` and both should be in running state.

### Pod Description:

`avx-mid-server-starter`: Responsible for startup of the CC, this downloads the required artifacts from AppViewX servers to CC nodes during startup. Post successful startup, this pod does not play any significance. Restarting the pod would download the artifacts once again and upgrade to the latest cloud connector changes from the server.

`avx-mid-server-platform`: Responsible for all device communication, it checks with AppViewX SaaS servers, and if there are any actions that need to be performed and if it finds that actions have to be performed (for example **Discovery**, **Device Addition**, **Certificate push**, and so on), then the commands will be executed on the end device from this pod and response will be sent back to AppViewX servers.

## Steps to restart pods

### Restart specific pod:

1. Navigate to `deps/tools/`
2. Execute the command `./k3s kubectl delete pods -n cc <podname> --force`

```

-bash-4.2$ ./k3s kubectl get pods -n cc
NAME                                READY   STATUS    RESTARTS   AGE
avx-mid-server-starter-5b8c7d4c49-bhhsq  1/1     Running   0           2d
avx-mid-server-platform-5754947f99-hx7q6  1/1     Running   0           2d
-bash-4.2$ ./k3s kubectl delete pods -n cc avx-mid-server-platform-5754947f99-hx7q6 --force
warning: Immediate deletion does not wait for confirmation that the running resource has been terminated. The resource may continue to run on the cluster indefinitely.
pod "avx-mid-server-platform-5754947f99-hx7q6" force deleted
-bash-4.2$

```

## Restart both starter and platform pods:

1. Navigate to `deps/tools/`
2. Execute the command `./k3s kubectl delete pods -n cc --force --all`

```

-bash-4.2$ ./k3s kubectl delete pods -n cc --force --all
warning: Immediate deletion does not wait for confirmation that the running resource has been terminated. The resource may continue to run on the cluster indefinitely.
pod "avx-mid-server-starter-5b8c7d4c49-bhhsq" force deleted
pod "avx-mid-server-platform-5754947f99-nb6n8" force deleted
-bash-4.2$

```

## Deploying the AppViewX OVA

The AppViewX Virtual Image is an OVA that is bundled with the [software](#), [network](#), and [Docker](#) prerequisites for installing the AppViewX Cloud Connector without altering the OS configuration on their systems. (The `.ova` file that can be downloaded from [here](#)).

Detailed instructions for the OVA deployment are documented are [here](#).

Detailed instructions for updating the AppViewX virtual image from the AppViewX repository are documented [here](#).

## Updating the AppViewX Virtual Image from the AppViewX Repository

The AppViewX virtual image, which is used to [install the AppViewX Cloud Connector](#), can be updated with the latest OS-level updates and security patches from the AppViewX repository.

- [Prerequisites for Updating the Virtual Image from the Repository](#)
- [Updating the Virtual Image from the Repository](#)

### Prerequisites for Updating the Virtual Image from the Repository

- For the AppViewX nodes, access to the following URL: <https://repos.appviewx.com>
- Root/sudo access to configure `yum`.

## Updating the Virtual Image from the Repository

1. From the terminal, login as the root user to the AppViewX Cloud Connector OVA.
2. To get the latest updates from the AppViewX repository, execute the `yum update` command, as shown in the image below:

```
[root@pesrv05-devops07-95-141 ~]# yum update
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
base | 2.2 kB 00:00:00
centosplus | 1.5 kB 00:00:00
epel | 3.3 kB 00:00:00
extras | 1.5 kB 00:00:00
updates | 1.5 kB 00:00:00
(1/6): epel/x86_64/updateinfo | 1.0 MB 00:00:02
(2/6): extras/7/x86_64/primary | 98 kB 00:00:02
(3/6): centosplus/7/x86_64/primary | 689 kB 00:00:05
(4/6): updates/7/x86_64/primary | 1.4 MB 00:00:09
(5/6): epel/x86_64/primary | 3.8 MB 00:00:15
(6/6): base/7/x86_64/primary | 2.9 MB 00:00:17
base 10072/10072
centosplus 34/34
epel 13470/13470
extras 448/448
updates 293/293
```

## Synchronizing the Node Clock with the Network Time

- 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
```

## Validating the SHA256 Checksum

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>
```