AppViewX SaaS Onboarding and Getting Started Guide

Version: 2022.1.0

Copyright AppViewX, Inc.

Trademarks

External Reference Links

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

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

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

Contact Information

AppViewX, Inc.

222 Broadway, FL 19

New York, NY 10038

Email: info@appviewx.com

Web: www.appviewx.com

Contents

Preface....v

Revision History.... v

About this Guide v

Audience....v

Text Conventions.... v

Chapter 1. Key Highlights of AppViewX Software as a Service....6

Chapter 2. Introduction to the the AppViewX Cloud Connector....8

The AppViewX Cloud Connector....8

Features of the AppViewX Cloud Connector....9

Data Center-based Routing....9

Cloud Connector High Availability.... 11

Custom Certificates for Core Communication.... 11

Communication Authentication and Encryption....11

Chapter 3. Prerequisites for Setting up AppViewX Cloud Connector.... 12

Prerequisites for Setting Up the Cloud Connector via the Native OS.... 12

Overview.... 12

Hardware....13

Operating System.... 13

Docker Prerequisites....13

Server and Network Prerequisites....14

Executing the Prerequisite Check Script....16

Prerequisites for Setting Up the Cloud Connector using the Virtual Image....18

.... 19

Hardware....19

Server and Network Prerequisites....19

Executing the Prerequisite Check Script....21

Chapter 4. Getting Started with the AppViewX Free Trial.... 25

Chapter 5. Signing Up for the Free Trial via the AppViewX Website.....26
Accessing the Website Sign Up Page.....26
Entering Details on the Sign Up Page.....27
Verifying your Email 28
Logging in to your SaaS Account.....28
Getting Started with AppViewX SaaS.....31
Methods to Set up the AppViewX Cloud Connector.....31
Setting up the AppViewX Cloud Connector via the Native OS.....32
Setting up the AppViewX Cloud Connector via a Virtual Image.....47
Working with AppViewX SaaS.....68
Chapter 6. Signing Up for the Free Trial via the AWS Marketplace.....69
Signing Up for the Free Trial via the AWS Marketplace.....69
Accessing the AWS Marketplace Sign Up Page.....69
Entering Details on the Sign Up Page.....73
Verifying your Email 74
Logging in to your SaaS Account.....75
Getting Started with CERT+ SaaS.....78
Methods to Set up the AppViewX Cloud Connector.....78
Setting up the AppViewX Cloud Connector via the Native OS.....79
Setting up the AppViewX Cloud Connector via a Virtual Image.....94
Working with AppViewX SaaS.....115
Chapter 7. Support.....116
Support.....116
Using the AppViewX Chatbot.....116

Preface

Revision History

Revision	Description	Date
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

This guide outlines the steps for onboarding customers to the AppViewX SaaS platform and enables them to get started with the AppViewX SaaS products.

Audience

This guide is intended for customers who want to evaluate the product and services that AppViewX offers for its SaaS customers.

Text Conventions

The following text conventions are used in this document:

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

Chapter 1: 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.

Chapter 2: Introduction to the the AppViewX Cloud Connector

- The AppViewX Cloud Connector
- Features of the AppViewX Cloud Connector

The AppViewX Cloud Connector

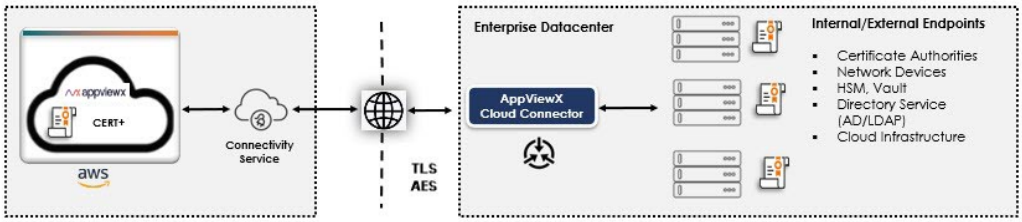
AppViewX Cloud Connector is a lightweight plug-in that establishes connectivity between AppViewX Cloud and the Enterprise Network. The cloud connector serves as a secure channel for communication between AppViewX SaaS and your enterprise network without requiring any complex network or infrastructure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

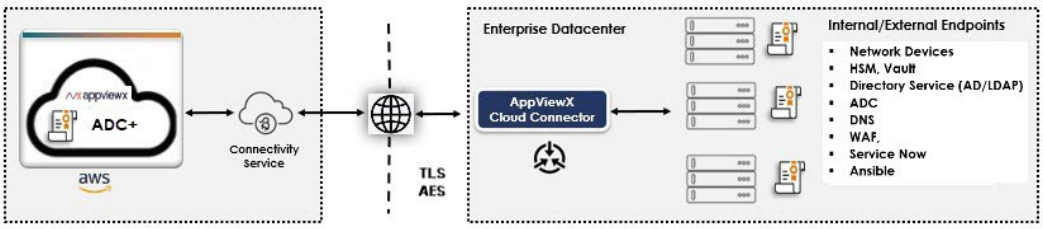
    To enable this, at the time of adding a new AppViewX Cloud Connector instance, from the **Data Center** dropdown list, select **DC Routing**.



- **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)

Features of the AppViewX Cloud Connector

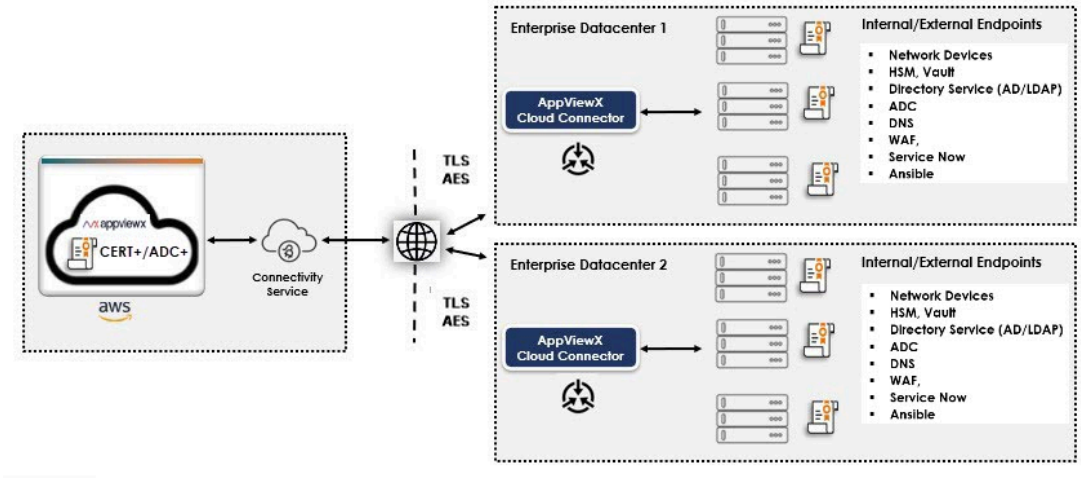
- Data Center-based Routing
- Cloud Connector High Availability
- Custom Certificates for Core Communication
- Communication Authentication and Encryption

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.

Chapter 3: Prerequisites for Setting up AppViewX Cloud Connector

- Prerequisites for Setting Up the Cloud Connector via the Native OS
- Prerequisites for Setting Up the Cloud Connector using the Virtual Image

Prerequisites for Setting Up the Cloud Connector via the Native OS

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.

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

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
- Operating System
- Docker Prerequisites
- Server and Network Prerequisites

Hardware

Each AppViewX Cloud Connector instance requires the following minimum configuration:

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

Operating System

- Ubuntu version 20.04
- CentOS version 7.7 and 7.9

Docker Prerequisites

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

Note: Support for rootless Docker is excluded.

For Docker installation instructions, refer to the links below:

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

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

Server and Network Prerequisites

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

  To do this, execute the following sequence of commands:

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

- Ensure that the AppViewX Cloud Connector can establish connectivity with the AppViewX 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 as given below. When connectivity has been established successfully, the command will return the HTTP code **200**.

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

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

```
apt-get install curl
```

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

```
yum install curl
```

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

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

  ```
  sudo ufw status
  ```

  To permanently disable firewalld, execute the command given below:

  ```
  sudo ufw disable
  ```

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

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

  ```
  sudo systemctl status firewalld --now
  ```

  To permanently disable the firewalld, execute the command given below:

  ```
  sudo systemctl disable firewalld --now
  ```

  To restrict other devices from enabling the firewalld, execute the command given below:

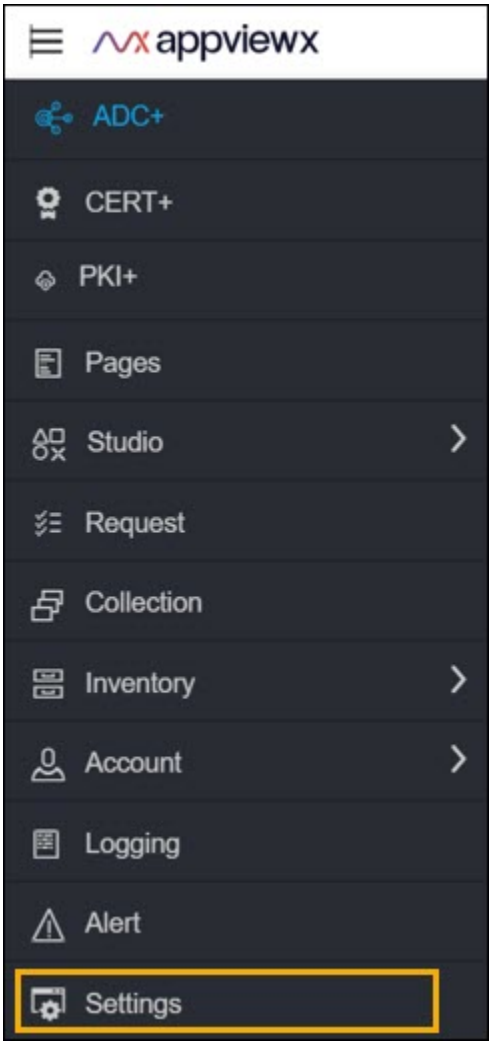
  ```
  sudo systemctl mask firewalld --now
  ```

Executing the Prerequisite Check Script

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

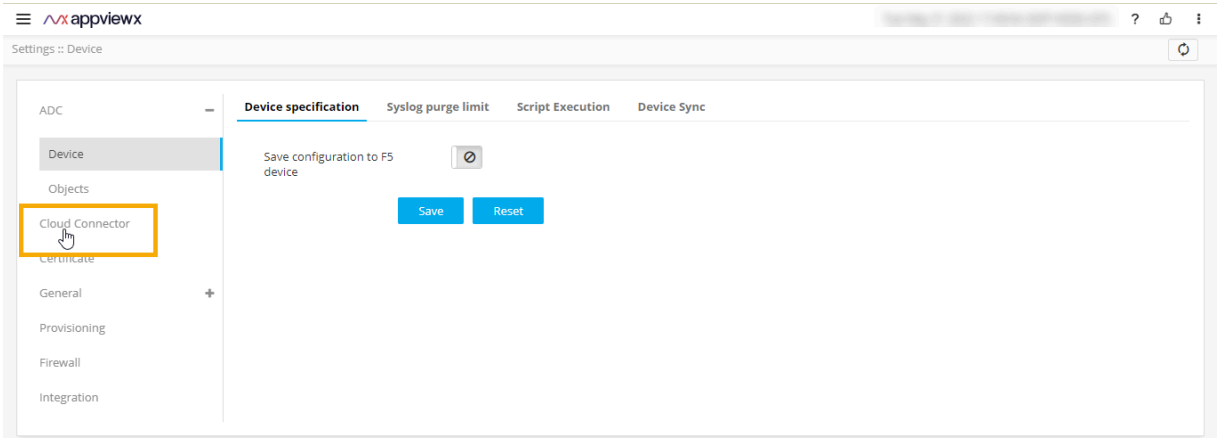
To perform a prerequisite check:

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**.



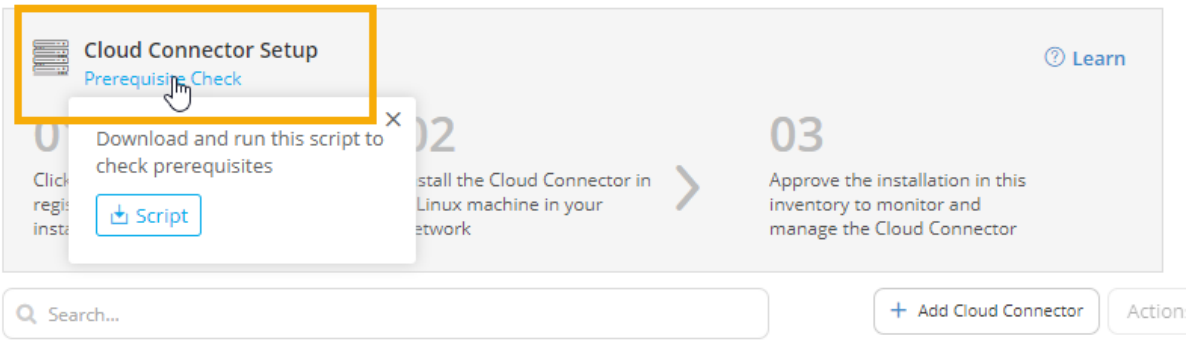
   By default, the **Settings :: Device** page is displayed.

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

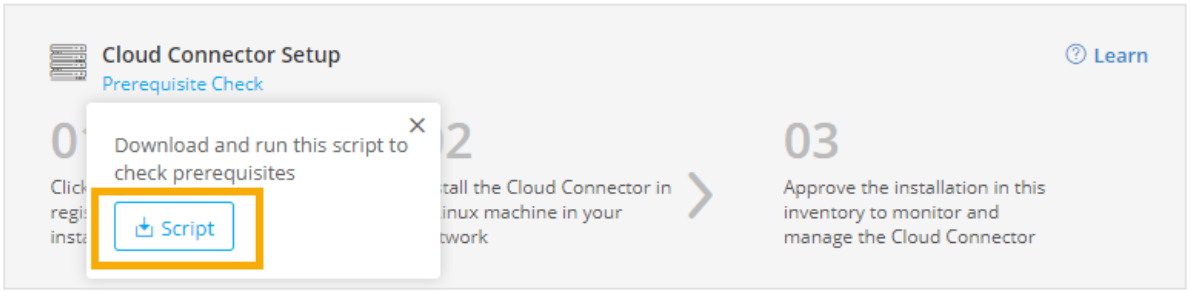


The **Settings :: AppViewX Cloud Connector** page is displayed.

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



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



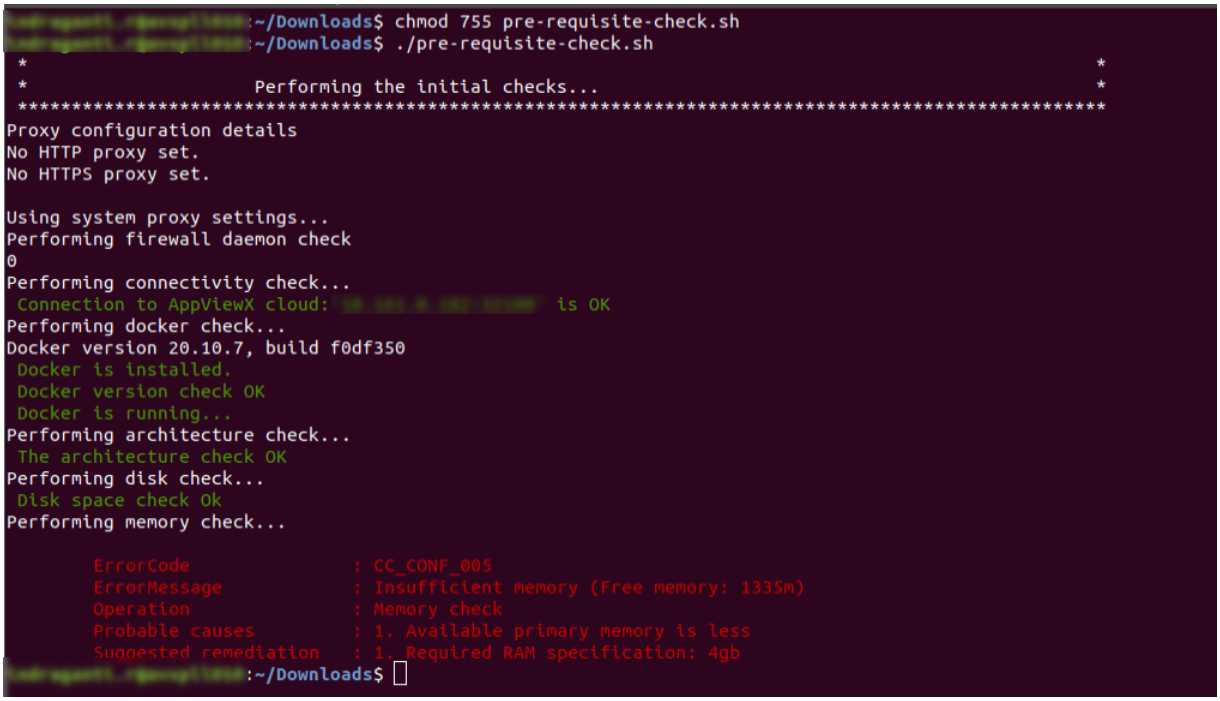
The **pre-requisite-check.sh** script file is downloaded.

6. 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.



> **Note:** For resolutions to the prerequisite check failure scenarios, refer the AppViewX Cloud Connector User Guide.

Prerequisites for Setting Up the Cloud Connector using the Virtual Image

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
- Server and Network Prerequisites
- Executing the Prerequisite Check Script

Hardware

Each AppViewX Cloud Connector instance requires the following minimum configuration:

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

Server and Network Prerequisites

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

  To do this, execute the following sequence of commands:

```
yum install -y ntp

systemctl enable ntp

systemctl start ntp
```

- Ensure that the AppViewX Cloud Connector can establish connectivity with the AppViewX 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 as given below. When connectivity has been established successfully, the command will return the HTTP code **200**.

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

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

```
apt-get install curl
```

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

```
yum install curl
```

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

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

  ```
  sudo ufw status
  ```

  To permanently disable firewalld, execute the command given below:

  ```
  sudo ufw disable
  ```

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

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

```
sudo systemctl status firewalld --now
```

To permanently disable the firewalld, execute the command given below:

```
sudo systemctl disable firewalld --now
```

To restrict other devices from enabling the firewalld, execute the command given below:

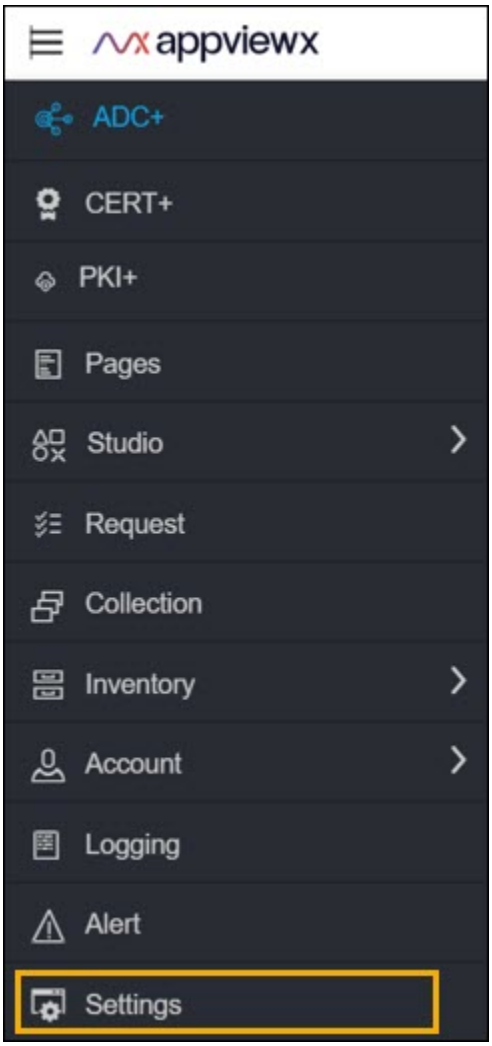
```
sudo systemctl mask firewalld --now
```

Executing the Prerequisite Check Script

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

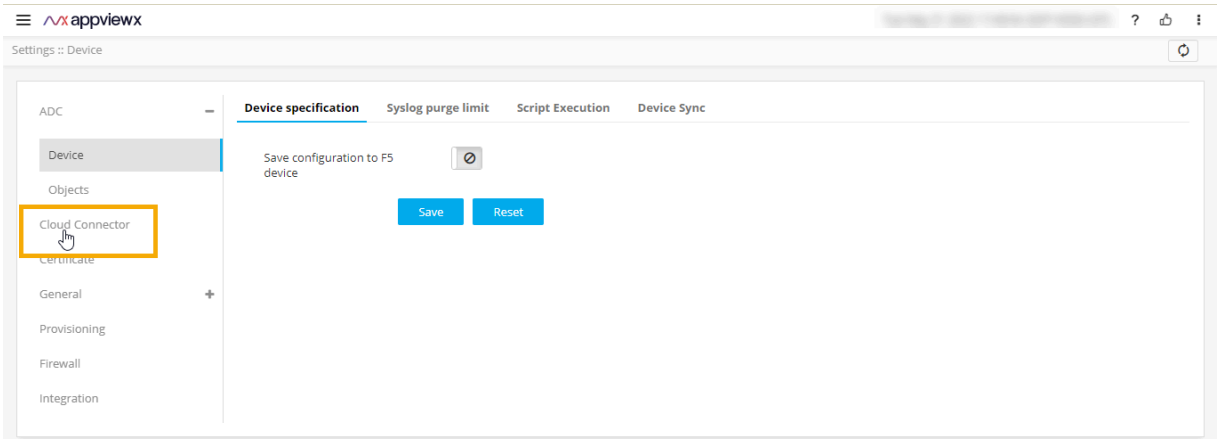
To perform a prerequisite check:

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**.



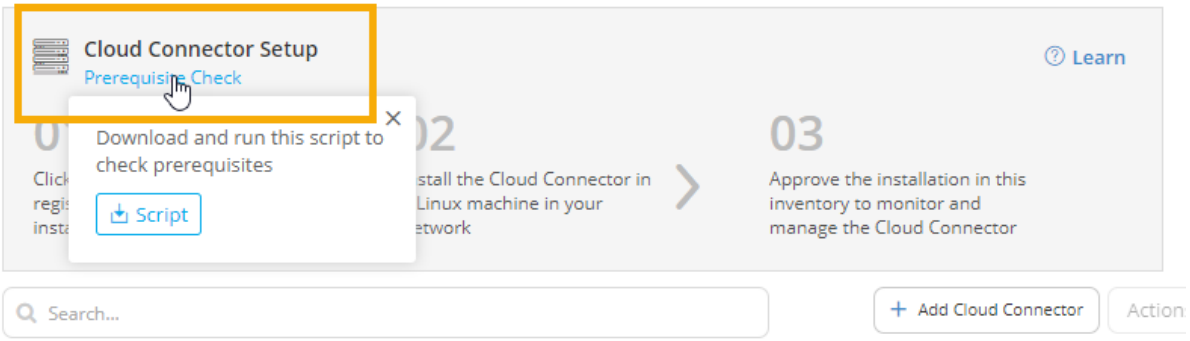
By default, the **Settings :: Device** page is displayed.

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

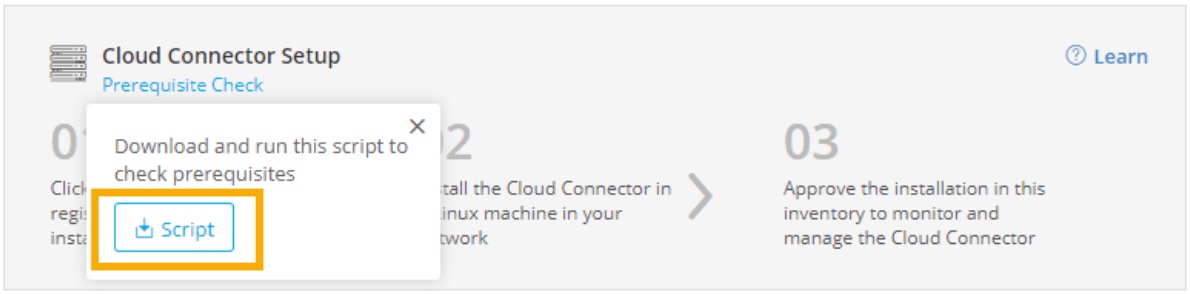


The **Settings :: AppViewX Cloud Connector** page is displayed.

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



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

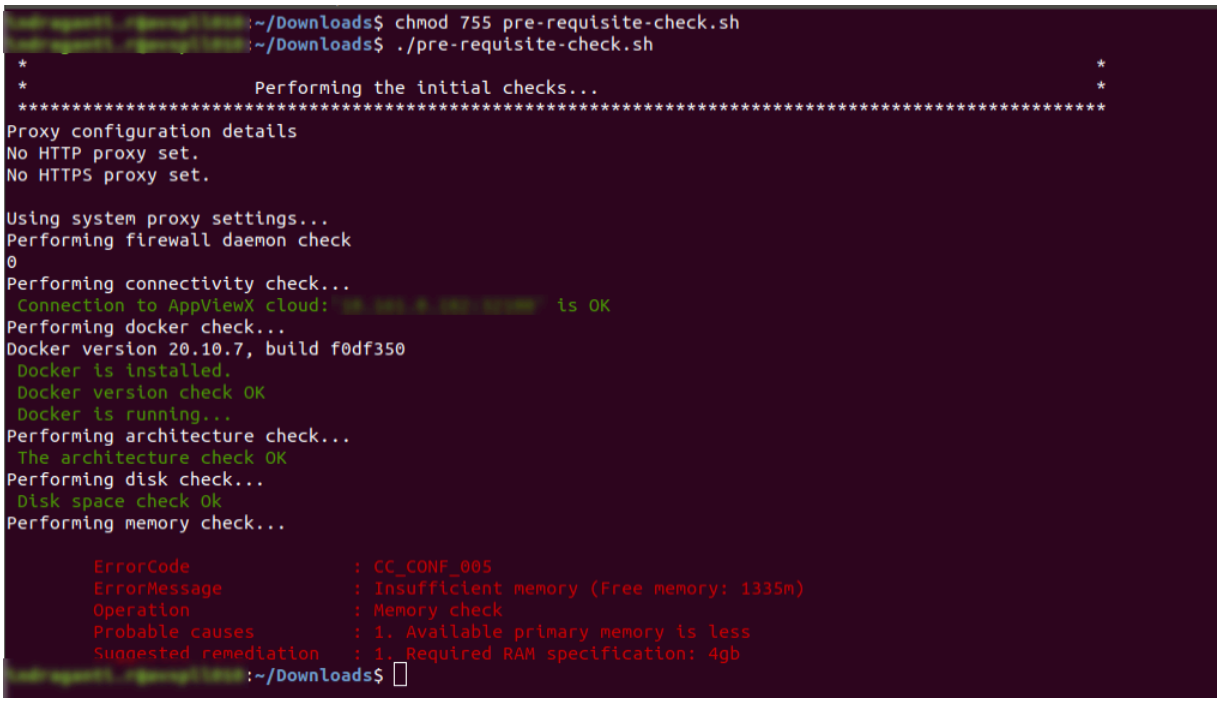


The **pre-requisite-check.sh** script file is downloaded.

6. 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.



Note: For resolutions to the prerequisite check failure scenarios, refer the AppViewX Cloud Connector User Guide.

Chapter 4: Getting Started with the AppViewX Free Trial

For users evaluating the AppViewX SaaS solution, which enables turnkey Certificate Lifecycle Management, ADC management and automation, and PKI, AppViewX enables two channels to onboard you for a free trial of the product:

- via the AppViewX website
- via the AWS Marketplace

Chapter 5: Signing Up for the Free Trial via the AppViewX Website

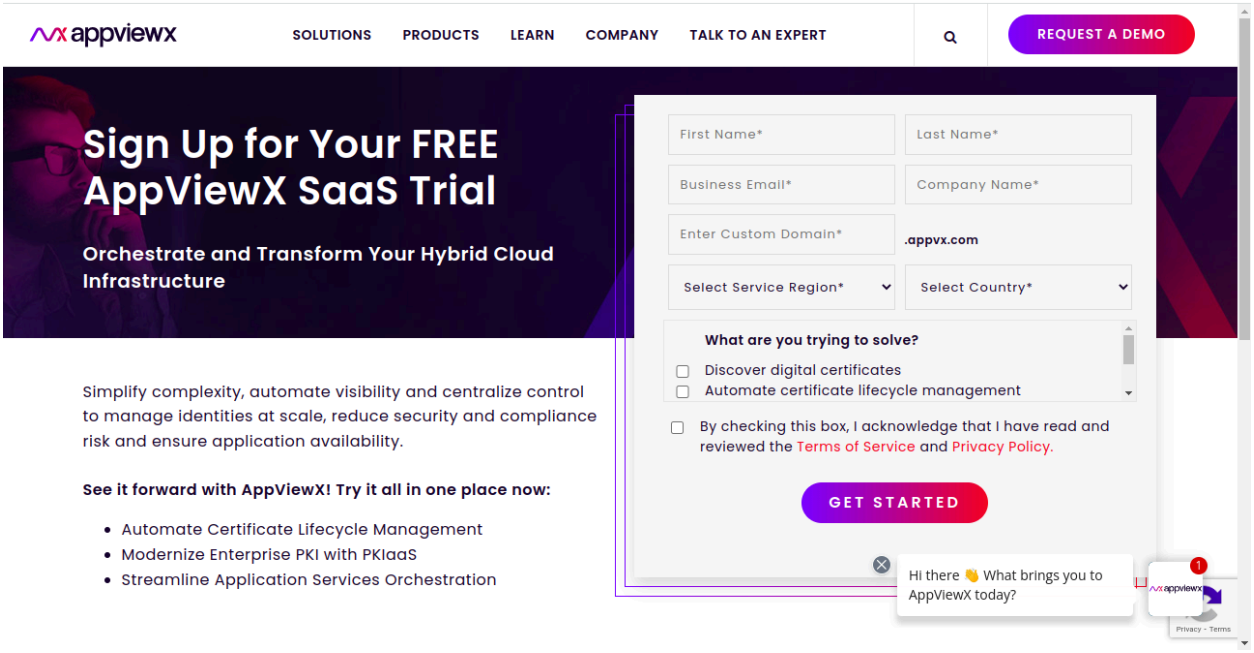
- Accessing the Website Sign Up Page
- Entering Details on the Sign Up Page
- Verifying your Email
- Logging in to your SaaS Account
- Getting Started with AppViewX SaaS
- Methods to Set up the AppViewX Cloud Connector
- Working with AppViewX SaaS

Accessing the Website Sign Up Page

To sign up from the AppViewX website:

Navigate to the AppViewX Sign Up page.

The **Sign Up** page appears.



Entering Details on the Sign Up Page

1. Enter the fields as described:

Field	Description
First Name*	Enter your first name.
Last name*	Enter your last name.
Business Email*	Enter your business email address.
Company Name*	Enter your company name.
Enter Custom Domain*	By default, the company name is auto-filled. Enter a custom domain if you want to.
Select Service Region*	The service region is where your SaaS account will be set up and localized. You cannot migrate data between regions. Select from one of the service regions: • US (Americas) • EMEA • APAC
Select Country*	Select the country from the dropdown list.

Note:

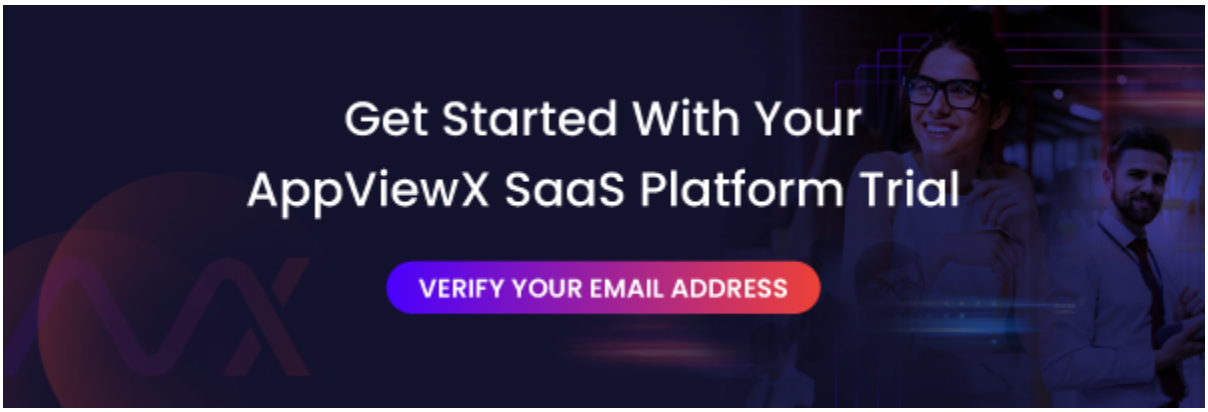
- Fields marked with the asterisk (*) symbol are mandatory.
- If you are creating a free or trial account, there are email restrictions put in place for security reasons. Email addresses from Gmail.com, Outlook.com, Yahoo.com, and other personalized email addresses are restricted and may not be used for trial account creation purposes.

2. From the **What are you trying to solve** list, select the corresponding checkboxes for your requirements.
3. To acknowledge that you have read and reviewed AppViewX's Terms of Service and their Privacy Policy, select the **By checking this box, I acknowledge...** checkbox.

4. Click **Get Started.**
   The message, *Thank you for signing up for the free trial! You will receive an email from us shortly,* is displayed.

Verifying your Email

On clicking **Get Started,** you will get a verification email to your registered email address. Click **Verify Email Address** to get your SaaS account set-up.



Note:

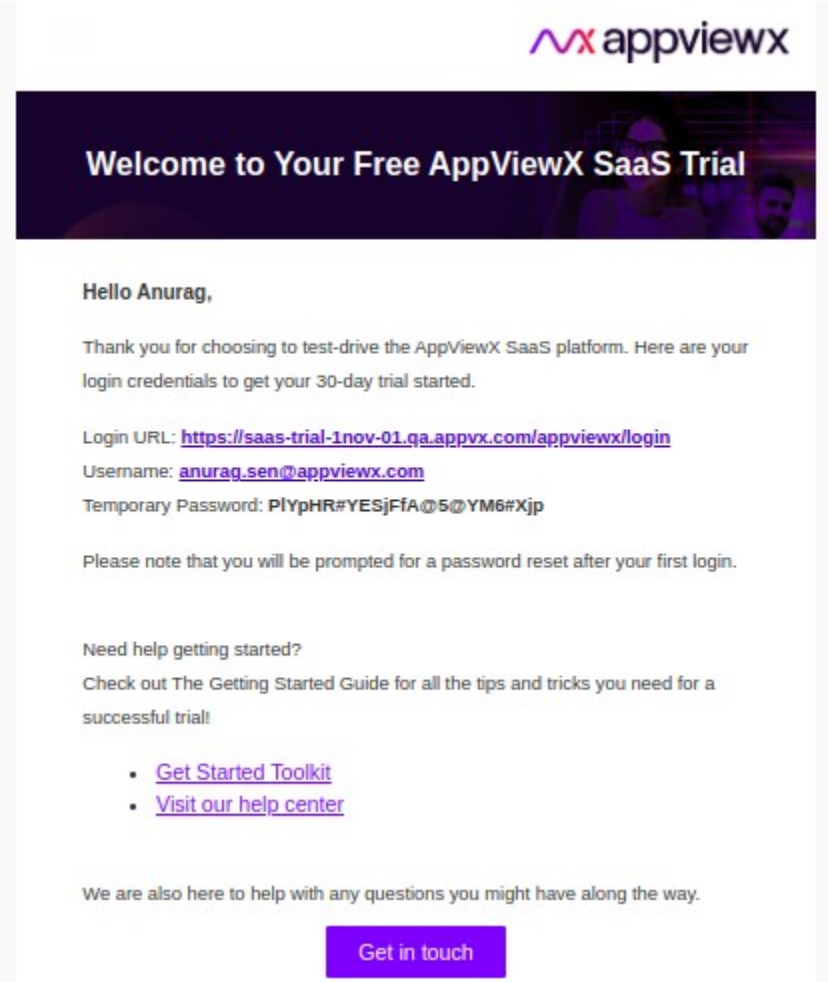
- If you do not see the email in your inbox, then check the Junk/Spam folder. Whitelist the email address so you receive all AppViewX emails in your inbox.
- Confirm your email address within 48hours.

Wait for a couple of minutes until your email address is successfully verified.

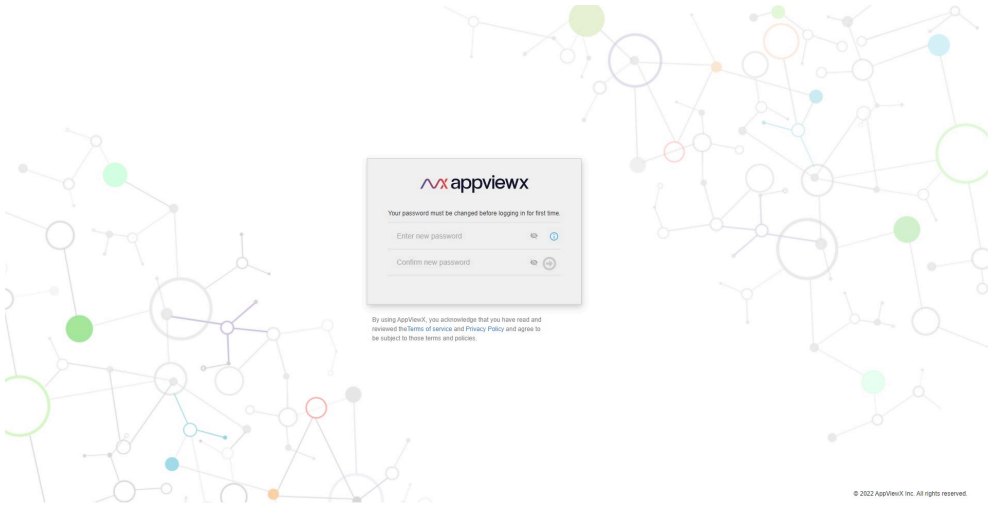
Logging in to your SaaS Account

Based on the sign-up information and region, you will be notified with your SaaS account and temporary credentials.

1. Check your inbox for another email welcoming you to the AppViewX trial along with the login credentials to access your free account.



2. Click the SaaS account URL (for example, **https://tetrik.appvx.com/appviewx/login**). You are redirected to the login page.
3. Login to your SaaS account with the credentials provided in the email. You are asked to change the current password.



4. Reset the default credentials by choosing a new password.

> **Note:** The password must:
> - Have at least one uppercase character
> - Have at least one lowercase character
> - Have one special character such as ~!@#$^&*_-+=|()
> - Have minimum of 6 characters and maximum of 24 characters
> - Not contain user name
> - Not contain more than 3 same characters continuously, for example, aaa
> - Not contain blank space

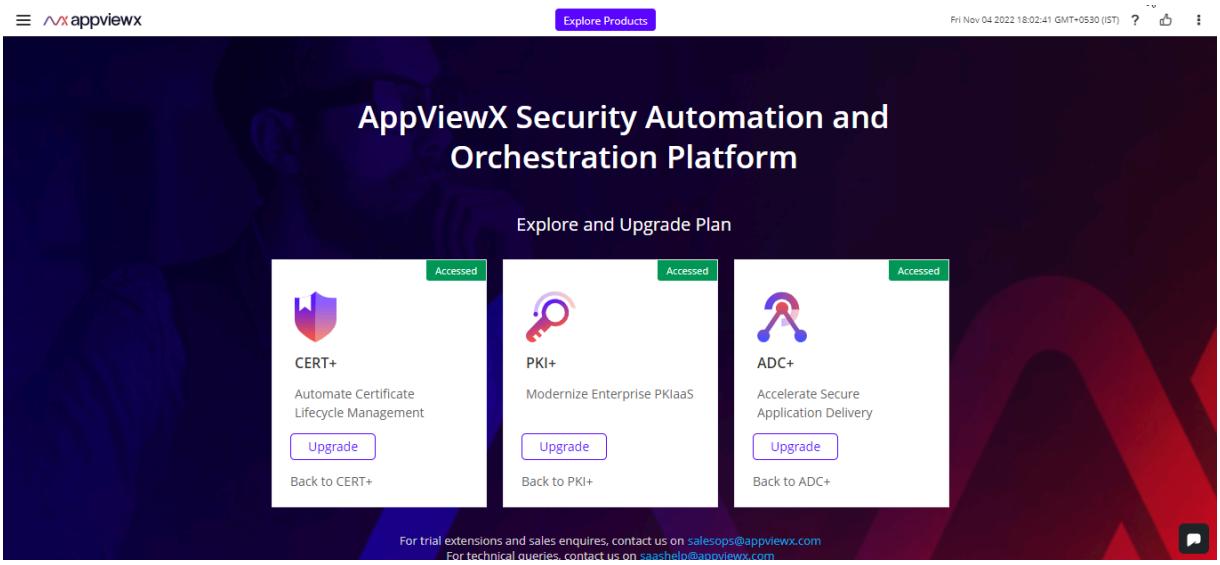
   A message, *Your password has been changed successfully. Please click here to login*, appears.

5. Click **login.**

   The login page appears.

6. Log in using your user name and the new password.

   The AppViewX SaaS landing page is displayed. Click the **Back to <product name>** link to be navigated to that product's individual interface.



Your onboarding is now complete and your account is valid for the next 30 days.

Getting Started with AppViewX SaaS

To help you get started, given below are links to the individual user guides for the AppViewX products:

- To get started with CERT+ SaaS, refer to the CERT+ SaaS User Guide.
- To get started with ADC+ SaaS, refer to the ADC+ SaaS User Guide.
- To get started with PKI+, refer to the PKI+ User Guide.

Methods to Set up the AppViewX Cloud Connector

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

- **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.

- **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).

• Setting up the AppViewX Cloud Connector via the Native OS
• Setting up the AppViewX Cloud Connector via a Virtual Image

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 occurs 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.

Note: If this AppViewX Cloud Connector installation requires configuing a proxy server, click here for instructions.

• Overview
• Step 1: Accessing the Setup Interface
• Step 2: Executing the Prerequisite Check Script
• Step 3: Installing the AppViewX Cloud Connector

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
- Operating System
- Docker Prerequisites
- Server and Network Prerequisites

Hardware

Each AppViewX Cloud Connector instance requires the following minimum configuration:

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

Operating System

- Ubuntu version 20.04
- CentOS version 7.7 and 7.9

Docker Prerequisites

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

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

For Docker installation instructions, refer to the links below:

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

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

Server and Network Prerequisites

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

  To do this, execute the following sequence of commands:

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

- Ensure that the AppViewX Cloud Connector can establish connectivity with the AppViewX 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 as given below. When connectivity has been established successfully, the command will return the HTTP code **200**.

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

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

```
apt-get install curl
```

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

```
yum install curl
```

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

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

  ```
  sudo ufw status
  ```

  To permanently disable firewalld, execute the command given below:

  ```
  sudo ufw disable
  ```

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

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

  ```
  sudo systemctl status firewalld --now
  ```

  To permanently disable the firewalld, execute the command given below:

  ```
  sudo systemctl disable firewalld --now
  ```

  To restrict other devices from enabling the firewalld, execute the command given below:

  ```
  sudo systemctl mask firewalld --now
  ```

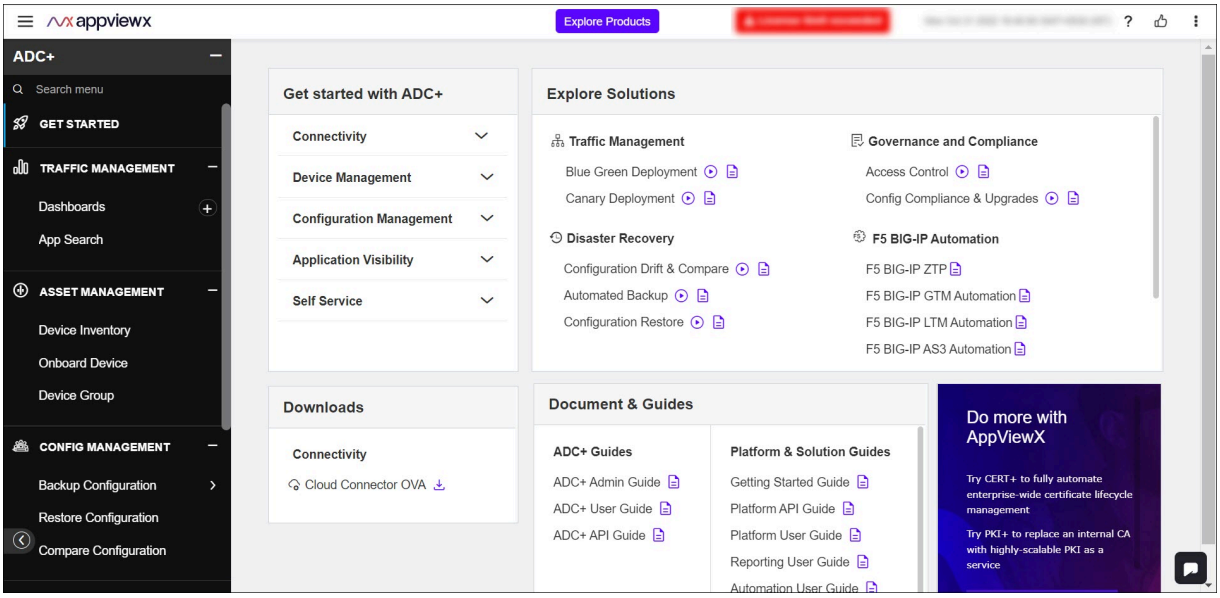
Step 1: Accessing the Setup Interface

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.

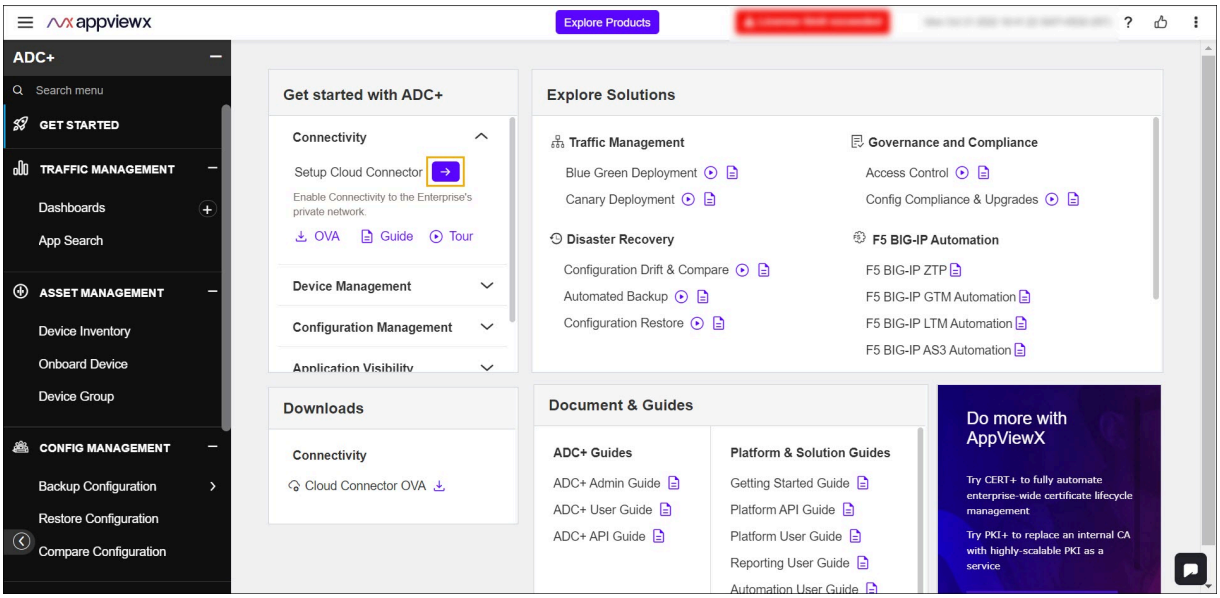
Important: As an additional layer of security, AppViewX issues client certificates to access the AppViewX GUI. The client certificate will be made available as part of the onboarding process. Upload this client certificate to the browser to start accessing the product.

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.



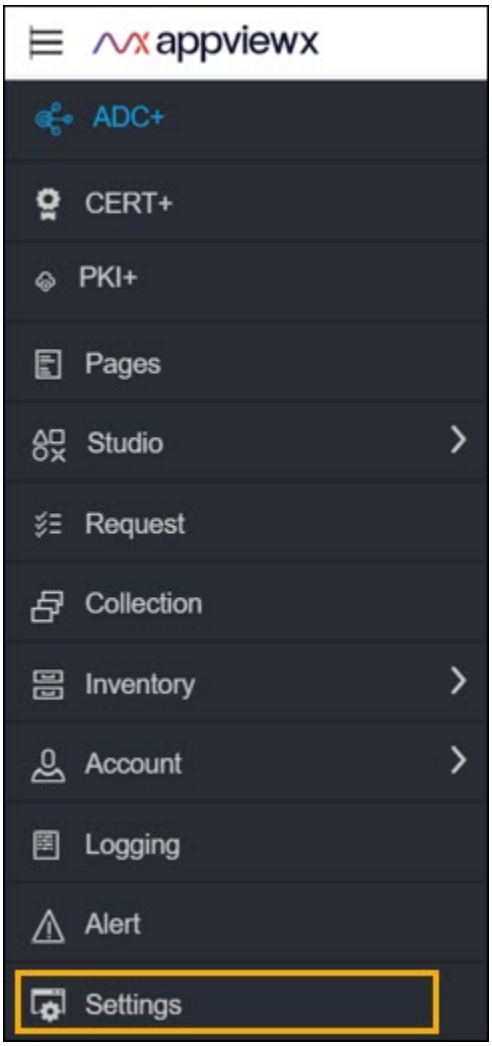
Redirected to the Settings :: Cloud Connector page.

Step 2: 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.

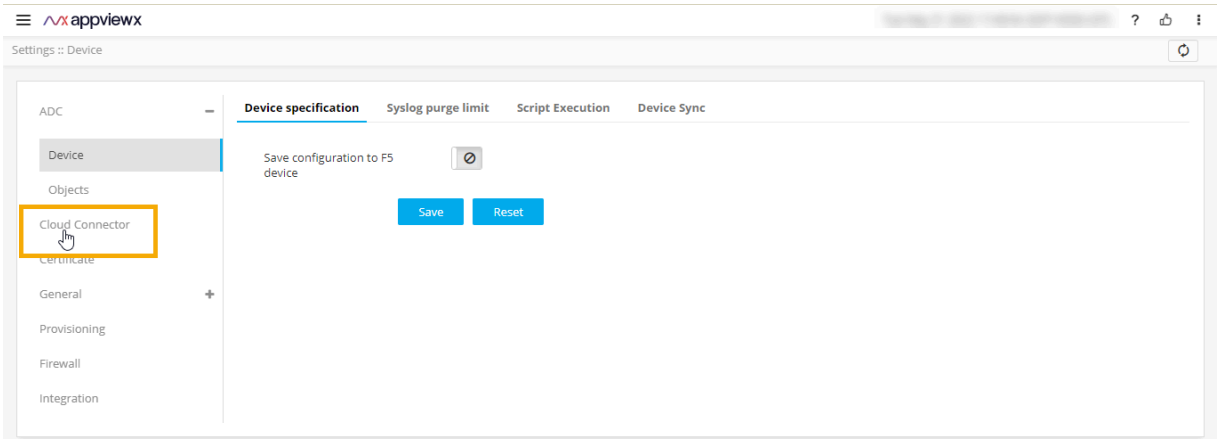
To perform a prerequisite check:

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**.



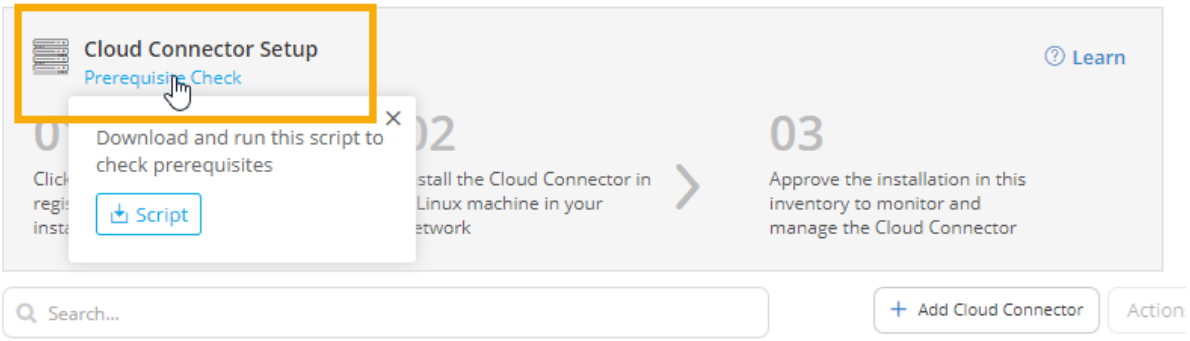
By default, the **Settings :: Device** page is displayed.

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

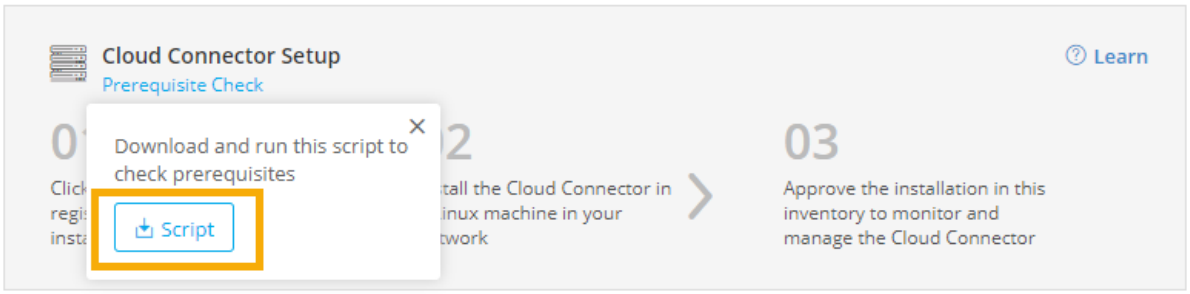


The **Settings :: AppViewX Cloud Connector** page is displayed.

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



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

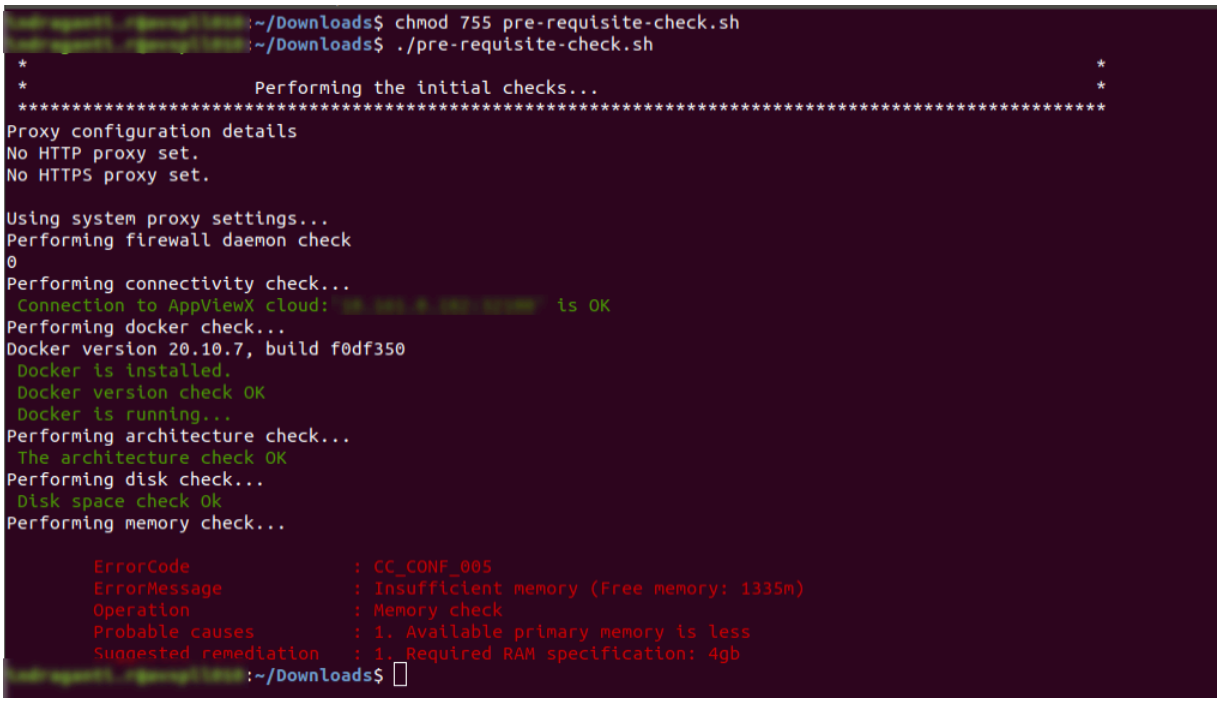


The **pre-requisite-check.sh** script file is downloaded.

6. 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.



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

Step 3: Installing the AppViewX Cloud Connector

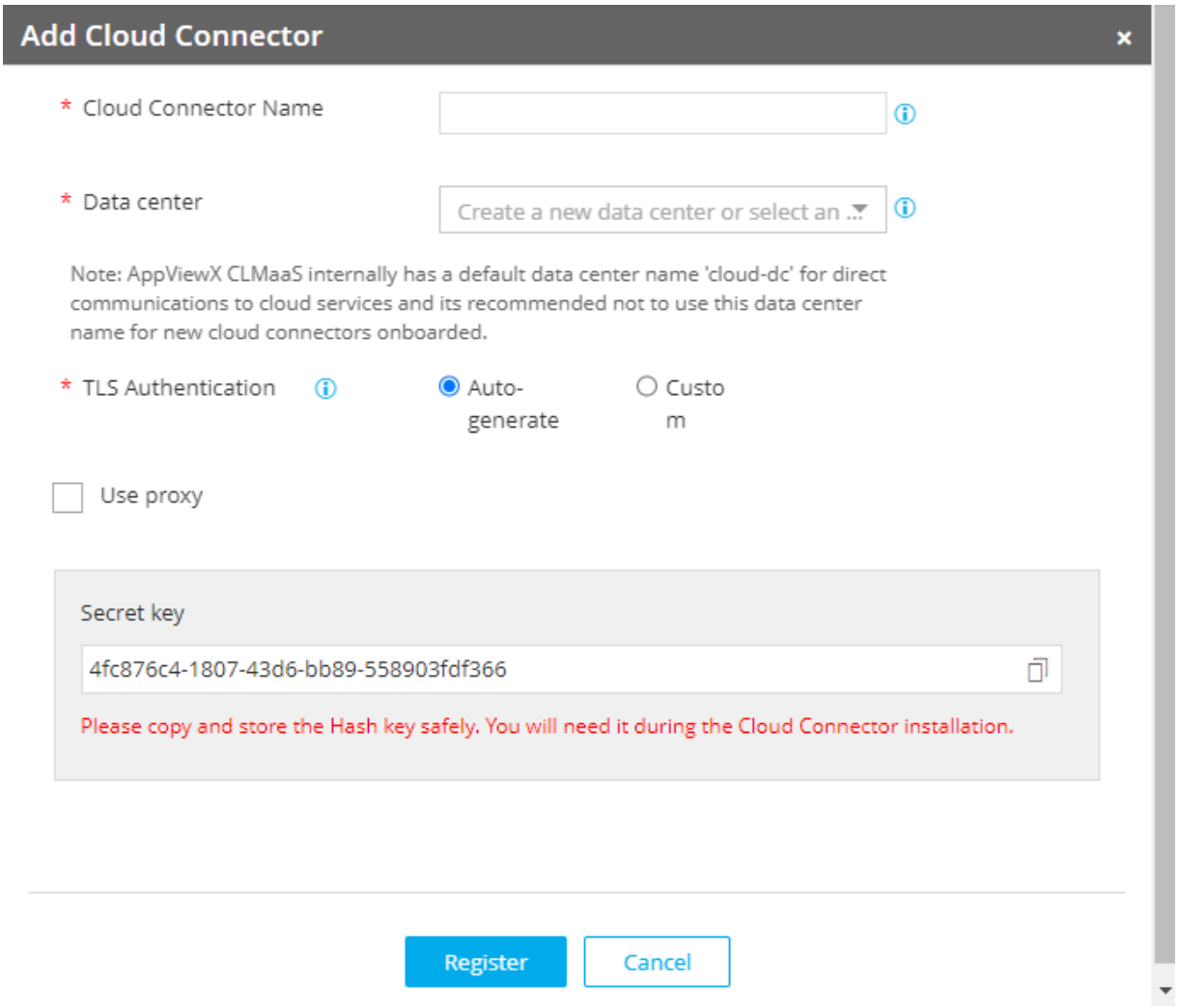
The process of deploying the AppViewX Cloud Connector involves the steps listed below. The substeps for each of these steps are outlined in the subsequent sections.

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



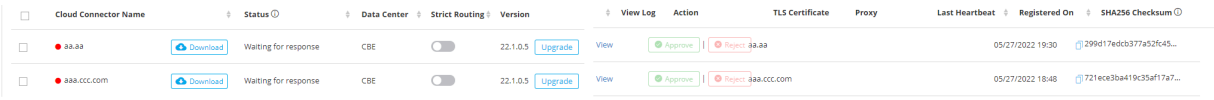
2. In the **Add Cloud Connector** action pane, enter the following details (sample values are shown in an image below the table):

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

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

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

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



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

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

.
5. 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.

1. 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`
2. 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:
kubectl cluster-info
Cluster setup is completed. Will start the deployment shortly...
Importing the required images...
[36mINFO[0m[0000] Importing image(s) into cluster 'cc'
[36mINFO[0m[0000] Importing images from 1 tarball(s)...
[36mINFO[0m[0000] Importing images '[/home/appviewx/CCTEST/deps/tools/mid-server-docker-image/avx-mid-server-base-22.1.0.0.tar]' into node
 'k3d-cc-server-0'...
[36mINFO[0m[0024] Successfully imported image(s)
[36mINFO[0m[0024] Successfully imported 1 image(s) into 1 cluster(s)
Import in progress...
[36mINFO[0m[0000] Importing image(s) into cluster 'cc'
[36mINFO[0m[0000] Importing images from 1 tarball(s)...
[36mINFO[0m[0000] Importing images '[/home/appviewx/CCTEST/deps/tools/mid-server-docker-image/k3d-tools-5.2.2.tar]' into node 'k3d-cc-server-0'...
[36mINFO[0m[0005] Successfully imported image(s)
[36mINFO[0m[0005] Successfully imported 1 image(s) into 1 cluster(s)
Import in progress...
[36mINFO[0m[0000] Importing image(s) into cluster 'cc'
[36mINFO[0m[0000] Importing images from 1 tarball(s)...
[36mINFO[0m[0000] Importing images '[/home/appviewx/CCTEST/deps/tools/mid-server-docker-image/rancher-mirrored-coredns-coredns-1.8.6.tar]' into
 node 'k3d-cc-server-0'...
[36mINFO[0m[0007] Successfully imported image(s)
[36mINFO[0m[0007] Successfully imported 1 image(s) into 1 cluster(s)
```

```
[36mINFO[0m[0000] Importing image(s) into cluster 'cc'
[36mINFO[0m[0000] Importing images from 1 tarball(s)...
[36mINFO[0m[0000] Importing images '[/home/appviewx/CCTEST/deps/tools/mid-server-docker-image/rancher-local-path-provisioner-v0.0.21.tar]' into node
 'k3d-cc-server-0'...
[36mINFO[0m[0004] Successfully imported image(s)
[36mINFO[0m[0004] Successfully imported 1 image(s) into 1 cluster(s)
[36mINFO[0m[0000] Importing image(s) into cluster 'cc'
[36mINFO[0m[0000] Importing images from 1 tarball(s)...
[36mINFO[0m[0000] Importing images '[/home/appviewx/CCTEST/deps/tools/mid-server-docker-image/rancher-mirrored-pause-3.6.tar]' into node
 'k3d-cc-server-0'...
[36mINFO[0m[0003] Successfully imported image(s)
[36mINFO[0m[0003] Successfully imported 1 image(s) into 1 cluster(s)
Deploying the Cloud Connector...
NAME: avx-mid-server-starter
LAST DEPLOYED: Mon May 30 15:51:13 2022
NAMESPACE: cc
STATUS: deployed
REVISION: 1
NOTES:
1. It may take a couple of minutes for the Cloud Connector to be up.
  kubectl get pod --namespace cc
**************************************************************************
*    Congratulations!!! The installation completed successfully.    *
*    Please wait till the Cloud Connector is up and running.       *
**************************************************************************
(1%) Cloud Connector status: Running
[32m Cloud Connector is up and running. (B[m
```

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

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

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

When the platform component download is in progress, it is indicated by the symbol prefixed

to the platform component version number in the AppViewX Cloud Connector inventory details  . A completed download/upgrade is indicated by the  symbol in the same location  .

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

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 a Virtual Image

The AppViewX Virtual Image is an Open Virtual Appliance (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.

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 configuing a proxy server, click here for instructions.

To set up the AppViewX Cloud Connector via a virtual image:

- 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**.

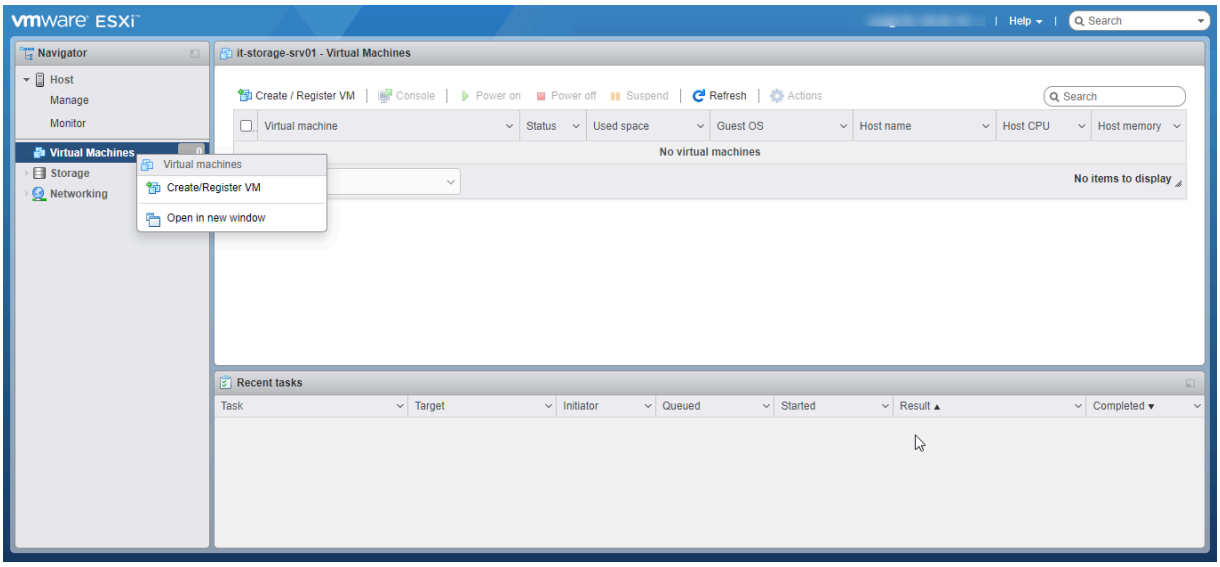
- Downloading the OVA Release Package
- Installing the AppViewX OVA

Downloading the OVA Release Package

From the download URL, download the release package in the OVA format.

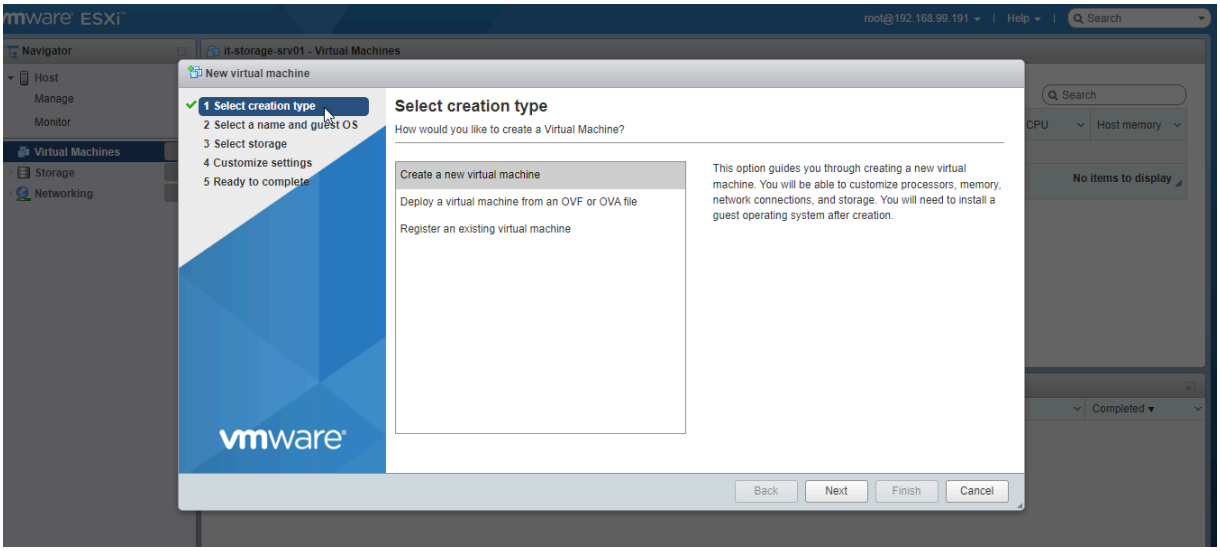
Installing the AppViewX OVA

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

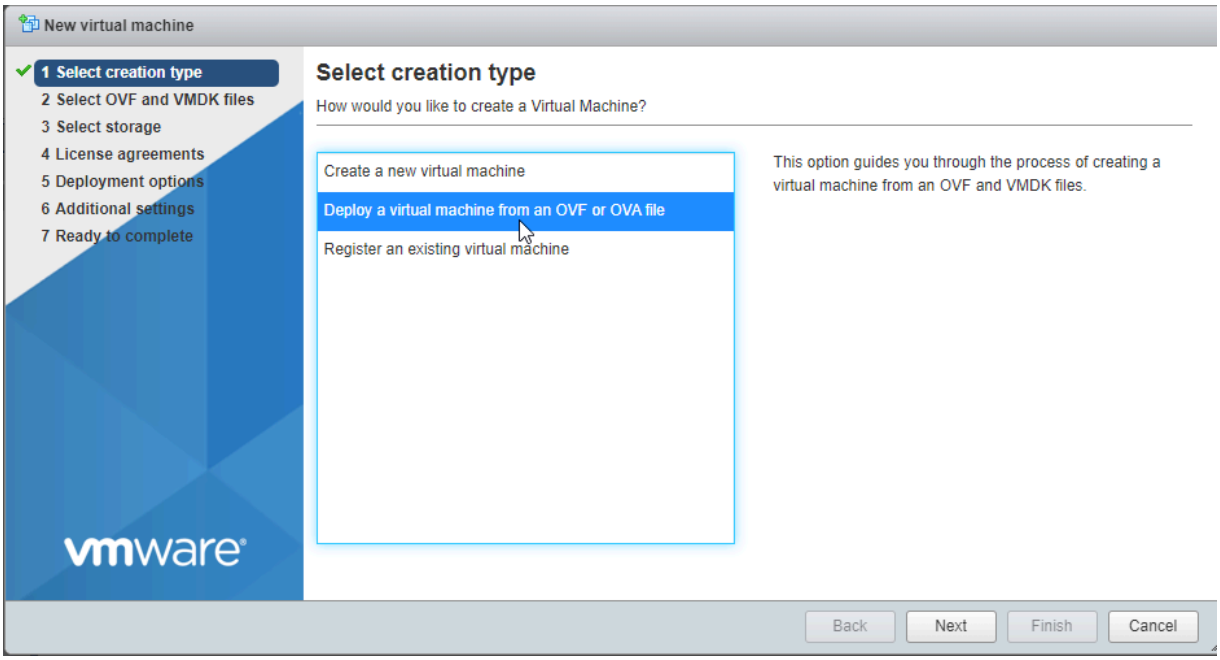


The **New Virtual machine** window is displayed.

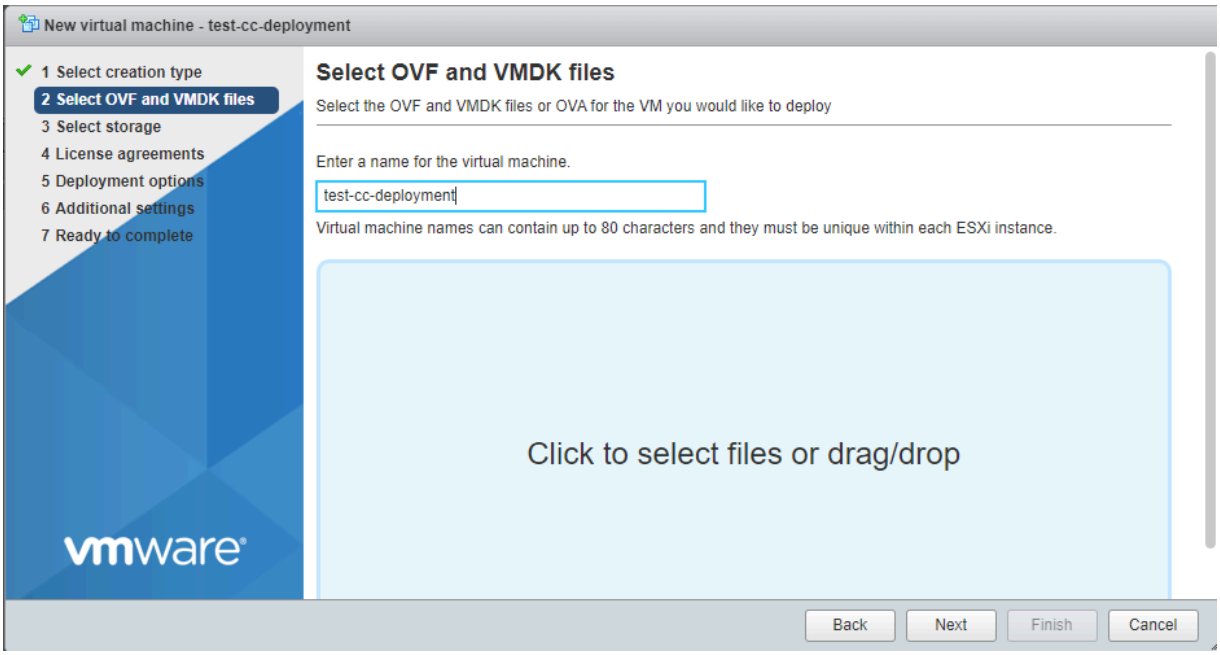
4. From the navigation pane in the left, select **Select creation type**.



5. In the **Select creation type** window, select the **Deploy a virtual machine from an OVA or OVF file** option.



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

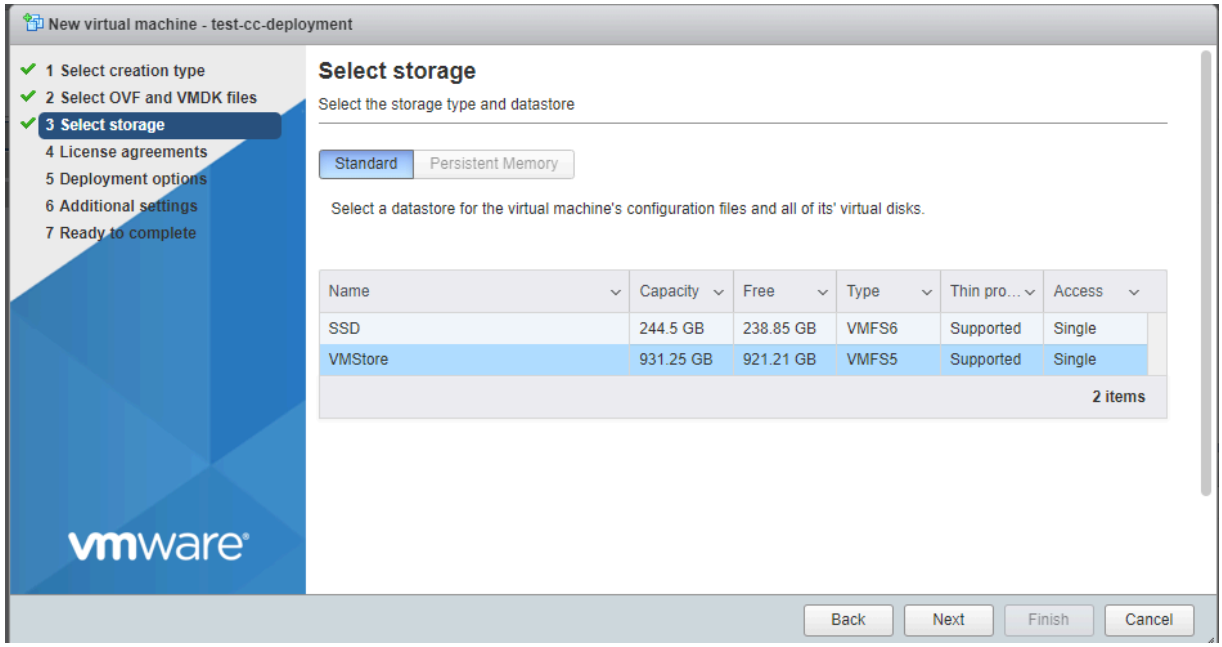


   a. Enter a name for the virtual machine.

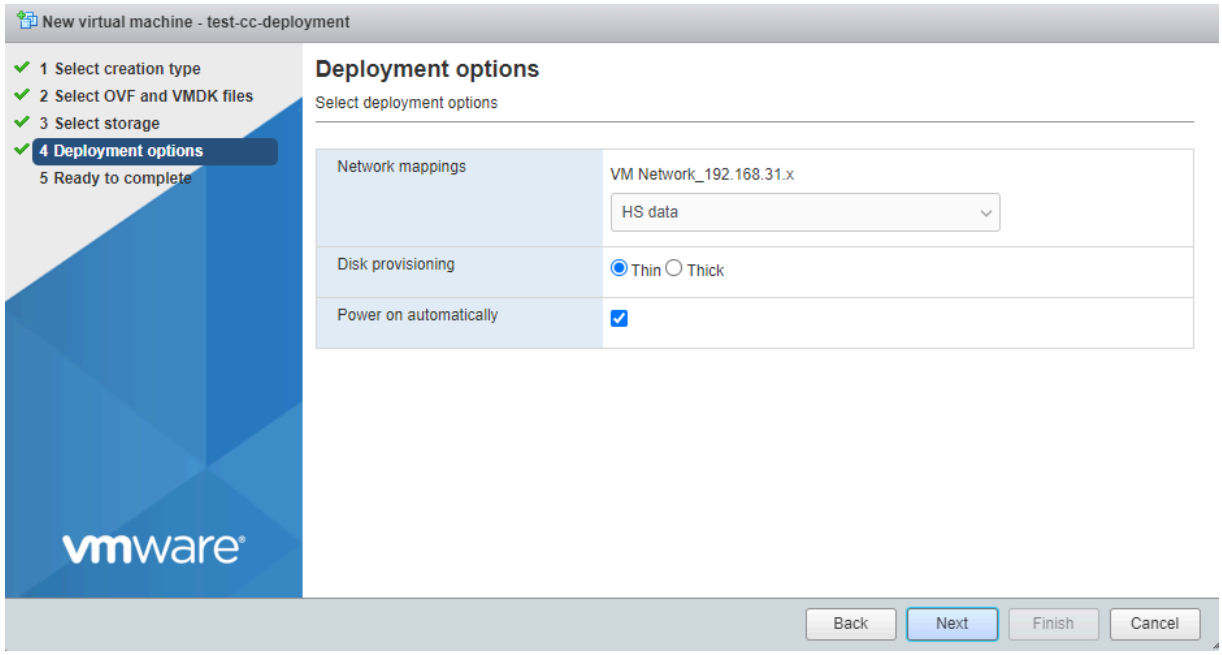
      For the purpose of this document, we will name it **test-cc-deployment**.

   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**.

8. Click **Next**.
9. 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.



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

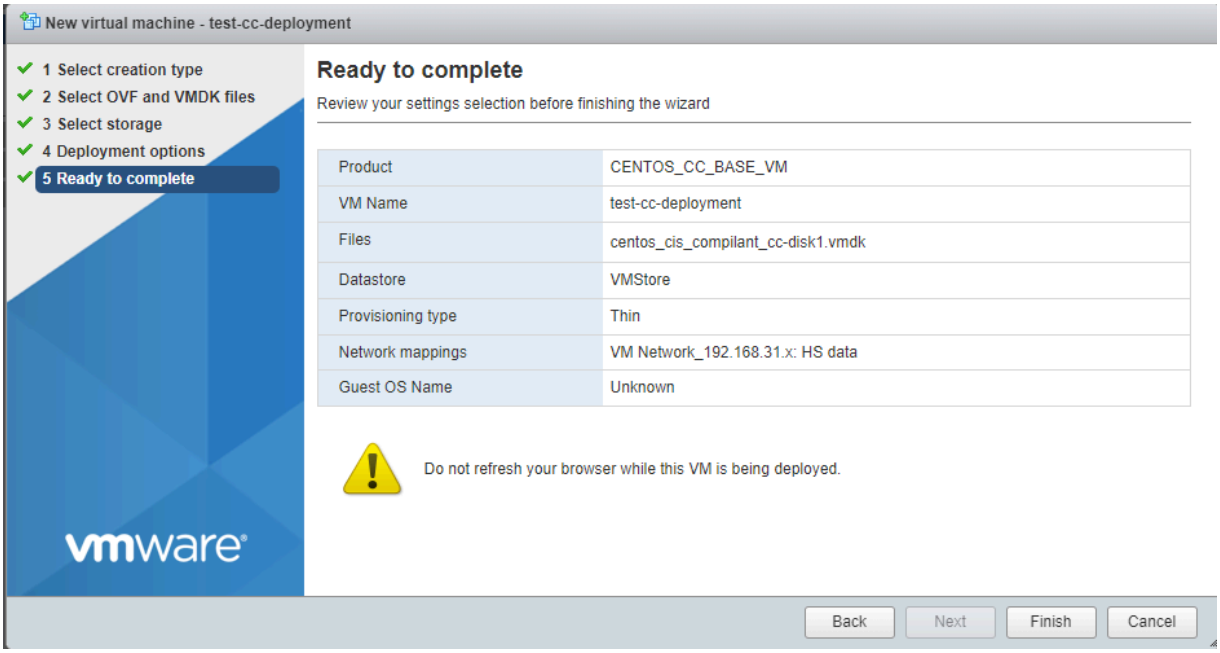


a. Select the network mapping.

b. Select the disk provisioning required.

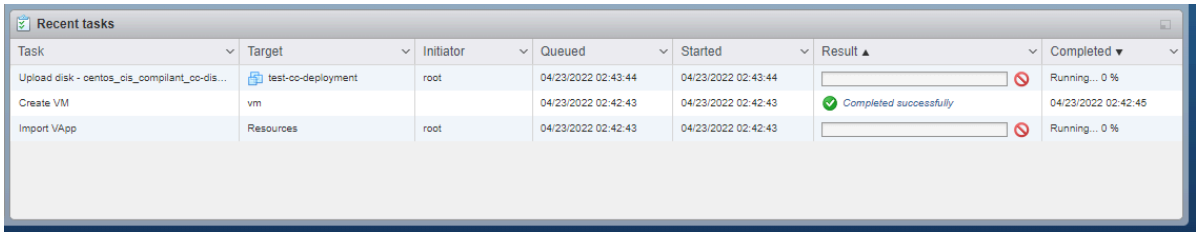
c. Select the **Power on automatically** checkbox.

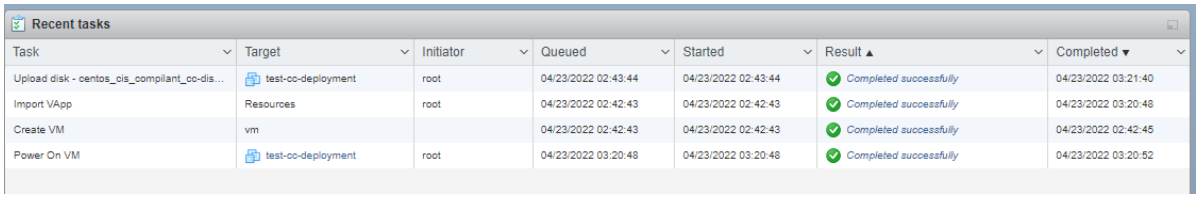
12. Click **Next**.
13. In the **Ready to complete** window, review your settings.



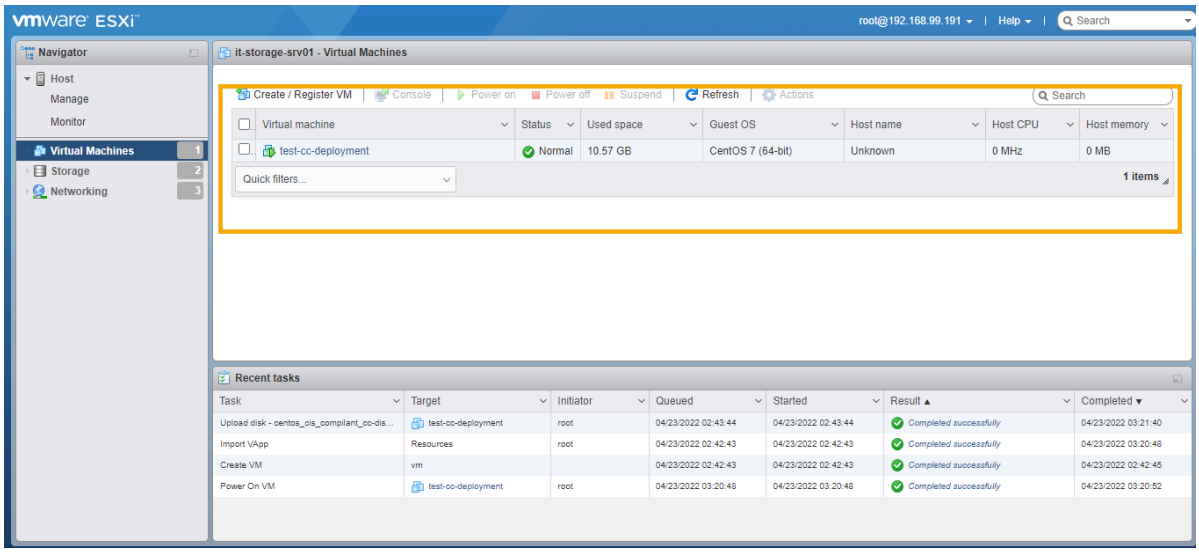
14. Click **Finish**.

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





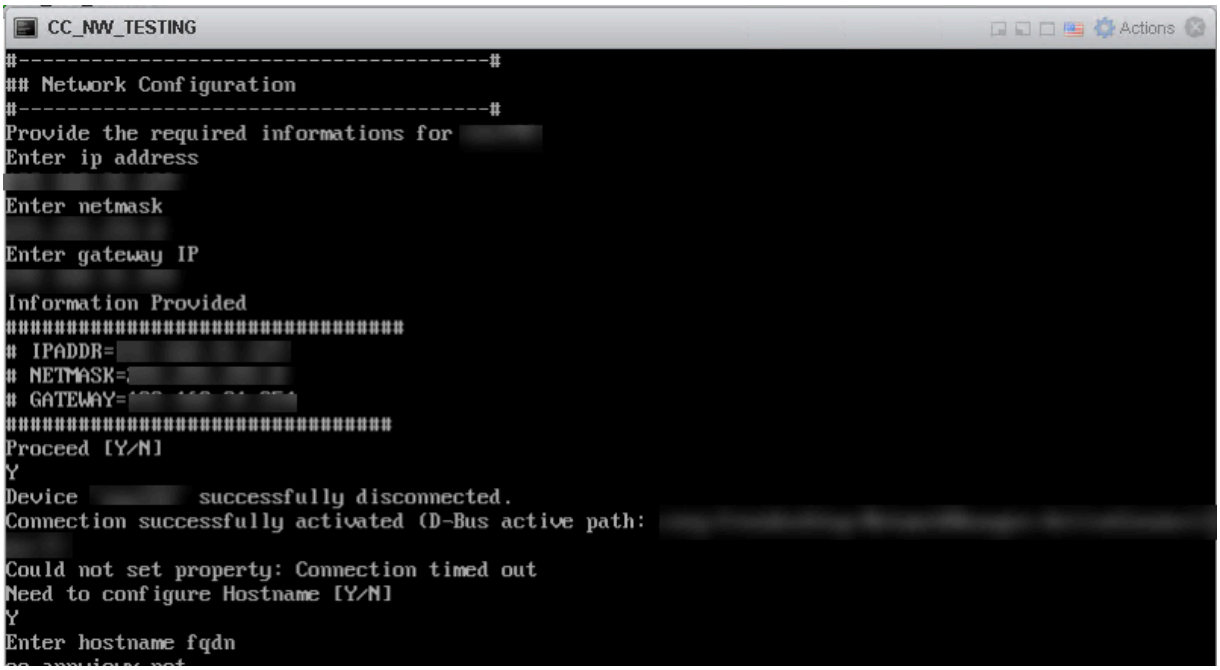
- On successful completion of the OVA deployment, the new virtual machine is displayed in the **Virtual Machines** inventory. For each virtual machine in the inventory, the following details are displayed:



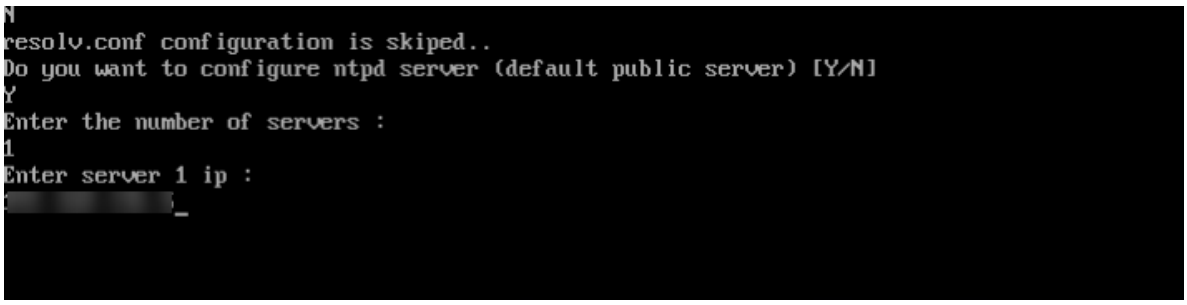
15. From the **Virtual Machines** inventory, click the virtual machine just added.
    The terminal window for the virtual machine is displayed. The script for configuring the network IP is executed automatically.
16. To configure the IP address, when prompted, enter the required values for the following 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:



17. 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**.
18. To configure an NTP server(s):
    a. When prompted, **Do you want to configure ntpd server (default public server)**press **Y**.



    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**.
19. After the script is executed, to login to the VM, when prompted, use one of the following set of credentials:
    • For the appviewx user
      • Username: **appviewx**
      • Password: **XApp23**
    • For the root user:

> **Note:** The root user access is required for maintaining the OS configuration and for patching security updates.

- Username: **root**
- Password: **PlMQaZ23**

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

20. 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:

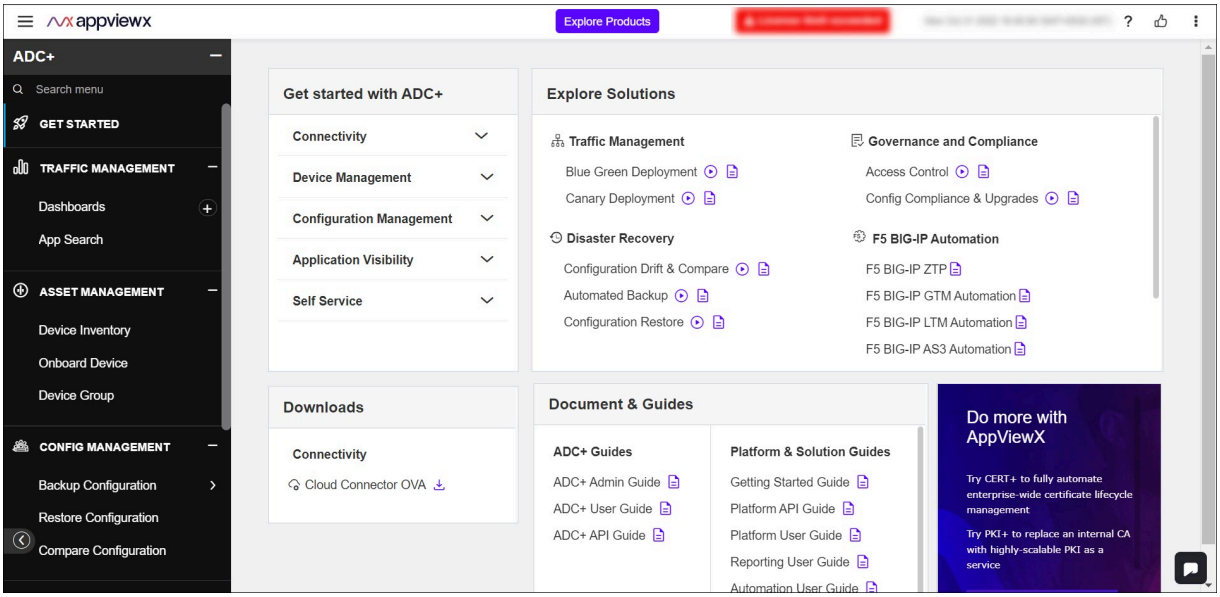
Step 2: Accessing the Setup Interface

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.

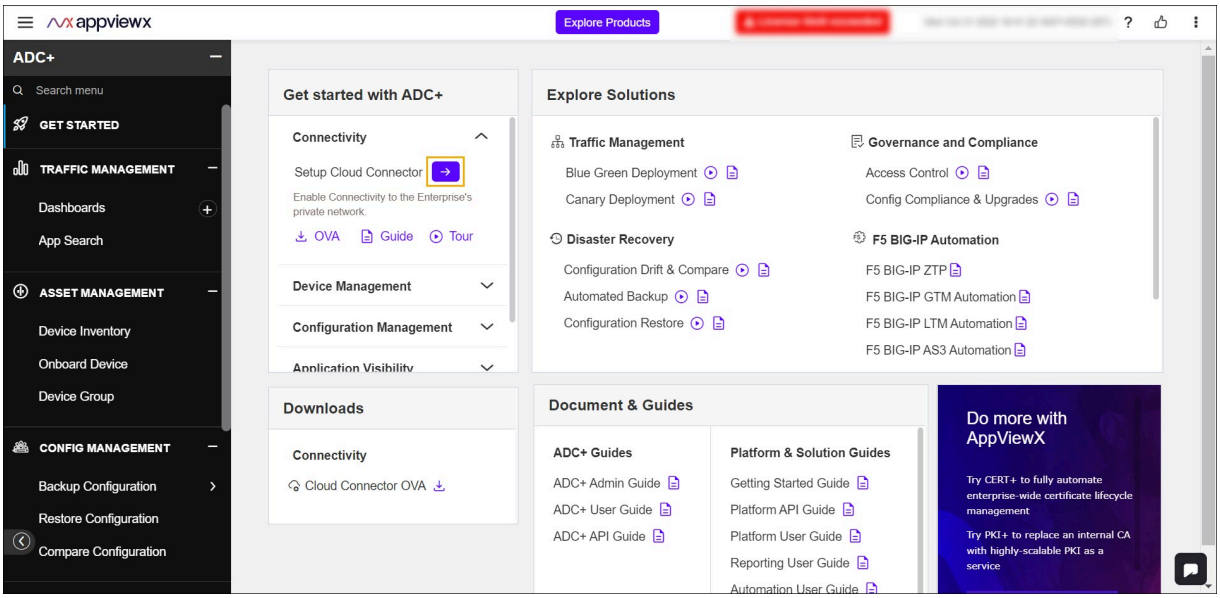
> **Important:** As an additional layer of security, AppViewX issues client certificates to access the AppViewX GUI. The client certificate will be made available as part of the onboarding process. Upload this client certificate to the browser to start accessing the product.

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.



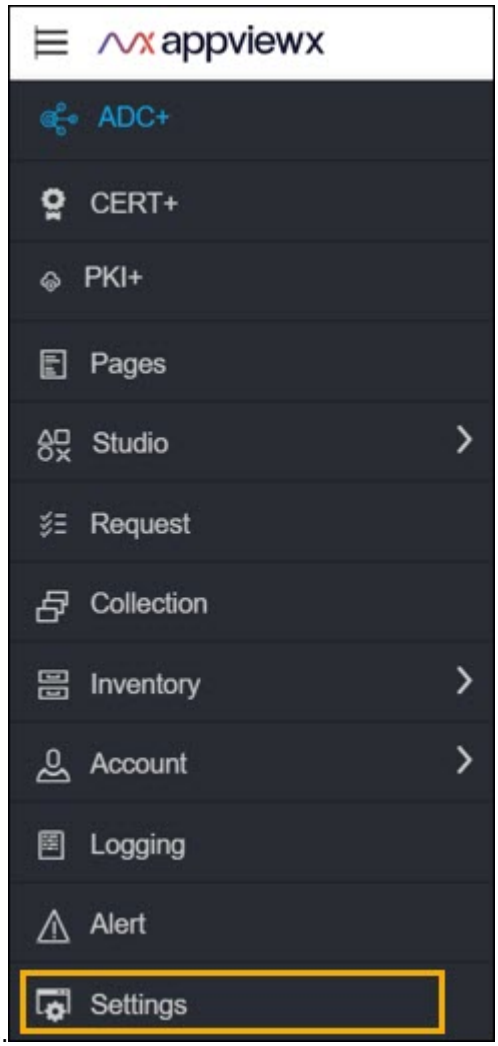
Redirected to the Settings :: Cloud Connector page.

Step 3: Executing the Prerequisite Check Script

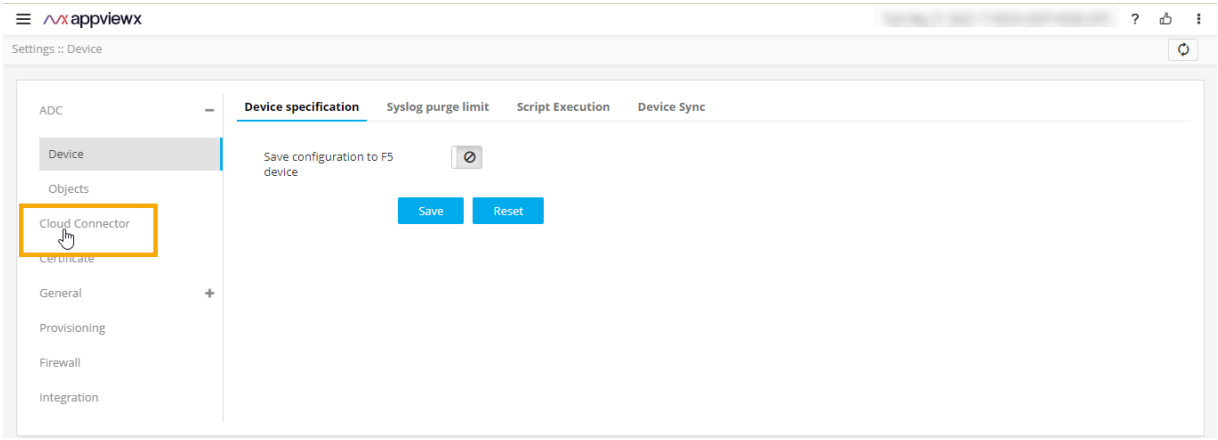
To simplify compliance to the AppViewX Cloud Connector installation prerequisites, you can execute a script to identify and rule out any deviations from the requirements.

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**.

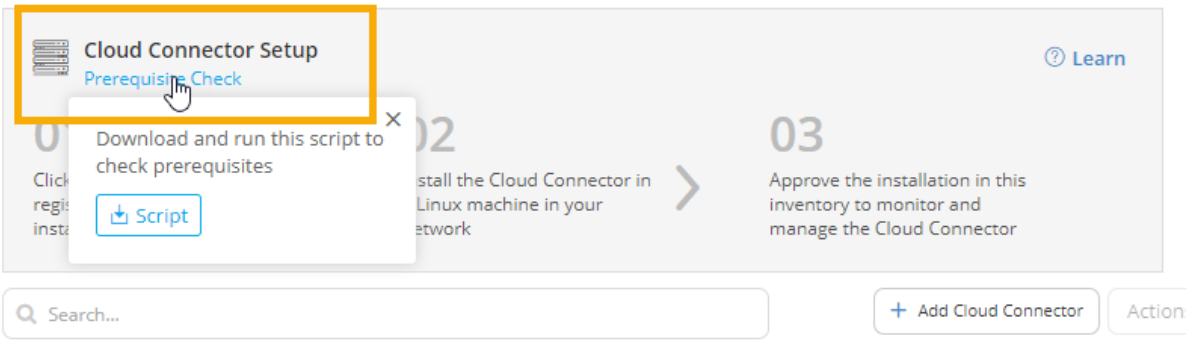


   By default, the **Settings :: Device** page is displayed.
3. From the left navigation pane, select **AppViewX Cloud Connector**.

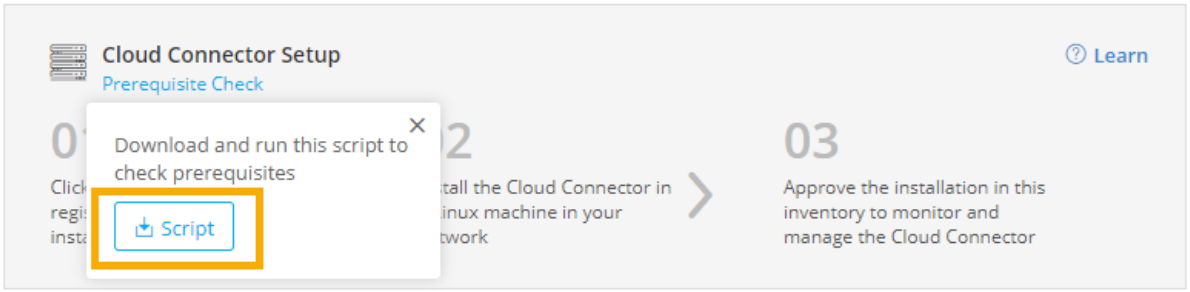


The **Settings :: AppViewX Cloud Connector** page is displayed.

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



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



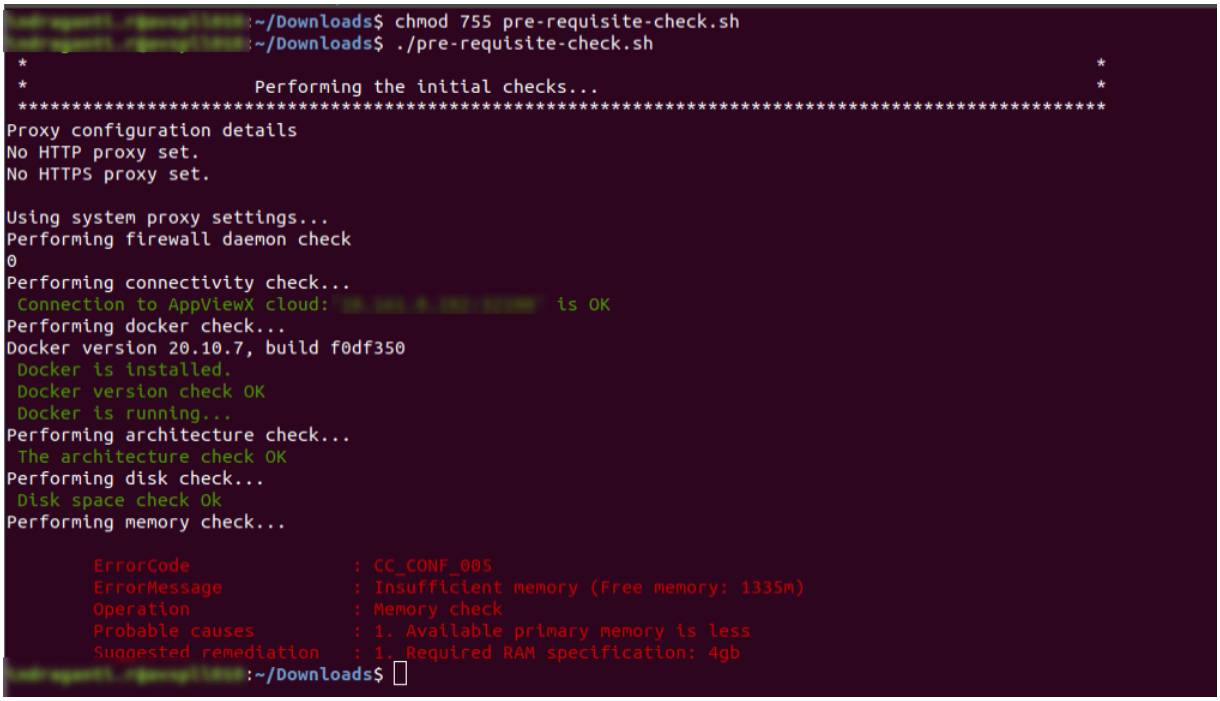
The **pre-requisite-check.sh** script file is downloaded.

6. From the command line interface, navigate to the node where the AppViewX Cloud Connector will be installed.

7. Convert the downloaded script file into an executable file using the chmod command, as shown below:`chmod 755 pre-requisite-check.sh`
8. Execute the **.sh** prerequisite check script file.

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

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



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

Step 4: Installing the AppViewX Cloud Connector

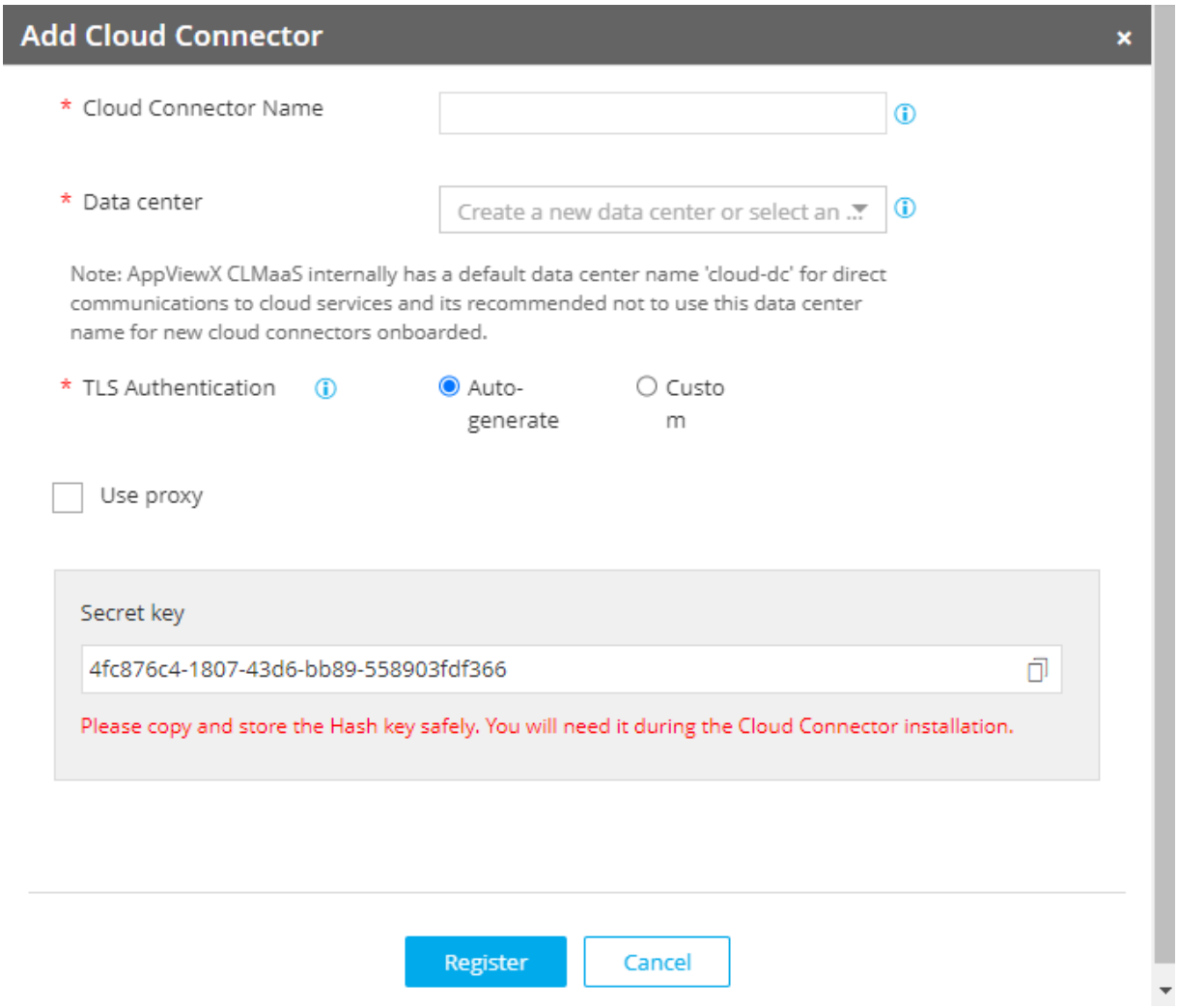
The process of deploying the AppViewX Cloud Connector involves the steps listed below. The substeps for each of these steps are outlined in the subsequent sections.

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



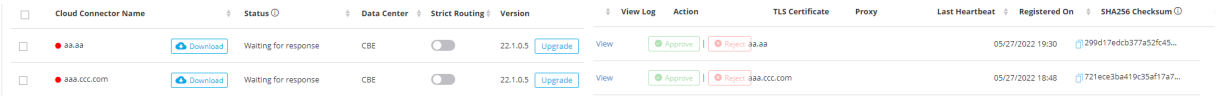
2. In the **Add Cloud Connector** action pane, enter the following details (sample values are shown in an image below the table):

Field	Description
Cloud Connector Name*	FQDN of the machine where the AppViewX Cloud Connector is to be installed
Data center*	Name of the data center where the AppViewX Cloud Connector is to be installed **Note:** The AppViewX SaaS, internally, has a default data center named **cloud-dc**, for direct communications to cloud services. It is recommended that you do not this data center name for new the AppViewX Cloud Connectors onboarded.
Authentication Type	• Select the authentication type from the following options: • **TLS** • **MTLS** • To auto-generate a TLS certificate, select **Auto-generate** (default selection). Automatically, the certificate is generated using the AppViewX CA. **Note:** The created certificate is available in the certificate inventory. You can: • Assign this certificate to a certificate group • Configure a certificate expiry alert for this certificate group from the **Server Certificate** dashboard, using the **Certificate Summary Report** widget settings • To enter details of a custom TLS certificate, select **Custom**. The **TLS Certificate Password** and **Custom TLS Certificate** fields are displayed. The instructions for filling these fields are given below.
TLS Certificate Password*	**Note:** This field is displayed only if you have selected to enter details of a Custom TLS certificate in the **TLS Authentication** field. Password of the TLS certificate (that will be uploaded in the next step)

Field	Description
	Note: This is a mandatory field if a Custom TLS certificate is uploaded. AppViewX supports only password-protected Custom TLS certificates.
Custom TLS Certificate	**Note:** This field is displayed only if you have selected to enter details of a Custom TLS certificate in the **TLS Authentication** field. To upload a custom TLS certificate: a. To navigate to the location of the custom TLS certificate, click within the field. b. Select the certificate file. c. Click **Open**. d. To upload the custom TLS certificate selected, click **Upload**. **Note:** AppViewX supports only password-protected Custom TLS Certificates.
Use proxy	To use a proxy server for the deployment: a. Select the **Use proxy** checkbox. b. To select a preconfigured proxy (for the selected data center), from the **Select Proxy** dropdown list, select a proxy server. **OR** To create a new proxy server setting: a. Use the Click here option shown below the **Select Proxy** dropdown list. b. For steps to create a new proxy server setting, click here.
Secret key	A unique key for the AppViewX Cloud Connector installation Click to copy this key and save it in a safe place.

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

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



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

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

.
5. 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.

1. 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`
2. 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:
kubectl cluster-info
Cluster setup is completed. Will start the deployment shortly...
Importing the required images...
[36mINFO[0m[0000] Importing image(s) into cluster 'cc'
[36mINFO[0m[0000] Importing images from 1 tarball(s)...
[36mINFO[0m[0000] Importing images '[/home/appviewx/CCTEST/deps/tools/mid-server-docker-image/avx-mid-server-base-22.1.0.0.tar]' into node
 'k3d-cc-server-0'...
[36mINFO[0m[0024] Successfully imported image(s)
[36mINFO[0m[0024] Successfully imported 1 image(s) into 1 cluster(s)
Import in progress...
[36mINFO[0m[0000] Importing image(s) into cluster 'cc'
```

```
[36mINFO[0m[0000] Importing images from 1 tarball(s)...
[36mINFO[0m[0000] Importing images '[/home/appviewx/CCTEST/deps/tools/mid-server-docker-image/k3d-tools-5.2.2.tar]' into node 'k3d-cc-server-0'...
[36mINFO[0m[0005] Successfully imported image(s)
[36mINFO[0m[0005] Successfully imported 1 image(s) into 1 cluster(s)
Import in progress...
[36mINFO[0m[0000] Importing image(s) into cluster 'cc'
[36mINFO[0m[0000] Importing images from 1 tarball(s)...
[36mINFO[0m[0000] Importing images '[/home/appviewx/CCTEST/deps/tools/mid-server-docker-image/rancher-mirrored-coredns-coredns-1.8.6.tar]' into
 node 'k3d-cc-server-0'...
[36mINFO[0m[0007] Successfully imported image(s)
[36mINFO[0m[0007] Successfully imported 1 image(s) into 1 cluster(s)
[36mINFO[0m[0000] Importing image(s) into cluster 'cc'
[36mINFO[0m[0000] Importing images from 1 tarball(s)...
[36mINFO[0m[0000] Importing images '[/home/appviewx/CCTEST/deps/tools/mid-server-docker-image/rancher-local-path-provisioner-v0.0.21.tar]' into node
 'k3d-cc-server-0'...
[36mINFO[0m[0004] Successfully imported image(s)
[36mINFO[0m[0004] Successfully imported 1 image(s) into 1 cluster(s)
[36mINFO[0m[0000] Importing image(s) into cluster 'cc'
[36mINFO[0m[0000] Importing images from 1 tarball(s)...
[36mINFO[0m[0000] Importing images '[/home/appviewx/CCTEST/deps/tools/mid-server-docker-image/rancher-mirrored-pause-3.6.tar]' into node
 'k3d-cc-server-0'...
[36mINFO[0m[0003] Successfully imported image(s)
[36mINFO[0m[0003] Successfully imported 1 image(s) into 1 cluster(s)
Deploying the Cloud Connector...
NAME: avx-mid-server-starter
LAST DEPLOYED: Mon May 30 15:51:13 2022
NAMESPACE: cc
STATUS: deployed
REVISION: 1
NOTES:
1. It may take a couple of minutes for the Cloud Connector to be up.
  kubectl get pod --namespace cc
**************************************************************************
*    Congratulations!!! The installation completed successfully.   *
*    Please wait till the Cloud Connector is up and running.       *
**************************************************************************
(1%) Cloud Connector status: Running
```

```
[32m Cloud Connector is up and running. (B[m
```

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

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

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

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

Working with AppViewX SaaS

The AppViewX SaaS deployment leverages the existing capabilities of the om-premise deployments of the product combining them with the benefits of a cloud-based deployment.

To simplify your interaction with the product's features, AppViewX offers exhaustive documentation in the form of the following guides:

- AppViewX SaaS Cloud Connector User Guide
- AppViewX SaaS Onboarding Guide
- CERT+ User Guide
- CERT+ Admin Guide
- Platform User Guide
- ADC+ User Guide
- ADC+ Admin Guide

You can access the complete AppViewX SaaS documentation here.

Chapter 6: Signing Up for the Free Trial via the AWS Marketplace

- Signing Up for the Free Trial via the AWS Marketplace
- Accessing the AWS Marketplace Sign Up Page
- Entering Details on the Sign Up Page
- Verifying your Email
- Logging in to your SaaS Account
- Getting Started with CERT+ SaaS
- Methods to Set up the AppViewX Cloud Connector
- Working with AppViewX SaaS

Signing Up for the Free Trial via the AWS Marketplace

Note: Currently, you can sign up only for the CERT+ SaaS trial via the AWS Marketplace.

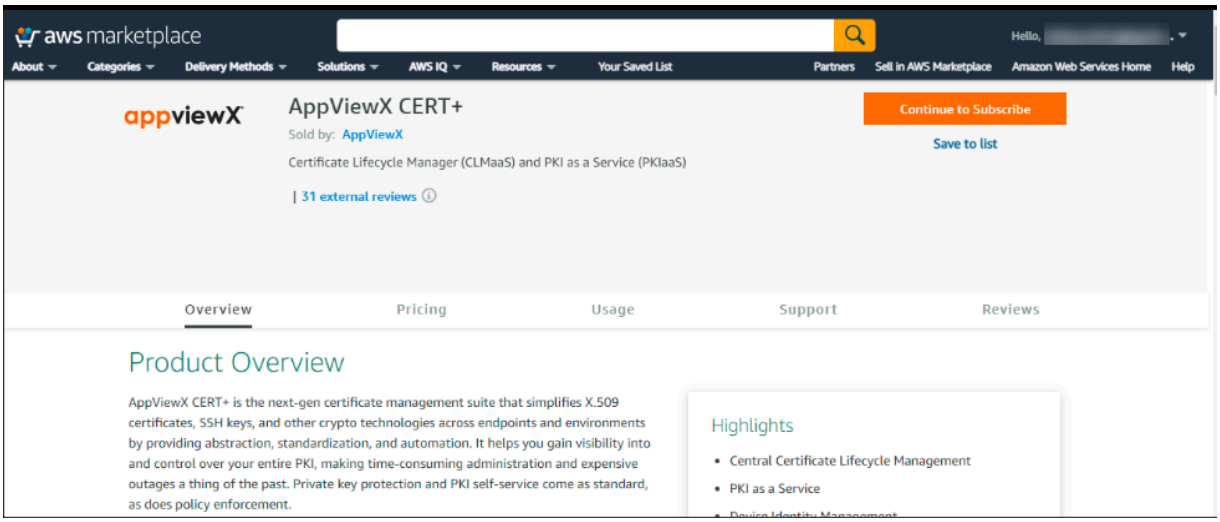
To get started with the CERT+ SaaS free trial, you can sign up via the AWS Marketplace and set up the SaaS by following the steps given below:

- Signing Up for the Free Trial via the AWS Marketplace
- Accessing the AWS Marketplace Sign Up Page
- Entering Details on the Sign Up Page
- Verifying your Email
- Logging in to your SaaS Account
- Getting Started with CERT+ SaaS
- Methods to Set up the AppViewX Cloud Connector
- Working with AppViewX SaaS

Accessing the AWS Marketplace Sign Up Page

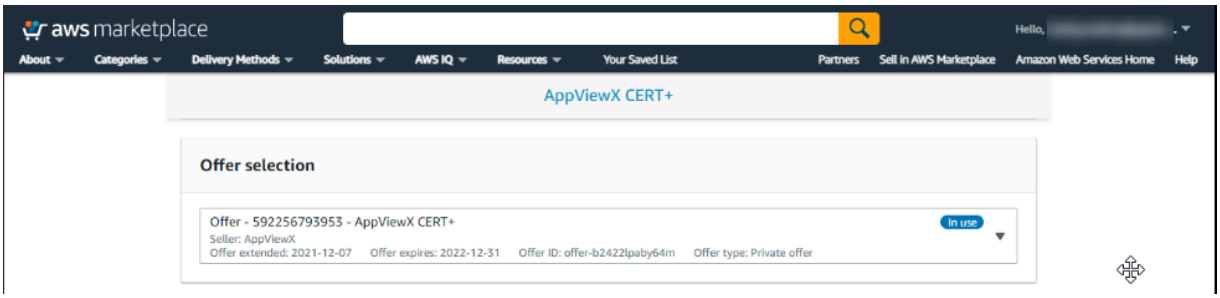
1. Navigate to the AWS Marketplace page.

   The **AppViewX CERT+** page appears.

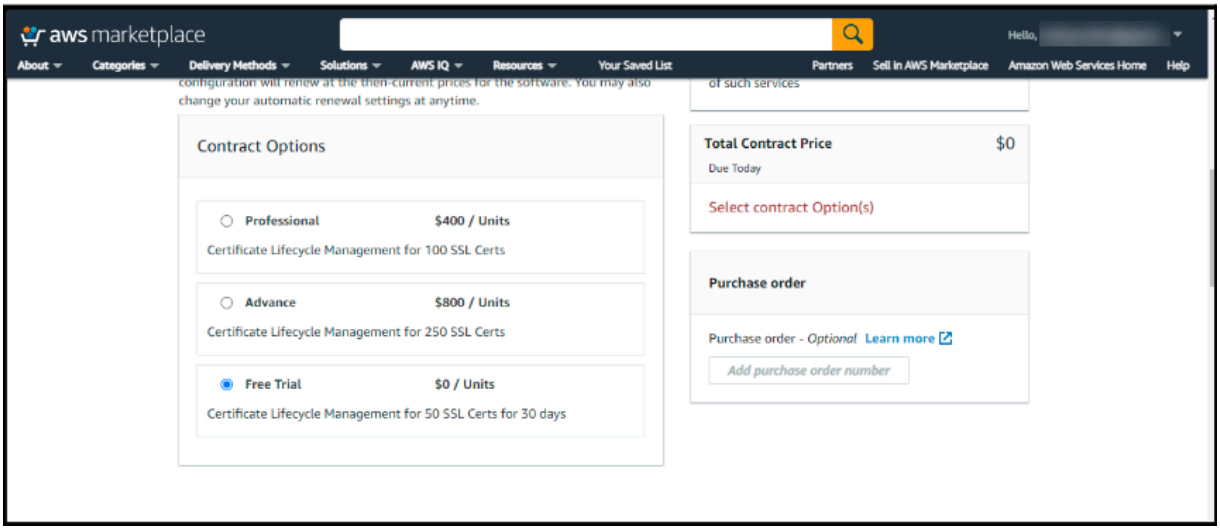


2. Sign into your account or create your account if you are new to AWS Marketplace.
3. Click **Continue to Subscribe.**

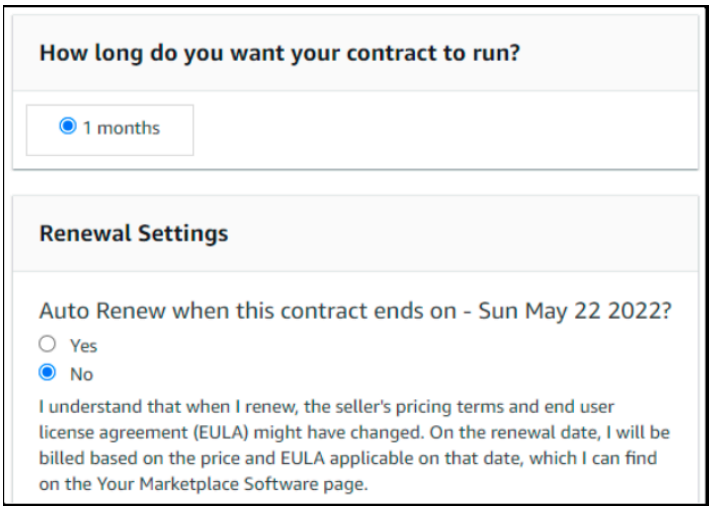
   In case you have a private offer from AppViewX, it will be listed on top of the next page. Make sure you select the private offer and not the public offers.



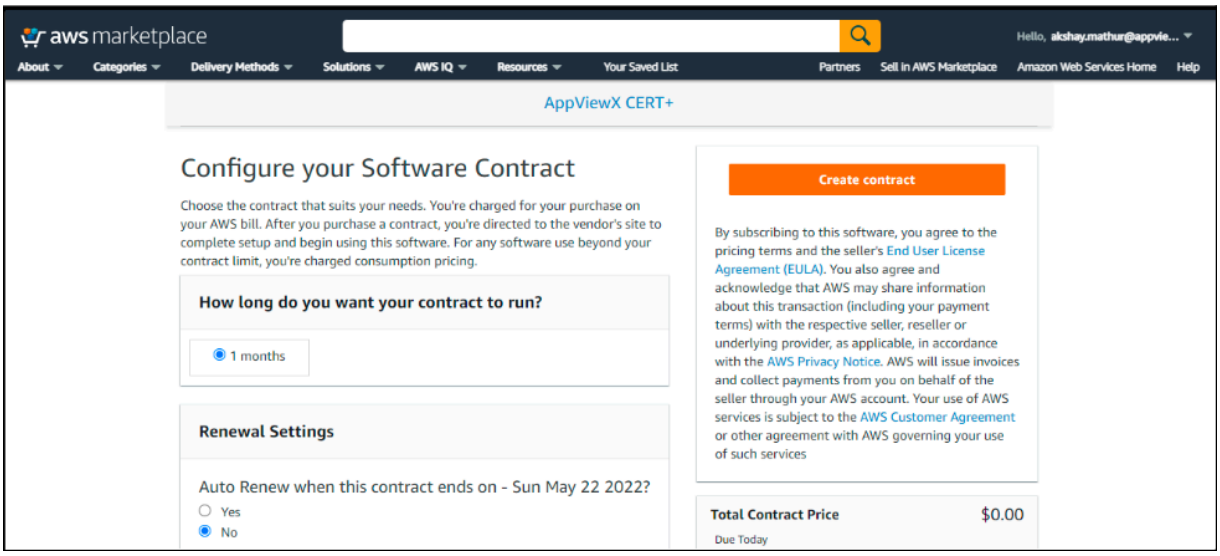
   In case you are here for the first time and do not have a private offer listed, select **Free Trial** option at bottom of the page from the **Pricing** menu.



Set **Auto Renew** to **No.**

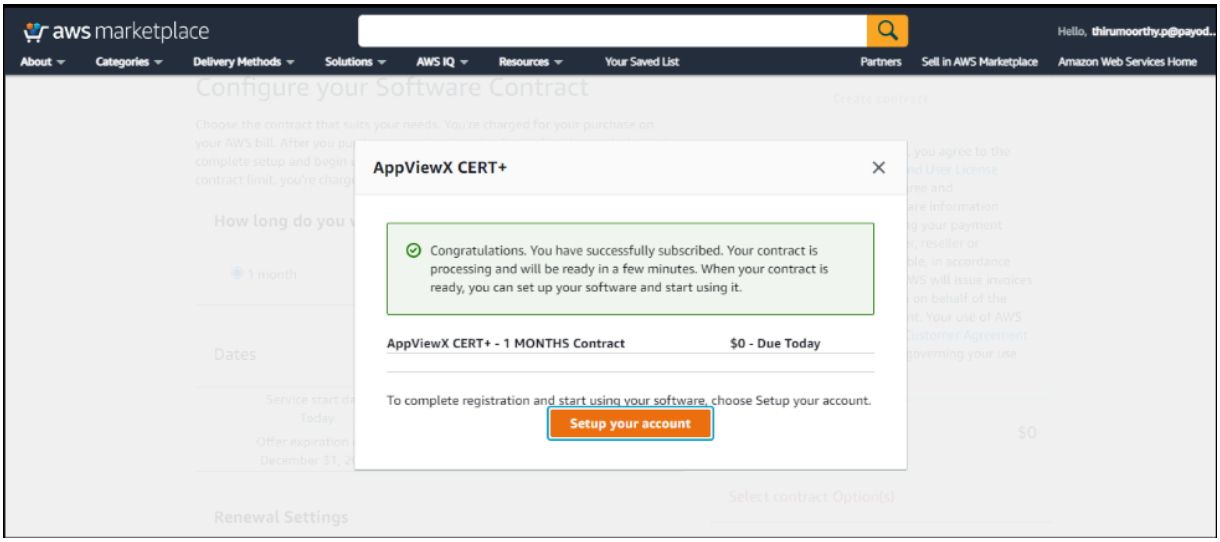


Once all the aforesaid configuration is complete, the **Create contract** button is enabled.



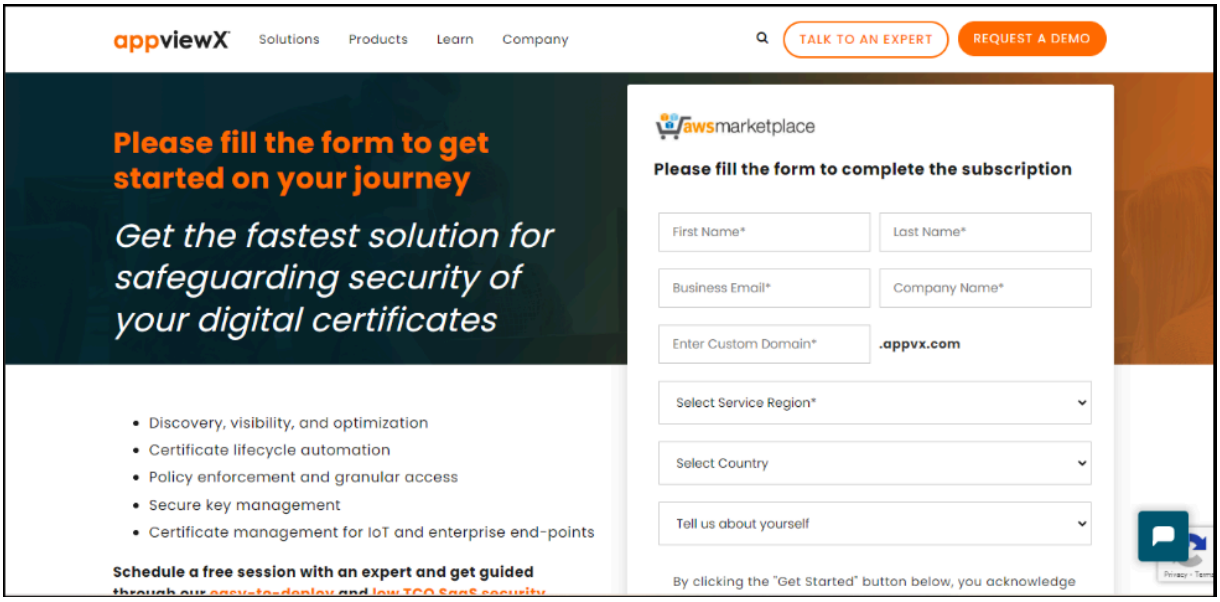
4. Click **Create contract** to create the contract for AppViewX CERT+.

   A confirmation popup message appears.



5. Click **Setup your account** to complete the signup.

You are redirected to the AppViewX SaaS registration page.



Entering Details on the Sign Up Page

1. Enter the fields as described:

Field	Description
First Name*	Enter your first name.
Last name*	Enter your last name.
Business Email*	Enter your business email address.
Company Name*	Enter your company name.
Enter Custom Domain*	By default, the company name is auto-filled. Enter a custom domain if you want to.
Select Service Region*	The service region is where your SaaS account will be set up and localized. You cannot migrate data between regions. Select from one of the service regions:

Field	Description
	• US (Americas) • EMEA • APAC
Select Country*	Select the country from the dropdown list.

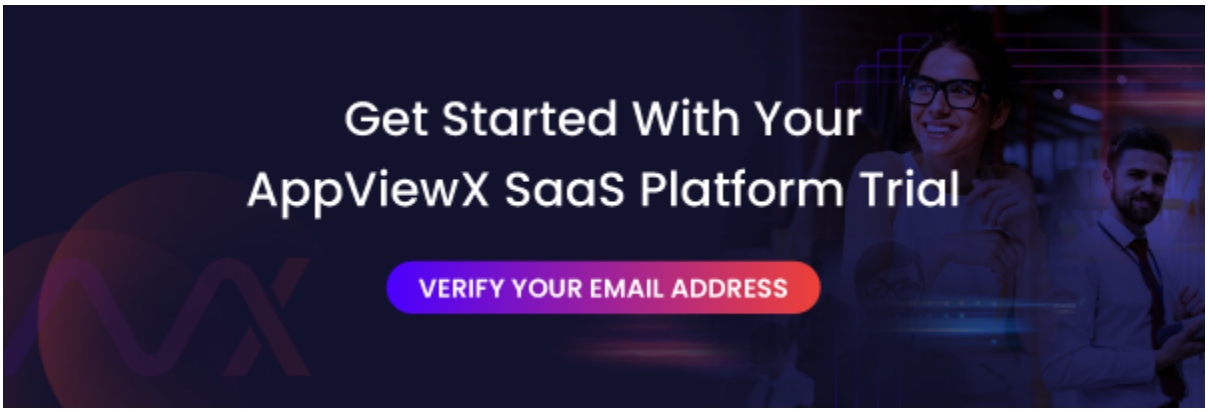
Note:

- Fields marked with the asterisk (*) symbol are mandatory.
- If you are creating a free or trial account, there are email restrictions put in place for security reasons. Email addresses from Gmail.com, Outlook.com, Yahoo.com, and other personalized email addresses are restricted and may not be used for trial account creation purposes.

2. From the **What are you trying to solve** list, select the corresponding checkboxes for your requirements.
3. To acknowledge that you have read and reviewed AppViewX's Terms of Service and their Privacy Policy, select the **By checking this box, I acknowledge...** checkbox.
4. Click **Get Started.**
   The message, *Thank you for signing up for the free trial! You will receive an email from us shortly,* is displayed.

Verifying your Email

On clicking **Get Started,** you will get a verification email to your registered email address. Click **Verify Email Address** to get your SaaS account set-up.



Note:

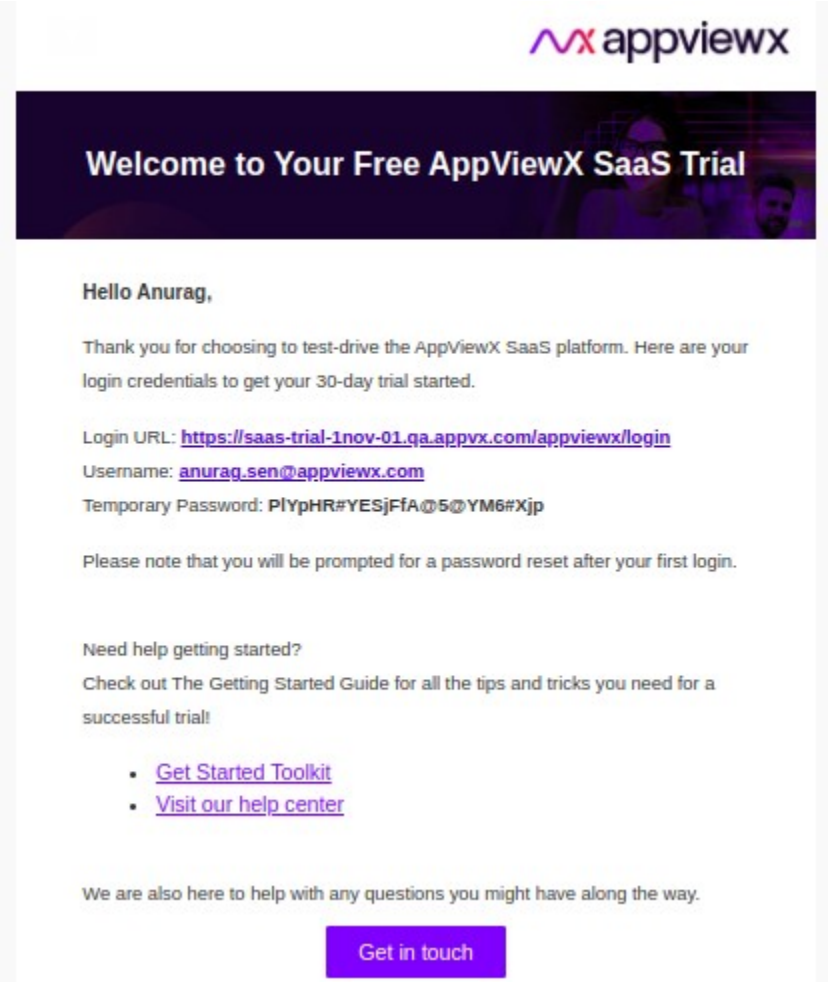
- If you do not see the email in your inbox, then check the Junk/Spam folder. Whitelist the email address so you receive all AppViewX emails in your inbox.
- Confirm your email address within 48hours.

Wait for a couple of minutes until your email address is successfully verified.

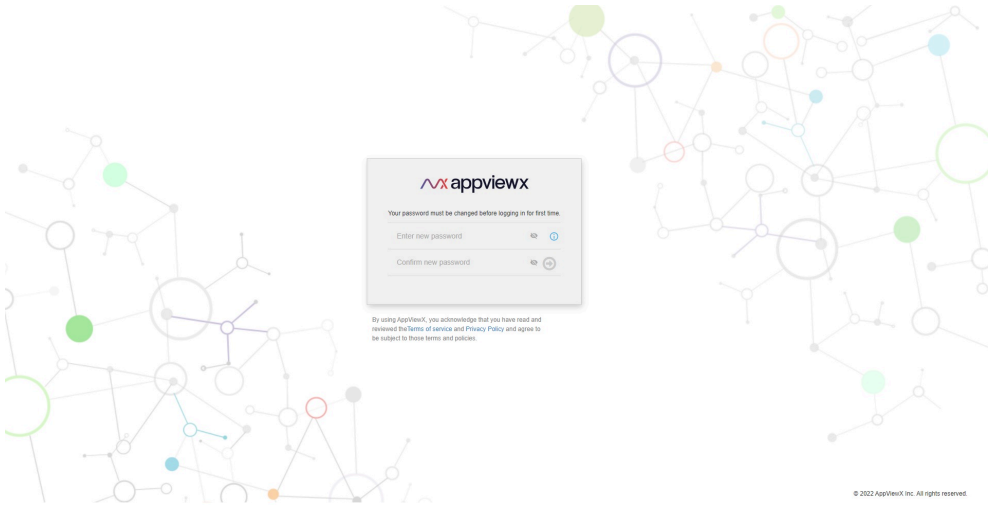
Logging in to your SaaS Account

Based on the sign-up information and region, you will be notified with your SaaS account and temporary credentials.

1. Check your inbox for another email welcoming you to the AppViewX trial along with the login credentials to access your free account.



2. Click the SaaS account URL (for example, **https://tetrik.appvx.com/appviewx/login**). You are redirected to the login page.
3. Login to your SaaS account with the credentials provided in the email. You are asked to change the current password.



4. Reset the default credentials by choosing a new password.

> **Note:** The password must:
> - Have at least one uppercase character
> - Have at least one lowercase character
> - Have one special character such as ~!@#$^&*_-+=|()
> - Have minimum of 6 characters and maximum of 24 characters
> - Not contain user name
> - Not contain more than 3 same characters continuously, for example, aaa
> - Not contain blank space

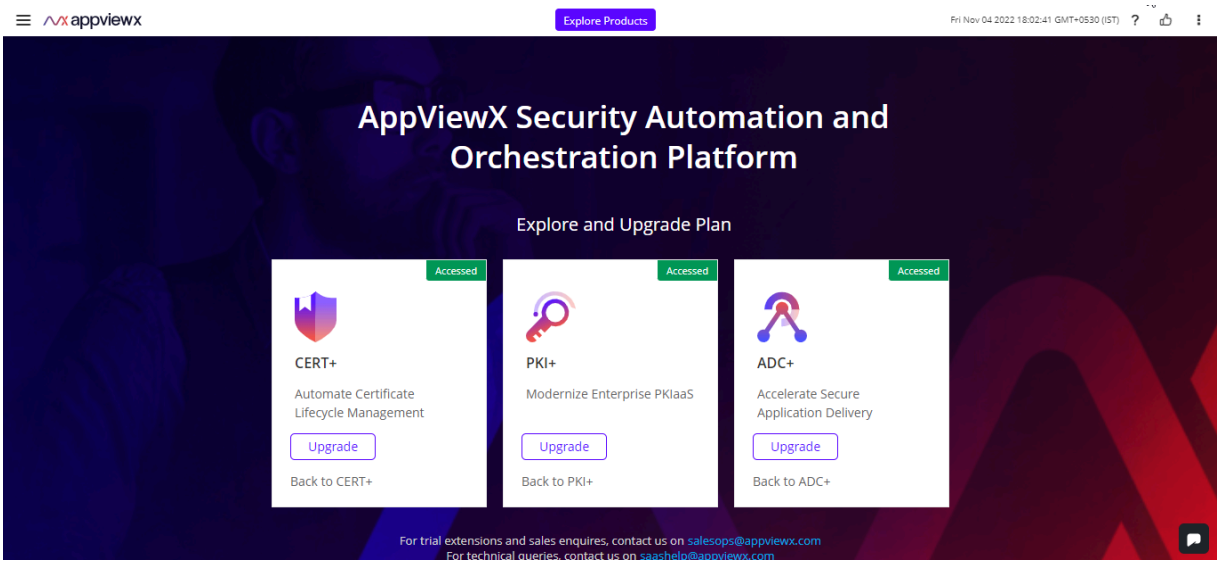
   A message, *Your password has been changed successfully. Please click here to login*, appears.

5. Click **login.**

   The login page appears.

6. Log in using your user name and the new password.

   The AppViewX SaaS landing page is displayed. Click the **Back to <product name>** link to be navigated to that product's individual interface.



Your onboarding is now complete and your account is valid for the next 30 days.

Getting Started with CERT+ SaaS

To get started with CERT+ SaaS, refer to the CERT+ SaaS User Guide.

Methods to Set up the AppViewX Cloud Connector

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

- **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.

- **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).

- Setting up the AppViewX Cloud Connector via the Native OS
- Setting up the AppViewX Cloud Connector via a Virtual Image

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 occurs 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.

> **Note:** If this AppViewX Cloud Connector installation requires configuing a proxy server, click here for instructions.

- Overview
- Step 1: Accessing the Setup Interface
- Step 2: Executing the Prerequisite Check Script
- Step 3: Installing the AppViewX Cloud Connector

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
- Operating System

• Docker Prerequisites
• Server and Network Prerequisites

Hardware

Each AppViewX Cloud Connector instance requires the following minimum configuration:

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

Operating System

- Ubuntu version 20.04
- CentOS version 7.7 and 7.9

Docker Prerequisites

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

Note: Support for rootless Docker is excluded.

For Docker installation instructions, refer to the links below:

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

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

Server and Network Prerequisites

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

  To do this, execute the following sequence of commands:

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

- Ensure that the AppViewX Cloud Connector can establish connectivity with the AppViewX 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 as given below. When connectivity has been established successfully, the command will return the HTTP code **200**.

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

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

```
apt-get install curl
```

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

```
yum install curl
```

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

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

  ```
  sudo ufw status
  ```

  To permanently disable firewalld, execute the command given below:

  ```
  sudo ufw disable
  ```

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

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

  ```
  sudo systemctl status firewalld --now
  ```

  To permanently disable the firewalld, execute the command given below:

  ```
  sudo systemctl disable firewalld --now
  ```

  To restrict other devices from enabling the firewalld, execute the command given below:

  ```
  sudo systemctl mask firewalld --now
  ```

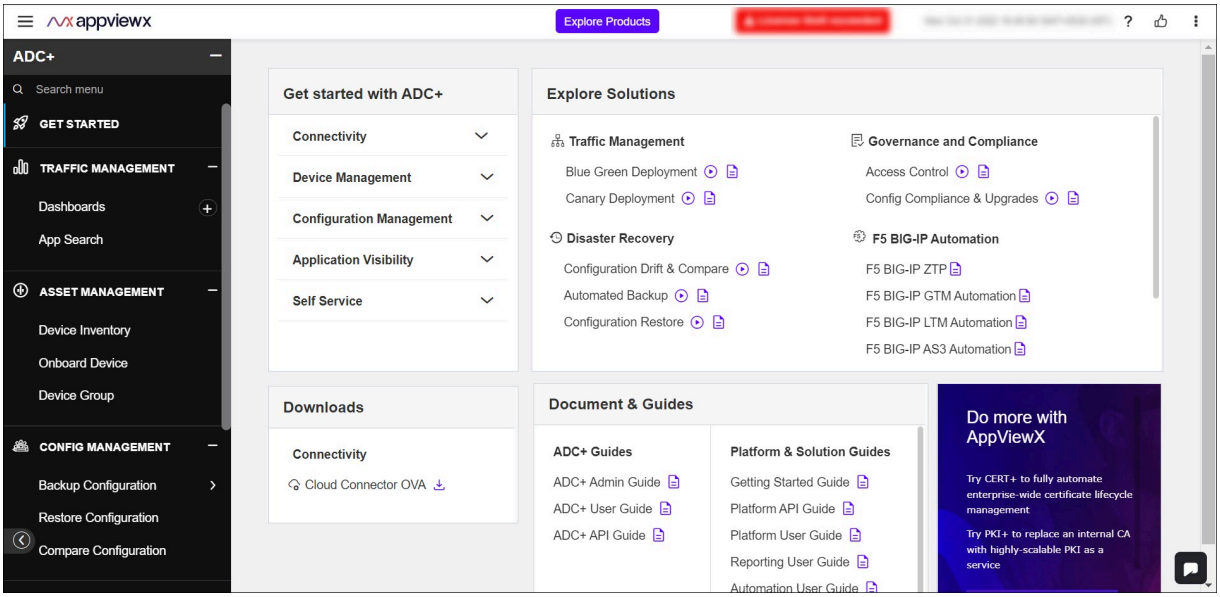
Step 1: Accessing the Setup Interface

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.

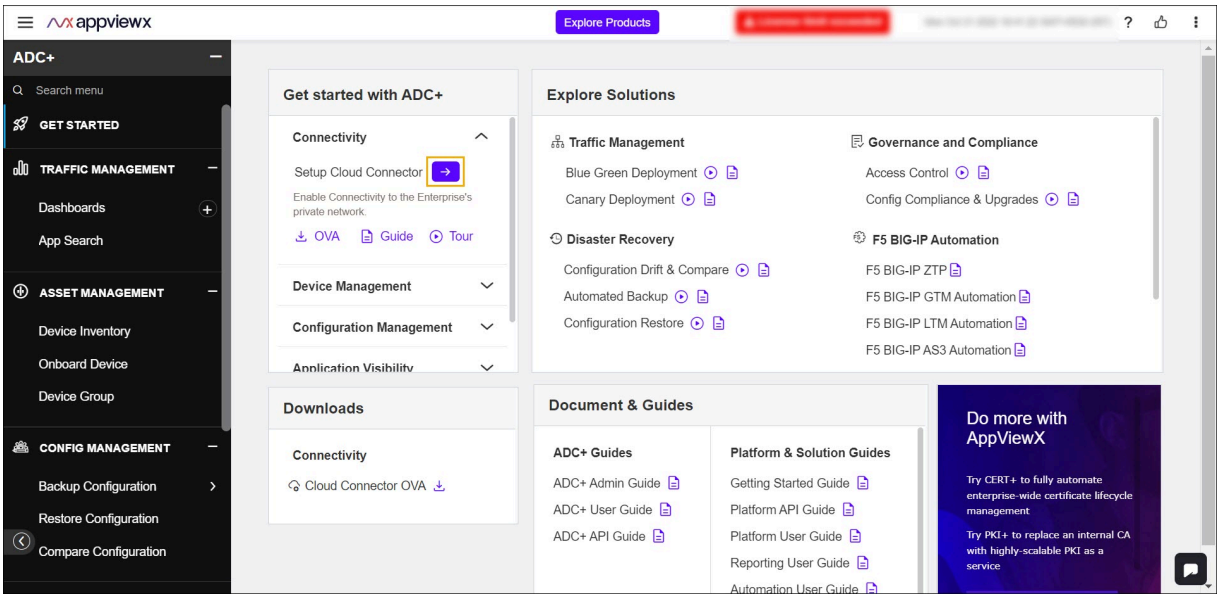
Important: As an additional layer of security, AppViewX issues client certificates to access the AppViewX GUI. The client certificate will be made available as part of the onboarding process. Upload this client certificate to the browser to start accessing the product.

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.



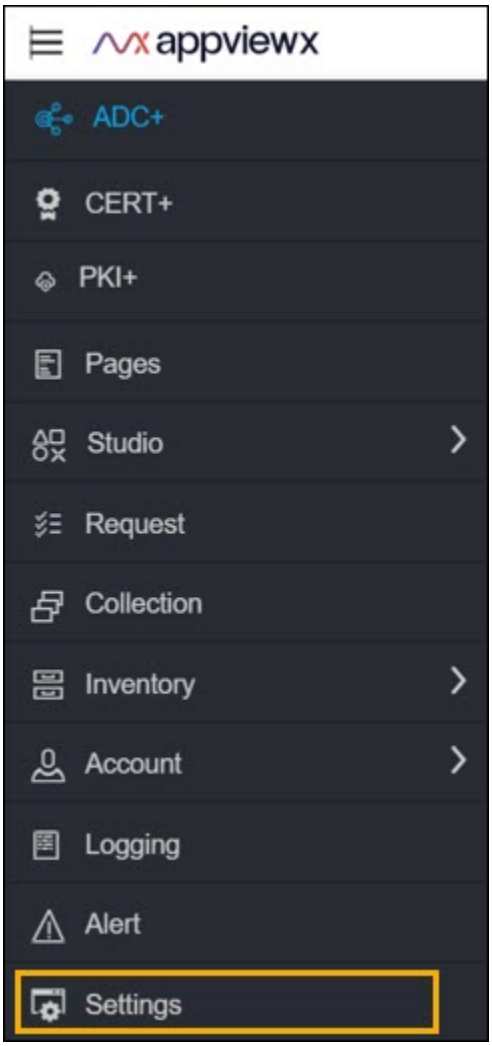
Redirected to the Settings :: Cloud Connector page.

Step 2: 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.

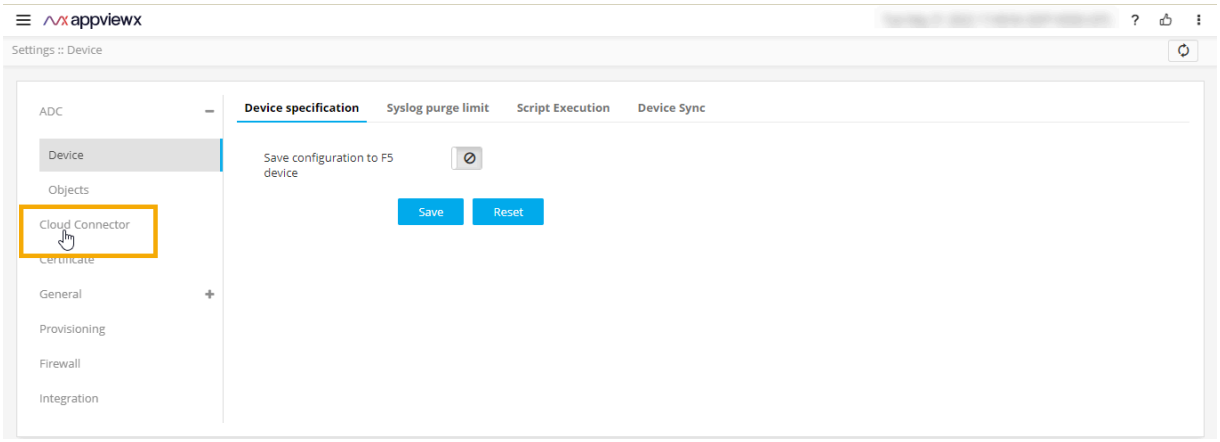
To perform a prerequisite check:

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**.



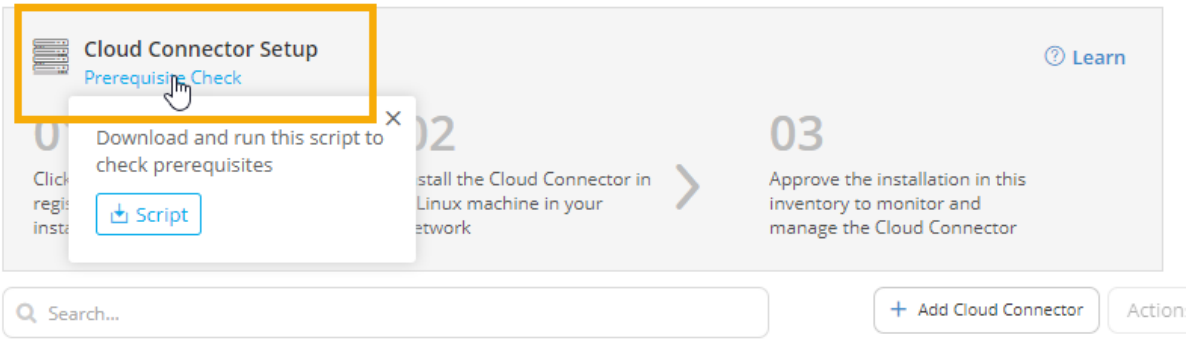
By default, the **Settings :: Device** page is displayed.

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

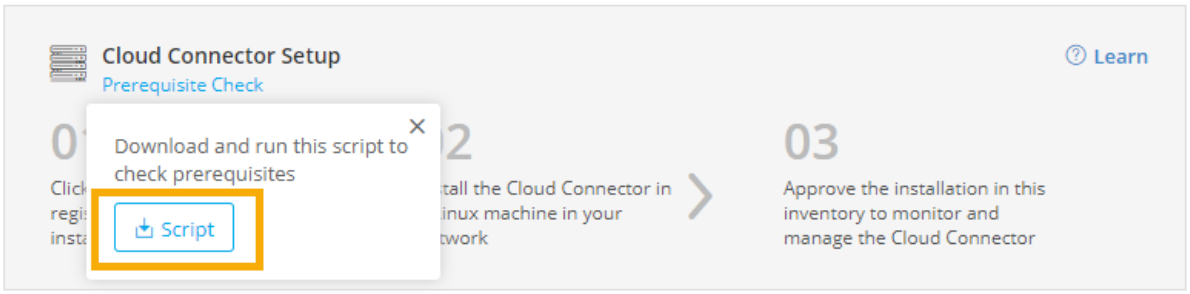


The **Settings :: AppViewX Cloud Connector** page is displayed.

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



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

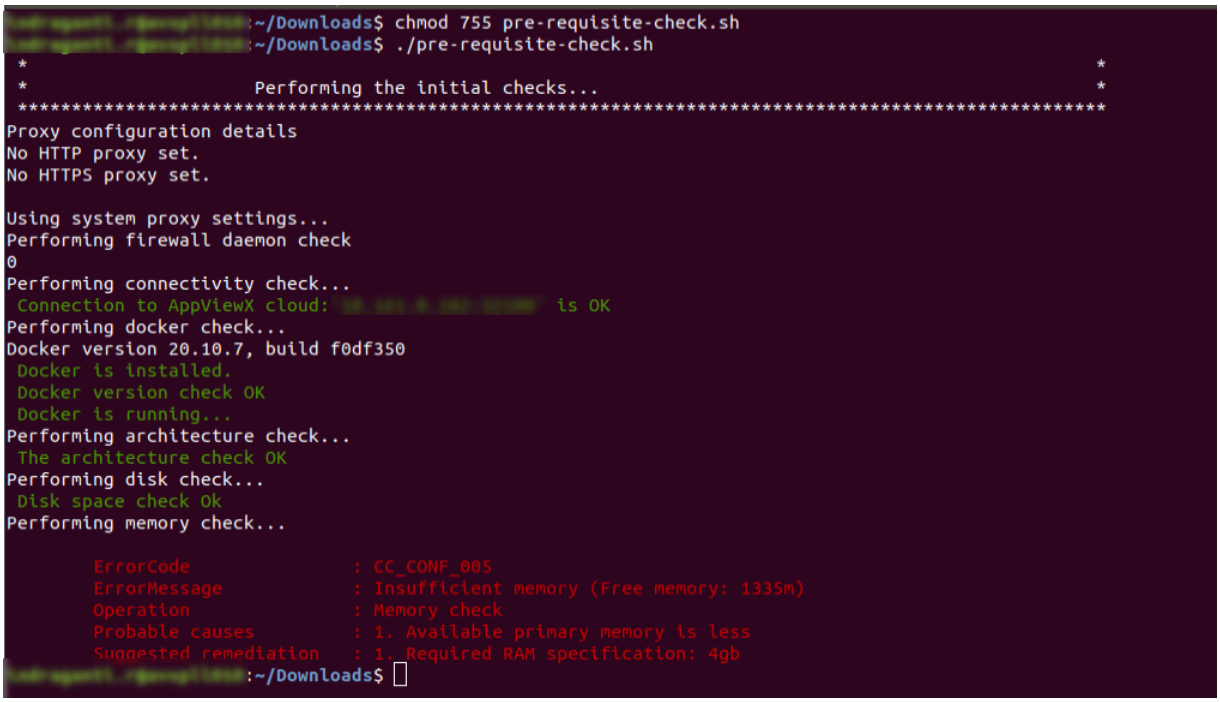


The **pre-requisite-check.sh** script file is downloaded.

6. 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.



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

Step 3: Installing the AppViewX Cloud Connector

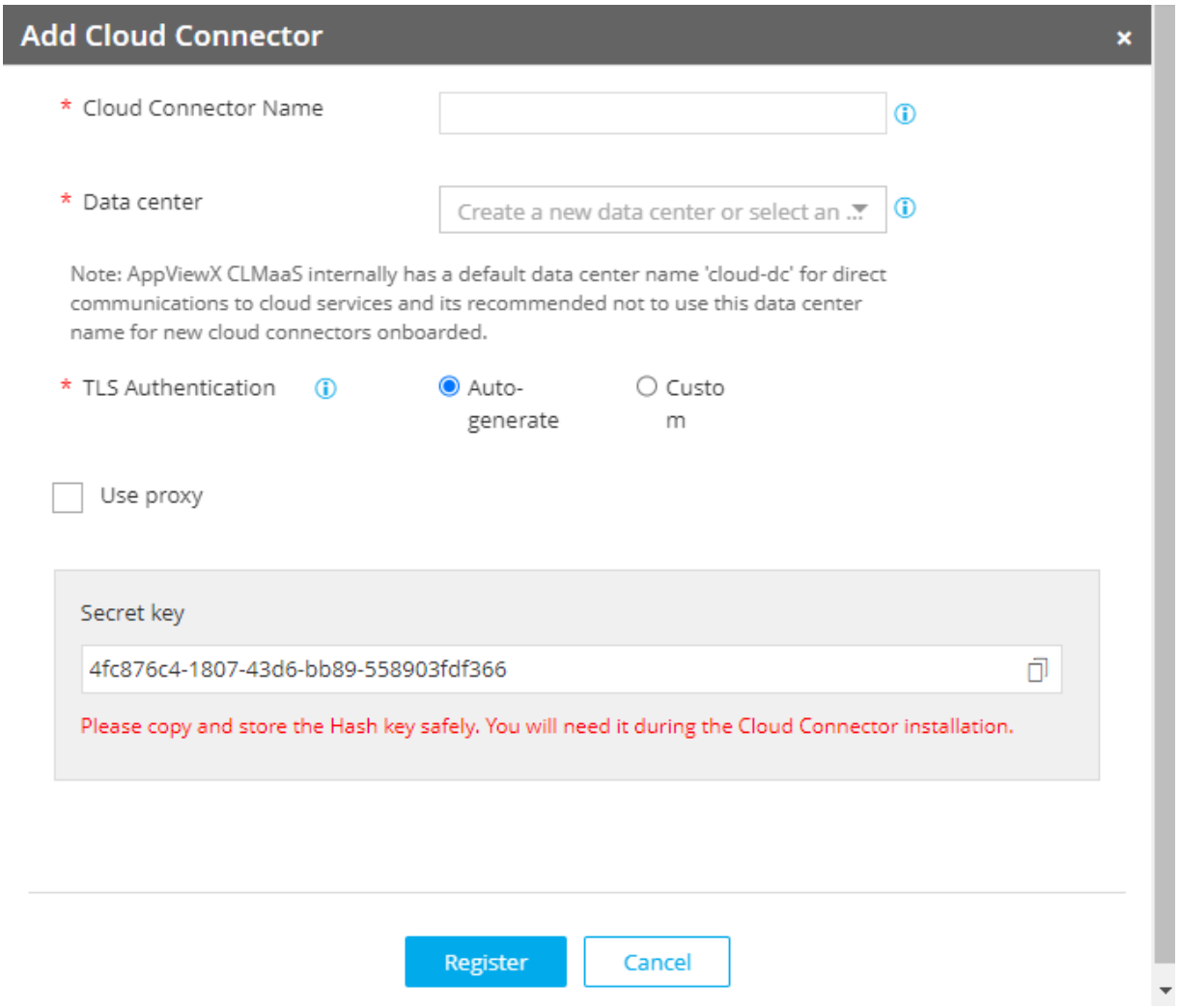
The process of deploying the AppViewX Cloud Connector involves the steps listed below. The substeps for each of these steps are outlined in the subsequent sections.

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



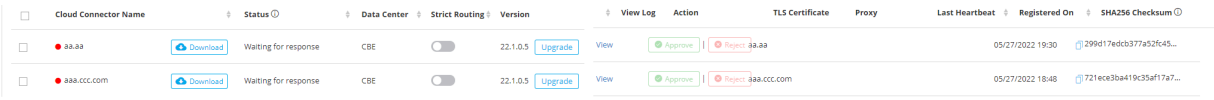
2. In the **Add Cloud Connector** action pane, enter the following details (sample values are shown in an image below the table):

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

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

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

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



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

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

.
5. 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.

1. 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`
2. 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:
kubectl cluster-info
Cluster setup is completed. Will start the deployment shortly...
Importing the required images...
[36mINFO[0m[0000] Importing image(s) into cluster 'cc'
[36mINFO[0m[0000] Importing images from 1 tarball(s)...
[36mINFO[0m[0000] Importing images '[/home/appviewx/CCTEST/deps/tools/mid-server-docker-image/avx-mid-server-base-22.1.0.0.tar]' into node
 'k3d-cc-server-0'...
[36mINFO[0m[0024] Successfully imported image(s)
[36mINFO[0m[0024] Successfully imported 1 image(s) into 1 cluster(s)
Import in progress...
[36mINFO[0m[0000] Importing image(s) into cluster 'cc'
[36mINFO[0m[0000] Importing images from 1 tarball(s)...
[36mINFO[0m[0000] Importing images '[/home/appviewx/CCTEST/deps/tools/mid-server-docker-image/k3d-tools-5.2.2.tar]' into node 'k3d-cc-server-0'...
[36mINFO[0m[0005] Successfully imported image(s)
[36mINFO[0m[0005] Successfully imported 1 image(s) into 1 cluster(s)
Import in progress...
[36mINFO[0m[0000] Importing image(s) into cluster 'cc'
[36mINFO[0m[0000] Importing images from 1 tarball(s)...
[36mINFO[0m[0000] Importing images '[/home/appviewx/CCTEST/deps/tools/mid-server-docker-image/rancher-mirrored-coredns-coredns-1.8.6.tar]' into
 node 'k3d-cc-server-0'...
[36mINFO[0m[0007] Successfully imported image(s)
[36mINFO[0m[0007] Successfully imported 1 image(s) into 1 cluster(s)
```

```
[36mINFO[0m[0000] Importing image(s) into cluster 'cc'
[36mINFO[0m[0000] Importing images from 1 tarball(s)...
[36mINFO[0m[0000] Importing images '[/home/appviewx/CCTEST/deps/tools/mid-server-docker-image/rancher-local-path-provisioner-v0.0.21.tar]' into node
 'k3d-cc-server-0'...
[36mINFO[0m[0004] Successfully imported image(s)
[36mINFO[0m[0004] Successfully imported 1 image(s) into 1 cluster(s)
[36mINFO[0m[0000] Importing image(s) into cluster 'cc'
[36mINFO[0m[0000] Importing images from 1 tarball(s)...
[36mINFO[0m[0000] Importing images '[/home/appviewx/CCTEST/deps/tools/mid-server-docker-image/rancher-mirrored-pause-3.6.tar]' into node
 'k3d-cc-server-0'...
[36mINFO[0m[0003] Successfully imported image(s)
[36mINFO[0m[0003] Successfully imported 1 image(s) into 1 cluster(s)
Deploying the Cloud Connector...
NAME: avx-mid-server-starter
LAST DEPLOYED: Mon May 30 15:51:13 2022
NAMESPACE: cc
STATUS: deployed
REVISION: 1
NOTES:
1. It may take a couple of minutes for the Cloud Connector to be up.
  kubectl get pod --namespace cc
**************************************************************************
*    Congratulations!!! The installation completed successfully.    *
*    Please wait till the Cloud Connector is up and running.       *
**************************************************************************
(1%) Cloud Connector status: Running
[32m Cloud Connector is up and running. (B[m
```

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

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

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

When the platform component download is in progress, it is indicated by the symbol prefixed

to the platform component version number in the AppViewX Cloud Connector inventory details  . A completed download/upgrade is indicated by the  symbol in the same location  .

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

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 a Virtual Image

The AppViewX Virtual Image is an Open Virtual Appliance (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.

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 configuing a proxy server, click here for instructions.

To set up the AppViewX Cloud Connector via a virtual image:

- 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**.

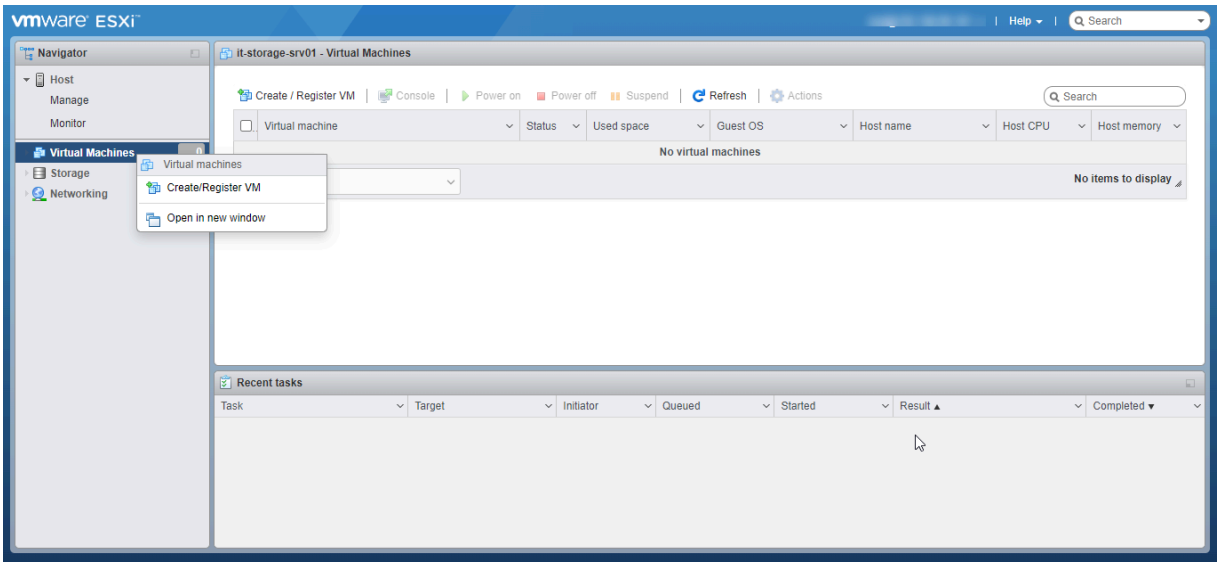
- Downloading the OVA Release Package
- Installing the AppViewX OVA

Downloading the OVA Release Package

From the download URL, download the release package in the OVA format.

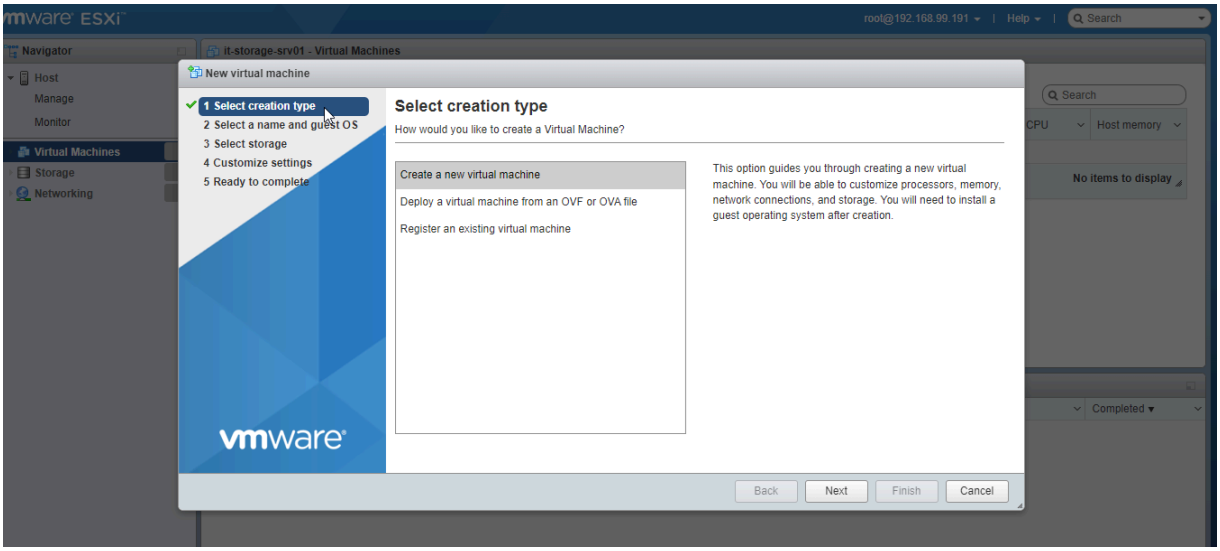
Installing the AppViewX OVA

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

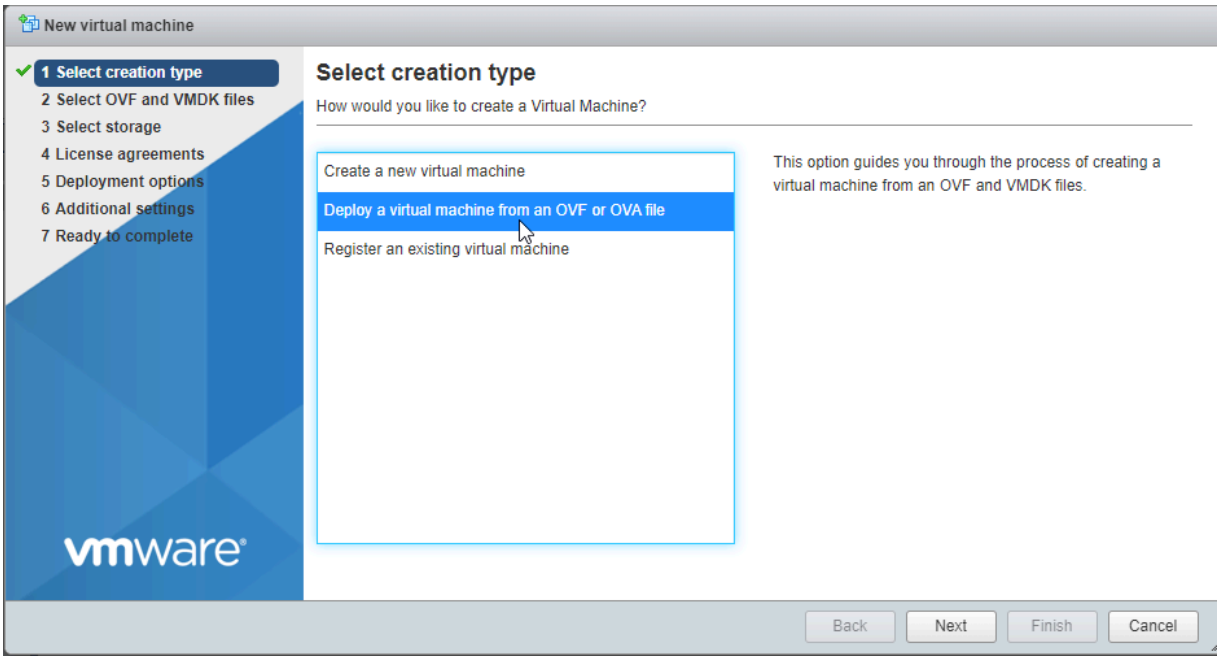


The **New Virtual machine** window is displayed.

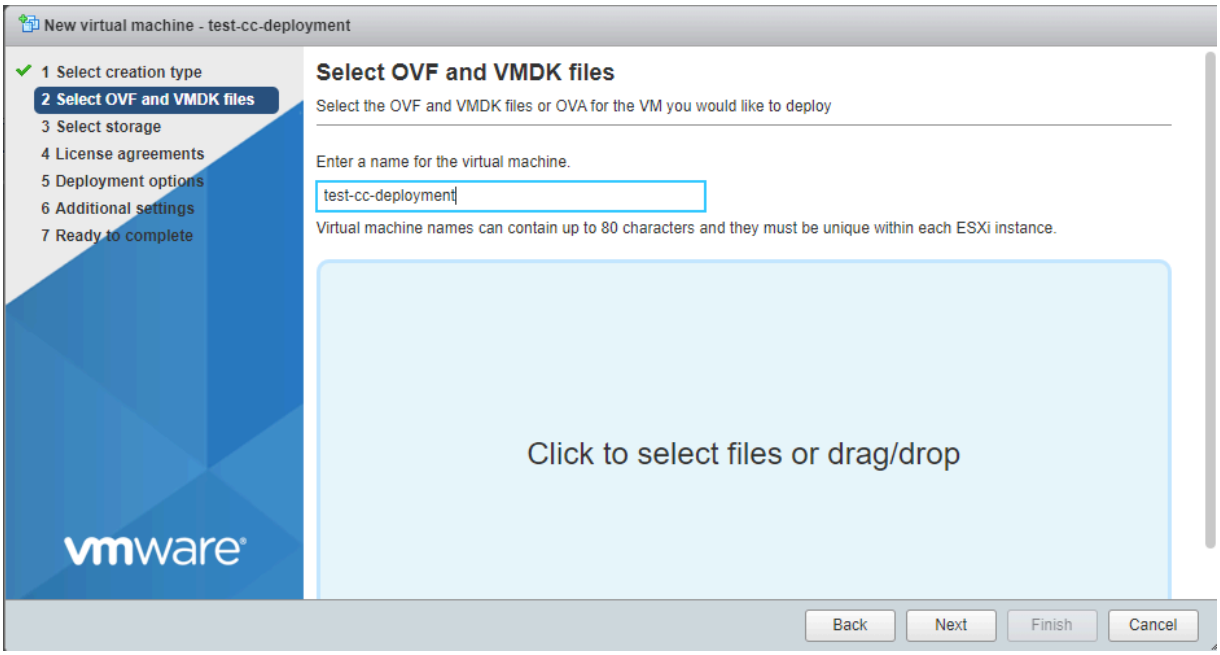
4. From the navigation pane in the left, select **Select creation type**.



5. In the **Select creation type** window, select the **Deploy a virtual machine from an OVA or OVF file** option.



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

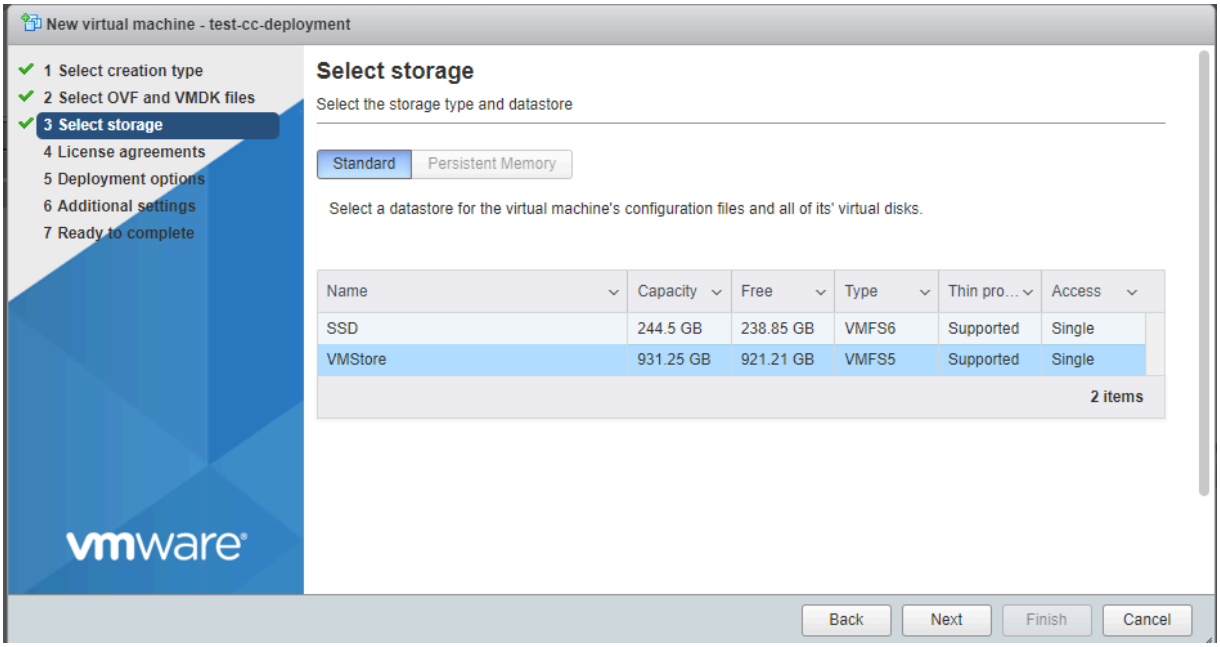


   a. Enter a name for the virtual machine.

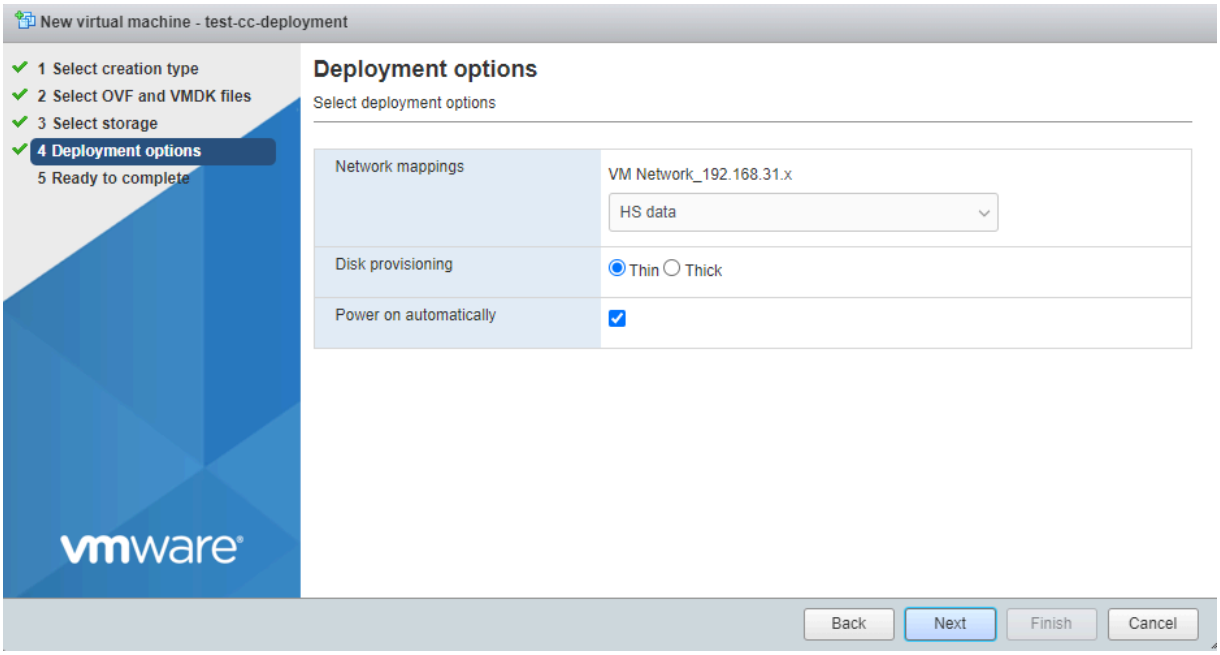
      For the purpose of this document, we will name it **test-cc-deployment**.

   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**.

8. Click **Next**.
9. 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.



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

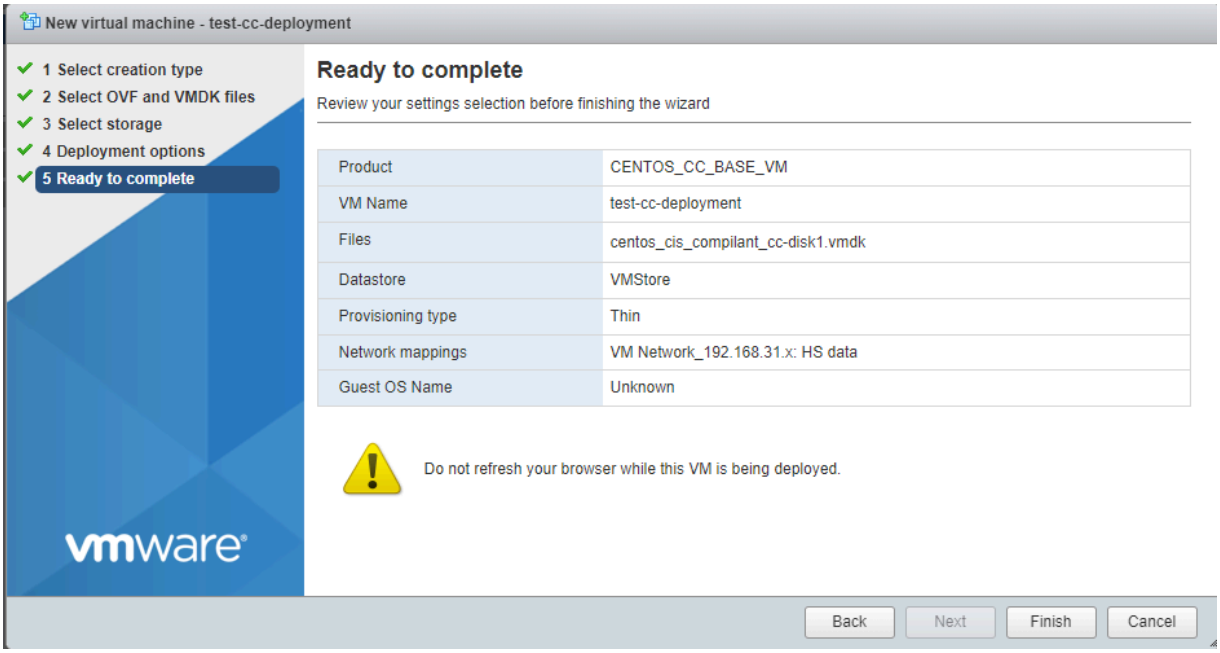


a. Select the network mapping.

b. Select the disk provisioning required.

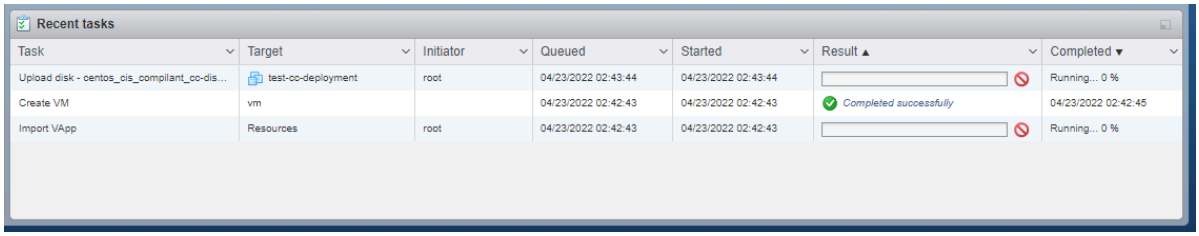
c. Select the **Power on automatically** checkbox.

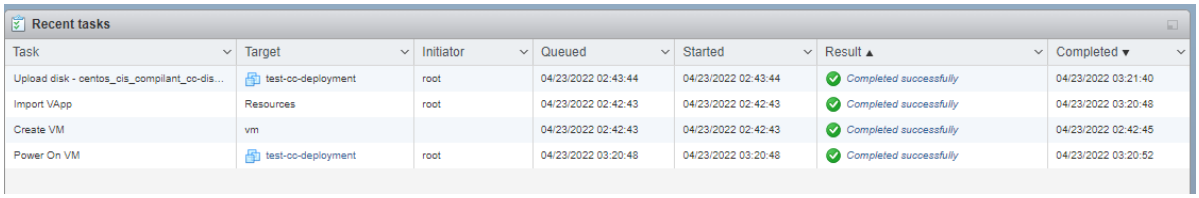
12. Click **Next**.
13. In the **Ready to complete** window, review your settings.



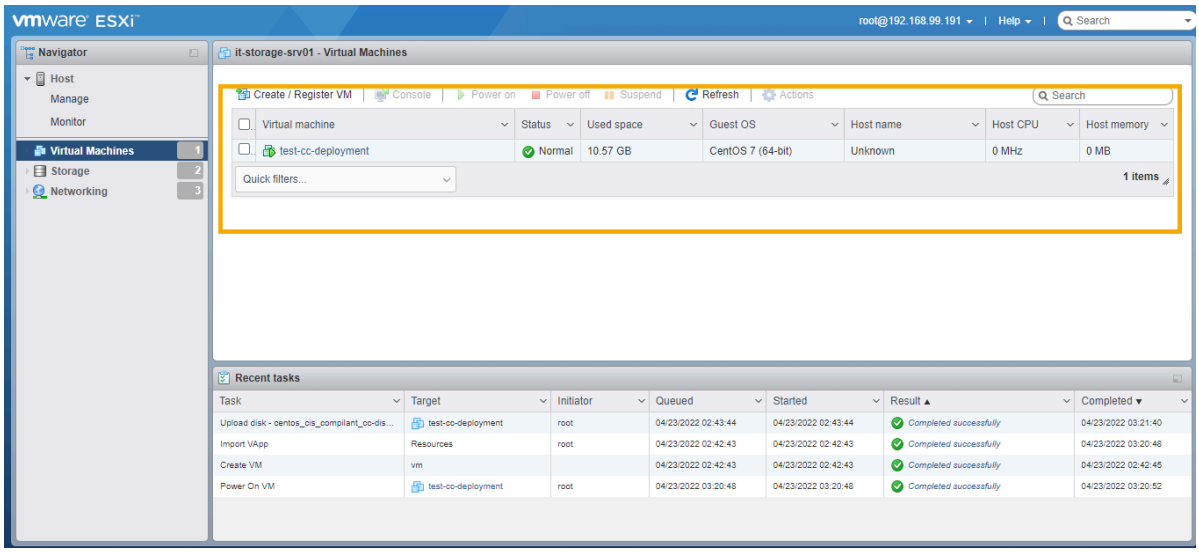
14. Click **Finish**.

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





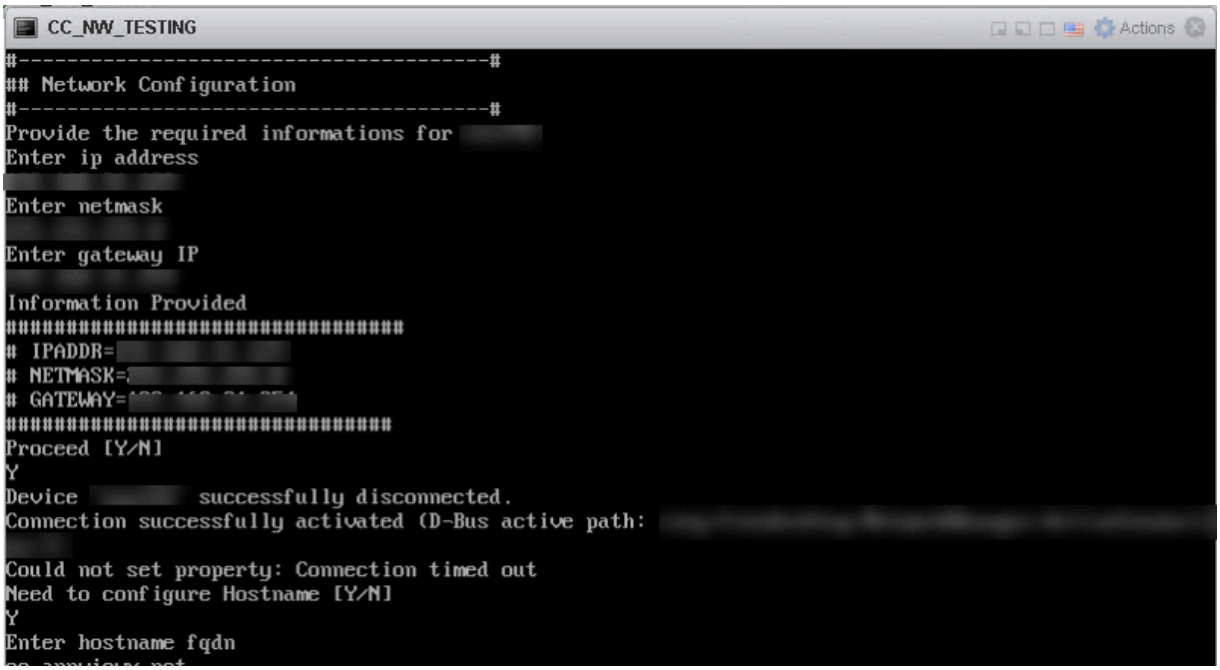
- On successful completion of the OVA deployment, the new virtual machine is displayed in the **Virtual Machines** inventory. For each virtual machine in the inventory, the following details are displayed:



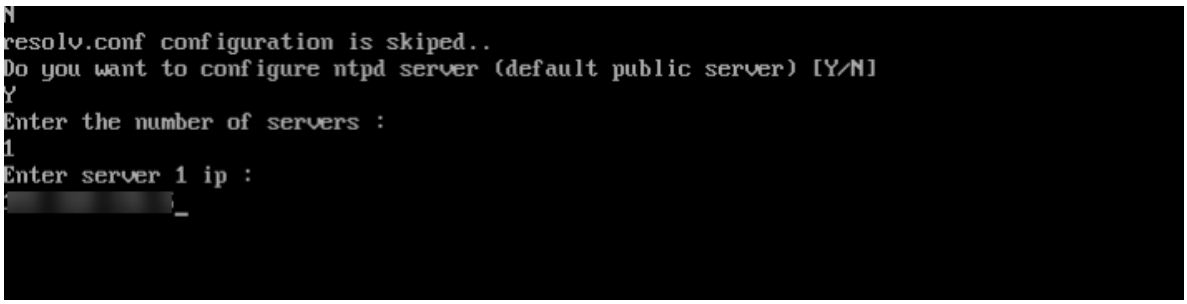
15. From the **Virtual Machines** inventory, click the virtual machine just added.
    The terminal window for the virtual machine is displayed. The script for configuring the network IP is executed automatically.
16. To configure the IP address, when prompted, enter the required values for the following 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:



17. 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**.
18. To configure an NTP server(s):
    a. When prompted, **Do you want to configure ntpd server (default public server)**press **Y**.



    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**.
19. After the script is executed, to login to the VM, when prompted, use one of the following set of credentials:
    • For the appviewx user
      • Username: **appviewx**
      • Password: **XApp23**
    • For the root user:

> **Note:** The root user access is required for maintaining the OS configuration and for patching security updates.

- Username: **root**
- Password: **PlMQaZ23**

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

20. 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:

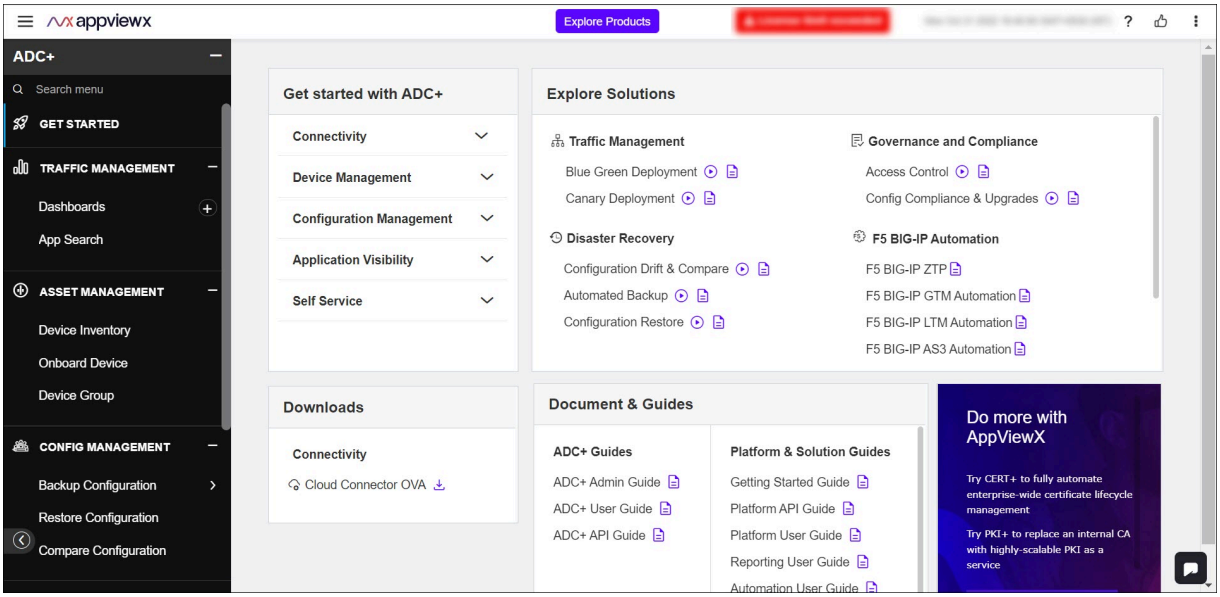
Step 2: Accessing the Setup Interface

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.

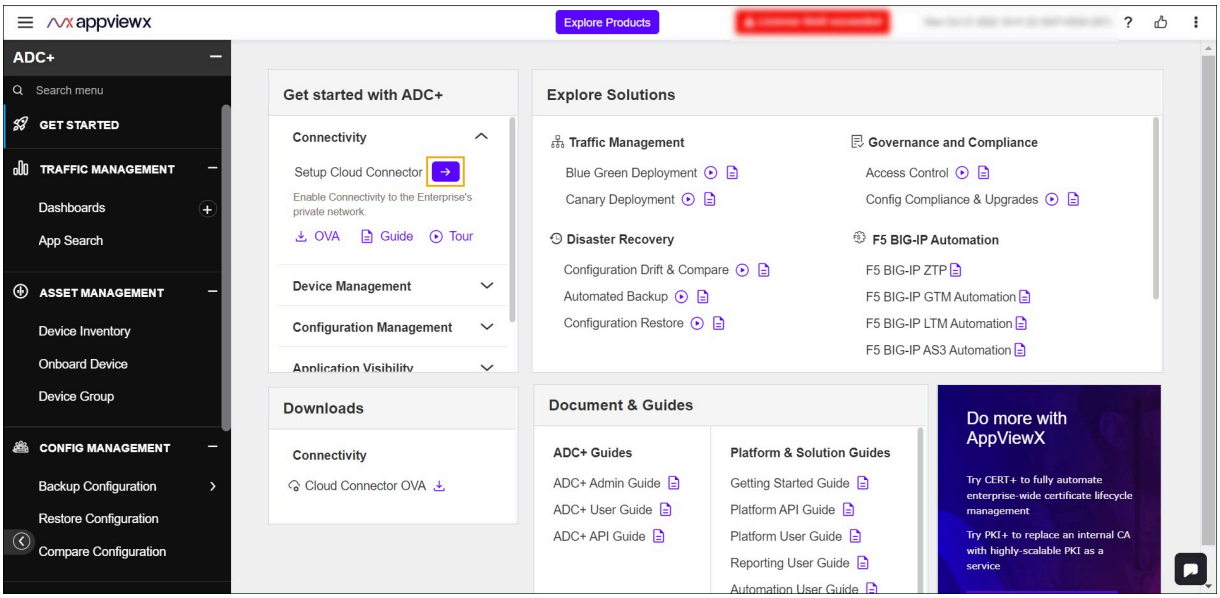
> **Important:** As an additional layer of security, AppViewX issues client certificates to access the AppViewX GUI. The client certificate will be made available as part of the onboarding process. Upload this client certificate to the browser to start accessing the product.

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.



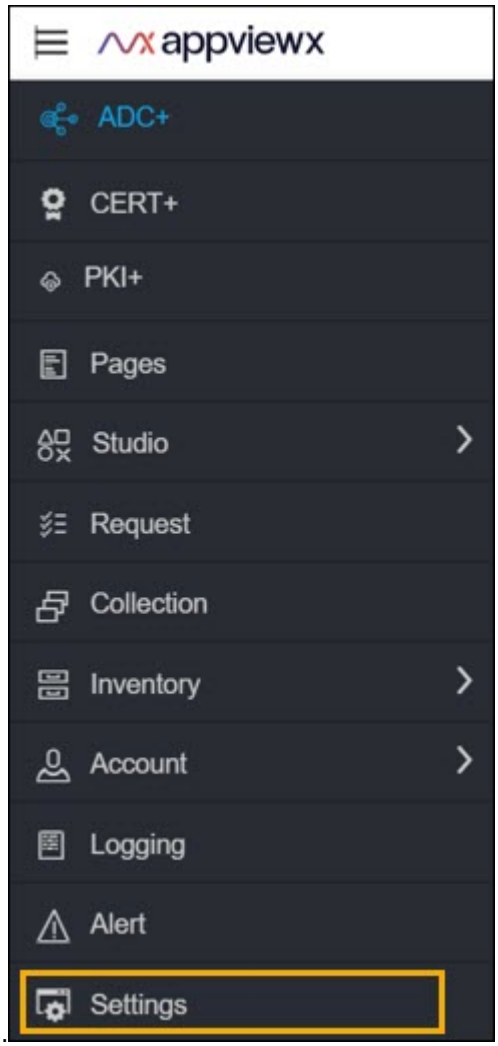
Redirected to the Settings :: Cloud Connector page.

Step 3: Executing the Prerequisite Check Script

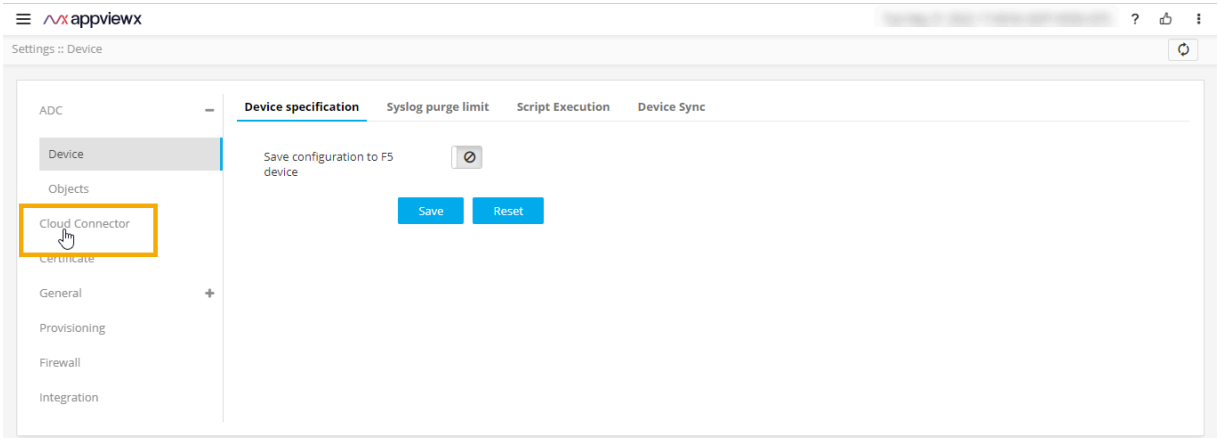
To simplify compliance to the AppViewX Cloud Connector installation prerequisites, you can execute a script to identify and rule out any deviations from the requirements.

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**.

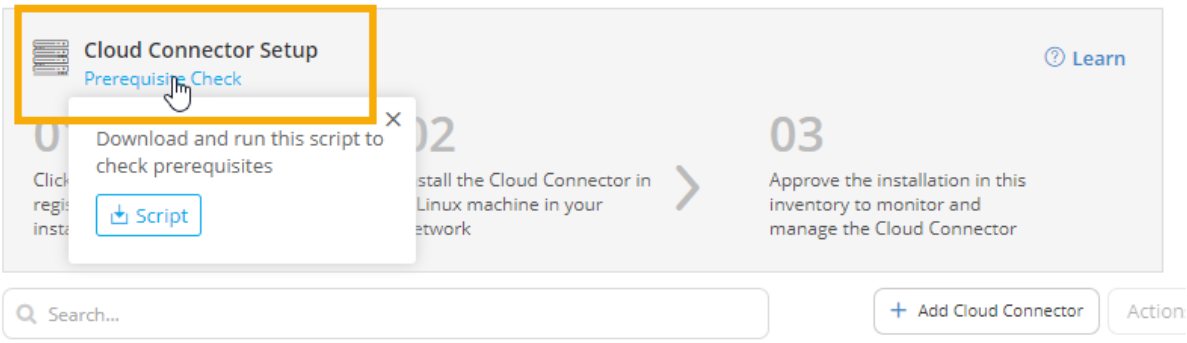


   By default, the **Settings :: Device** page is displayed.
3. From the left navigation pane, select **AppViewX Cloud Connector**.

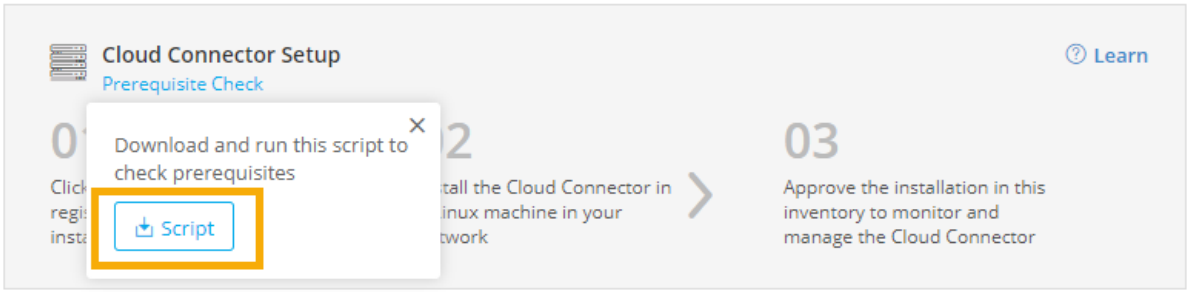


The **Settings :: AppViewX Cloud Connector** page is displayed.

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



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



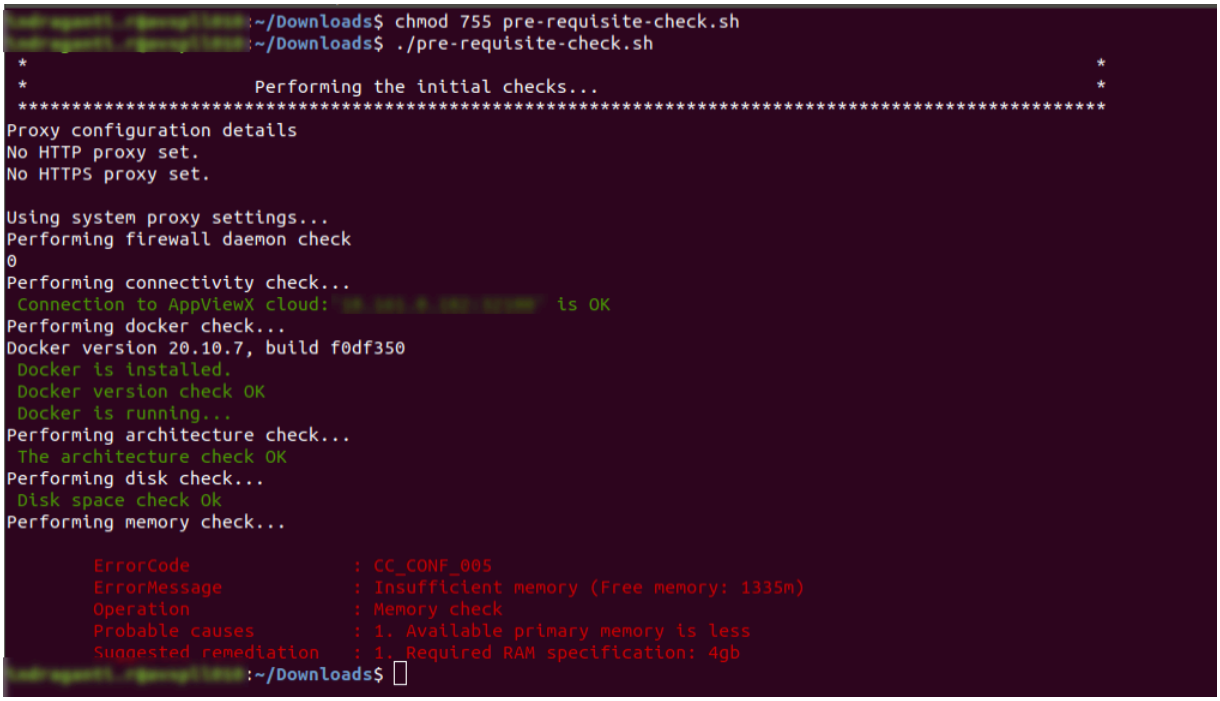
The **pre-requisite-check.sh** script file is downloaded.

6. From the command line interface, navigate to the node where the AppViewX Cloud Connector will be installed.

7. Convert the downloaded script file into an executable file using the chmod command, as shown below:`chmod 755 pre-requisite-check.sh`
8. Execute the **.sh** prerequisite check script file.

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

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



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

Step 4: Installing the AppViewX Cloud Connector

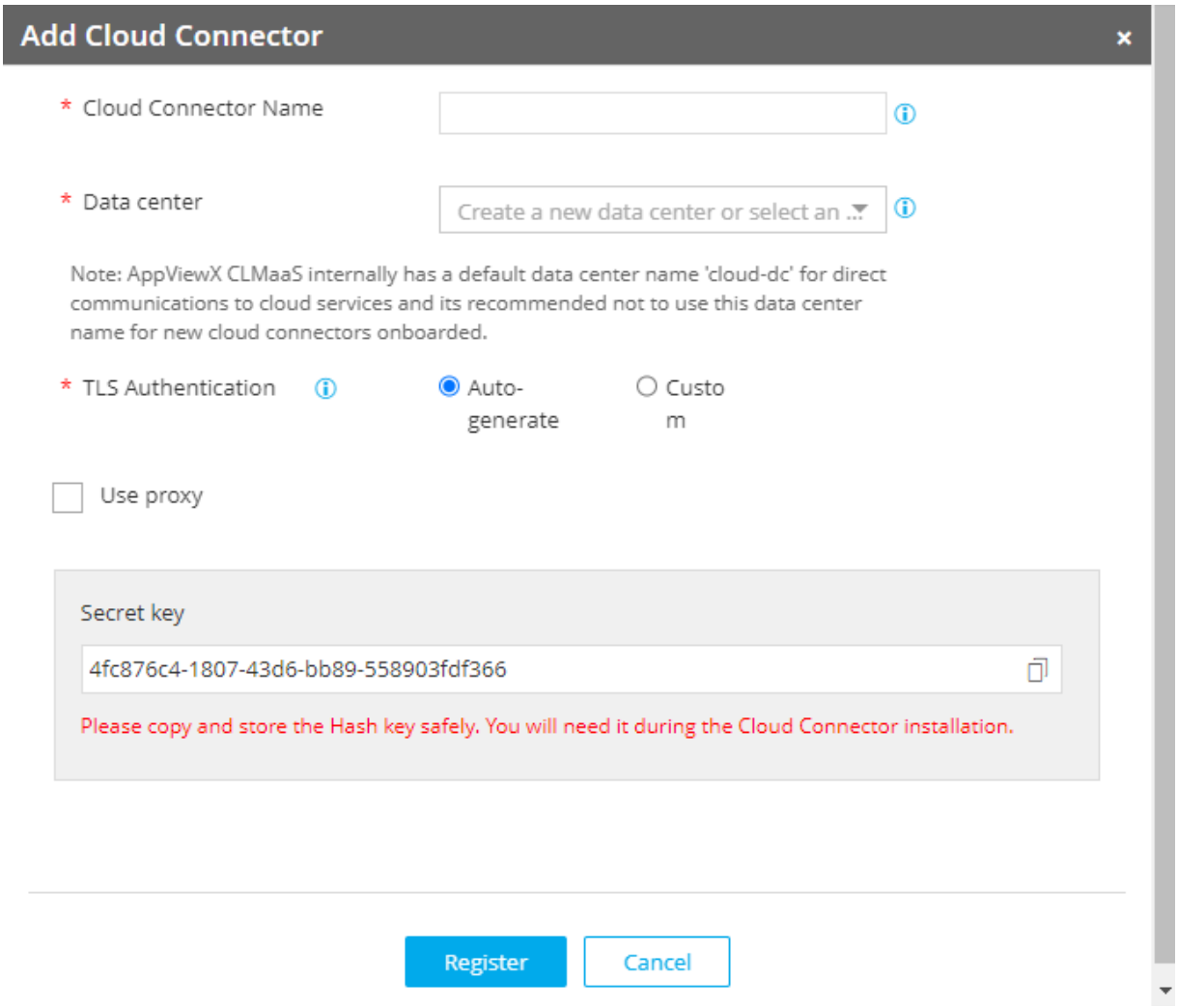
The process of deploying the AppViewX Cloud Connector involves the steps listed below. The substeps for each of these steps are outlined in the subsequent sections.

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



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

Field	Description
Cloud Connector Name*	FQDN of the machine where the AppViewX Cloud Connector is to be installed
Data center*	Name of the data center where the AppViewX Cloud Connector is to be installed **Note:** The AppViewX SaaS, internally, has a default data center named **cloud-dc**, for direct communications to cloud services. It is recommended that you do not this data center name for new the AppViewX Cloud Connectors onboarded.

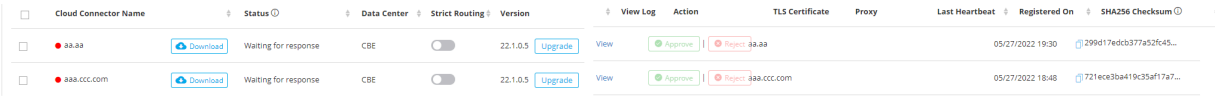
Authentication Type	• Select the authentication type from the following options:
	  • **TLS**
	  • **MTLS**
	
	• To auto-generate a TLS certificate, select **Auto-generate** (default selection).
	
	Automatically, the certificate is generated using the AppViewX CA.
	
	Note: The created certificate is available in the certificate inventory. You can:
	  • Assign this certificate to a certificate group
	  • Configure a certificate expiry alert for this certificate group from the **Server Certificate** dashboard, using the **Certificate Summary Report** widget settings
	
	• To enter details of a custom TLS certificate, select **Custom**.
	
	The **TLS Certificate Password** and **Custom TLS Certificate** fields are displayed. The instructions for filling these fields are given below.

TLS Certificate Password*	**Note:** This field is displayed only if you have selected to enter details of a Custom TLS certificate in the **TLS Authentication** field. Password of the TLS certificate (that will be uploaded in the next step)

Field	Description
	Note: This is a mandatory field if a Custom TLS certificate is uploaded. AppViewX supports only password-protected Custom TLS certificates.
Custom TLS Certificate	**Note:** This field is displayed only if you have selected to enter details of a Custom TLS certificate in the **TLS Authentication** field. To upload a custom TLS certificate: a. To navigate to the location of the custom TLS certificate, click within the field. b. Select the certificate file. c. Click **Open**. d. To upload the custom TLS certificate selected, click **Upload**. **Note:** AppViewX supports only password-protected Custom TLS Certificates.
Use proxy	To use a proxy server for the deployment: a. Select the **Use proxy** checkbox. b. To select a preconfigured proxy (for the selected data center), from the **Select Proxy** dropdown list, select a proxy server. **OR** To create a new proxy server setting: a. Use the Click here option shown below the **Select Proxy** dropdown list. b. For steps to create a new proxy server setting, click here.
Secret key	A unique key for the AppViewX Cloud Connector installation Click to copy this key and save it in a safe place.

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

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



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

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

.
5. 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.

1. 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`
2. 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:
kubectl cluster-info
Cluster setup is completed. Will start the deployment shortly...
Importing the required images...
[36mINFO[0m[0000] Importing image(s) into cluster 'cc'
[36mINFO[0m[0000] Importing images from 1 tarball(s)...
[36mINFO[0m[0000] Importing images '[/home/appviewx/CCTEST/deps/tools/mid-server-docker-image/avx-mid-server-base-22.1.0.0.tar]' into node
 'k3d-cc-server-0'...
[36mINFO[0m[0024] Successfully imported image(s)
[36mINFO[0m[0024] Successfully imported 1 image(s) into 1 cluster(s)
Import in progress...
[36mINFO[0m[0000] Importing image(s) into cluster 'cc'
```

```
[36mINFO[0m[0000] Importing images from 1 tarball(s)...
[36mINFO[0m[0000] Importing images '[/home/appviewx/CCTEST/deps/tools/mid-server-docker-image/k3d-tools-5.2.2.tar]' into node 'k3d-cc-server-0'...
[36mINFO[0m[0005] Successfully imported image(s)
[36mINFO[0m[0005] Successfully imported 1 image(s) into 1 cluster(s)
Import in progress...
[36mINFO[0m[0000] Importing image(s) into cluster 'cc'
[36mINFO[0m[0000] Importing images from 1 tarball(s)...
[36mINFO[0m[0000] Importing images '[/home/appviewx/CCTEST/deps/tools/mid-server-docker-image/rancher-mirrored-coredns-coredns-1.8.6.tar]' into
 node 'k3d-cc-server-0'...
[36mINFO[0m[0007] Successfully imported image(s)
[36mINFO[0m[0007] Successfully imported 1 image(s) into 1 cluster(s)
[36mINFO[0m[0000] Importing image(s) into cluster 'cc'
[36mINFO[0m[0000] Importing images from 1 tarball(s)...
[36mINFO[0m[0000] Importing images '[/home/appviewx/CCTEST/deps/tools/mid-server-docker-image/rancher-local-path-provisioner-v0.0.21.tar]' into node
 'k3d-cc-server-0'...
[36mINFO[0m[0004] Successfully imported image(s)
[36mINFO[0m[0004] Successfully imported 1 image(s) into 1 cluster(s)
[36mINFO[0m[0000] Importing image(s) into cluster 'cc'
[36mINFO[0m[0000] Importing images from 1 tarball(s)...
[36mINFO[0m[0000] Importing images '[/home/appviewx/CCTEST/deps/tools/mid-server-docker-image/rancher-mirrored-pause-3.6.tar]' into node
 'k3d-cc-server-0'...
[36mINFO[0m[0003] Successfully imported image(s)
[36mINFO[0m[0003] Successfully imported 1 image(s) into 1 cluster(s)
Deploying the Cloud Connector...
NAME: avx-mid-server-starter
LAST DEPLOYED: Mon May 30 15:51:13 2022
NAMESPACE: cc
STATUS: deployed
REVISION: 1
NOTES:
1. It may take a couple of minutes for the Cloud Connector to be up.
  kubectl get pod --namespace cc
**************************************************************************
*    Congratulations!!! The installation completed successfully.   *
*    Please wait till the Cloud Connector is up and running.       *
**************************************************************************
(1%) Cloud Connector status: Running
```

```
[32m Cloud Connector is up and running. (B[m
```

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

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

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

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

Working with AppViewX SaaS

The AppViewX SaaS deployment leverages the existing capabilities of the om-premise deployments of the product combining them with the benefits of a cloud-based deployment.

To simplify your interaction with the product's features, AppViewX offers exhaustive documentation in the form of the following guides:

- AppViewX SaaS Cloud Connector User Guide
- AppViewX SaaS Onboarding Guide
- CERT+ User Guide
- CERT+ Admin Guide
- Platform User Guide
- ADC+ User Guide
- ADC+ Admin Guide

You can access the complete AppViewX SaaS documentation here.

Chapter 7: Support

- Support

Support

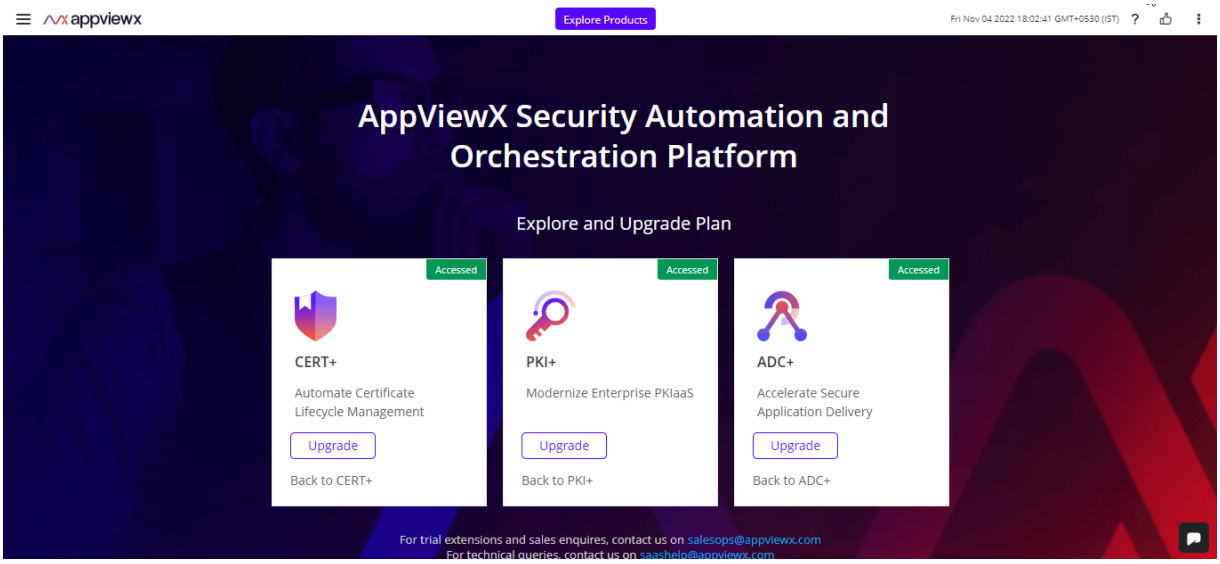
AppViewX's Customer Success team is dedicated to help you with the workings of AppViewX's SaaS-based product line. We have introduced the AppViewX Chatbot, an in-product support interface to help you make your queries specific and, therefore, enable AppViewX's support teams to facilitate expedited solutions. You can use the chatbot to request a demo, a trial extension, a subscription upgrade, or for a query resolution.

- Using the AppViewX Chatbot

Using the AppViewX Chatbot

To access the chatbot:

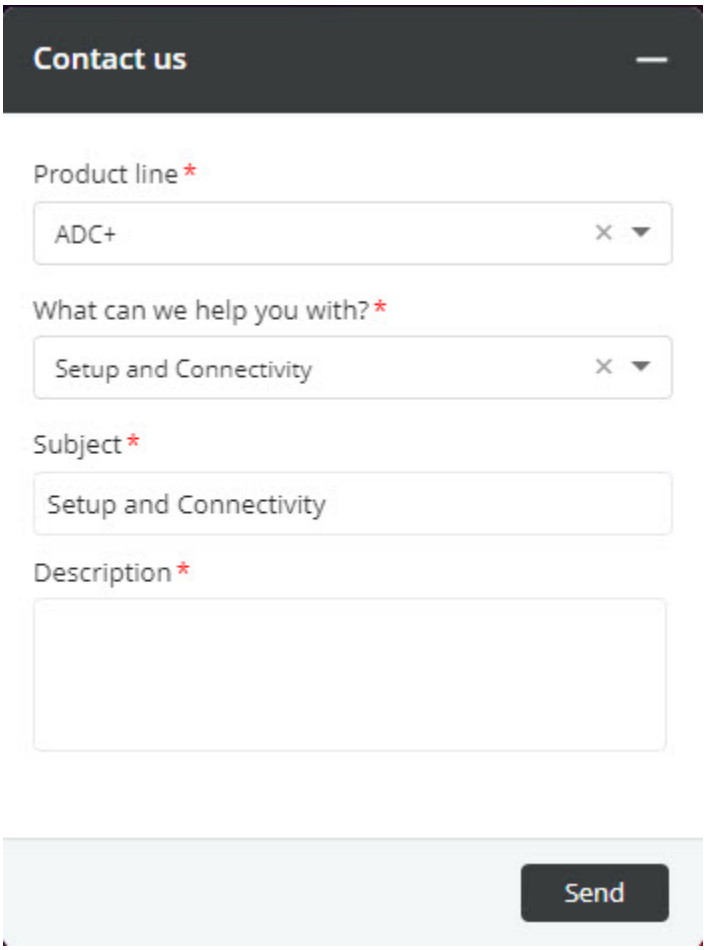
1. Log in to your SaaS account.
   The AppViewX SaaS landing page is displayed.



2. To access the AppViewX chatbot, click  from the bottom-right corner of the screen.

Note: This chatbot icon is available on all product screens, enabling you to send a request at any point during a process.

The **Contact Us** pop-up window is displayed.



3. In accordance to your query, enter the following details:

Table 1.

Field	Description
Product line*	From the dropdown list, select one from the following product line options:

Field	Description
	• CERT+ • ADC+ • PKI+
What can we help you with?*	From the dropdown list, select a category closest to your requirement. The categories in this list include: • Setup and Connectivity • Download/Installation • Artifacts/Solution Guides • System Impaired • Request for upgrade • Request for trial extension • Critical • Others
Subject*	This field is automatically updated with the category you selected in the **What can we help you with?** Field. This field is editable, so you can change the subject line if it helps to better explain your query.
Description*	In this field, enter the details of your requirement. **Note:** This field has a character limit of 255 characters.

4. Click **Send**.

Depending on the category selected, in the **What can we help you with?** field, the relevant AppViewX support team will get in touch with you.

Note: You can also reach out to our teams using the following details:

- salesops@appviewx.com
- saashelp@appviewx.com.
- Phone
  - +1 212 390 1644
  - +1 206 207 7541