

AppViewX Setup Guides

Version: 2024.3.0.0

Copyright AppViewX, Inc.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

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

Trademarks

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

Contact Information

AppViewX, Inc.

222 Broadway, FL 19

New York, NY 10038

Email: info@appviewx.com

Web: www.appviewx.com

Contents

Copyright AppViewX, Inc. ii

Copyright © 2025 AppViewX, Inc. All Rights Reserved. ii

Trademarks. ii

Contact Information. ii

Preface. 9

Revision History. 9

About this Guide. 9

Audience. 9

Third-Party Software Acknowledgements. 9

Text Conventions. 9

Chapter 1. AppViewX On-Prem Setup Guides. 11

Install, Upgrade, and Maintenance Guide. 11

Overview. 11

Working with Prerequisites. 20

Deploying the AppViewX Virtual Appliance. 44

Interactive-Based Installation with Terminal UI. 49

Monitoring and Maintaining AppViewX. 78

External Certificate for Kubernetes. 148

Uninstalling AppViewX. 155

Troubleshooting. 156

Steps to Change Mongodb Password. 158

Disable Kex Algorithm Guide. 160

Migrating CentOS to Ubuntu/RHEL. 162

Upgrading RHEL to v8.10/v9.4. 173

Upgrading Ubuntu 20.04 to Ubuntu 22.04. 176

Upgrading Ubuntu 22.04 to Ubuntu 24.04. 189

OS Patching on Ubuntu for Multi-Node Environment. 199

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Chapter 2. AppViewX SaaS Setup Guides..... 202
SaaS Architecture Guide..... 202
Key Highlights of AppViewX Software as a Service..... 202
AppViewX Architecture..... 204
Multi-Tenancy Architecture..... 208
SaaS Deployment Architecture.....209
AppViewX SaaS Onboarding and Getting Started Guide.....220
Key Highlights of AppViewX Software as a Service..... 220
Introduction to the AppViewX Cloud Connector.....222
Prerequisites for Setting up AppViewX Cloud Connector..... 222
Getting Started with the AppViewX Free Trial..... 223
Signing Up for the Free Trial via the AppViewX Website..... 223
Signing Up for the Free Trial via the AWS Marketplace..... 226
AppViewX Cloud Connector User Guide.....233
AppViewX Software as a Service.....233
Features of the AppViewX Cloud Connector..... 237
System Requirements for Setting up the AppViewX Cloud Connector..... 245
Setting Up the AppViewX Cloud Connector.....253
Managing ADC Devices.....386
Installing the AppViewX Windows Gateway.....389
Troubleshooting the AppViewX Cloud Connector..... 389
Managing the AppViewX Cloud Connector..... 403
Frequently Asked Questions.....415
Appendix A: Network Scan Recommendations.....421
Appendix B: Automated Installation without Internet.....423
Appendix C: CIS Benchmarking for AppViewX Cloud Connector..... 423
Appendix D: Verifying and Renewing Kubernetes Certificates in the AppViewX Cloud Connector.....437
Chapter 3. Managed Kubernetes.....441

Copyright © 2025 AppViewX Inc. All Rights Reserved.

AppViewX Install and Upgrade for AKS 441
AppViewX Architecture 441
Architecture Overview 443
AppViewX Deployment Architecture 444
Managed Kubernetes Architecture 447
AKS Components 448
Prerequisites 448
Install AppViewX in Managed Kubernetes 455
Upgrade AppViewX in Managed Kubernetes 467
Downloading Images from AppViewX Repository 481
Utility Tools and Functions 487
Log Collection 491
Uninstall and Cleanup 493
Troubleshooting 494
AppViewX Install and Upgrade for EKS 494
AppViewX Architecture 495
Architecture Overview 496
AppViewX Deployment Architecture 498
Managed Kubernetes Architecture 500
EKS Components 501
Prerequisites 501
Install AppViewX in Managed Kubernetes 508
Upgrade AppViewX in Managed Kubernetes 520
Downloading Images from AppViewX Repository 533
Utility Tools and Functions 540
Log Collection 544
Uninstall and Cleanup 546
Troubleshooting 547
AppViewX Install and Upgrade for GKE 547

Copyright © 2025 AppViewX Inc. All Rights Reserved.

AppViewX Architecture.....548
Architecture Overview.....549
AppViewX Deployment Architecture.....551
Managed Kubernetes Architecture.....553
GCP Components.....554
Prerequisites.....554
Install AppViewX in Managed Kubernetes.....560
Upgrade AppViewX in Managed Kubernetes.....572
Downloading Images from AppViewX Repository.....586
Utility Tools and Functions.....592
Log Collection.....596
Uninstall and Cleanup.....598
Troubleshooting.....599
ManagedK8s Kubernetes Upgrade: Guidelines for Cluster Admins.....599
Understanding the Managedk8s Kubernets Upgrade Process.....599
Potential Failure Scenarios and their Mitigations.....600
Chapter 4. OpenShift Deployment with ArgoCD.....602
Chapter 5. Overview.....603
Chapter 6. Prerequisites.....604
Chapter 7. Creating Image Pull Secret.....612
Chapter 8. Installing avx-infra.....613
Chapter 9. Installing avx-app.....617
Chapter 10. Deleting the avx-app.....620
Chapter 11. Upgrading the avx-app.....621
Chapter 12. Migrating Data from On-Premise to OCP.....622
Chapter 13. MSP Portal User Guide.....624
Benefits of AppViewX for MSSPs.....624
What is the MSP Portal?.....625
Key Benefits of the MSP Portal for Partners.....626

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Accessing the MSP Portal.... 626
User Preferences.... 626
Viewing the MSP Portal Dashboards.... 627
Usage and Adoption.... 628
Tenants Insights.... 628
Managing White Labels.... 629
Managing License.... 629
Managing Plans in the MSP Portal.... 630
Creating Custom Plans.... 632
Viewing Clusters in the MSP Portal.... 635
Managing Tenants in the MSP Portal.... 637
Creating a New Tenant.... 640
Extending Free Trial.... 643
Activating a Tenant License.... 645
Deleting Tenants.... 646
Upgrading License Details for a Tenant.... 650
Repropagating Tenant Details.... 651
Enable mTLS.... 652
Disable mTLS.... 653
Viewing MSP License Usage Details.... 654
Accessing the Tenant Environment.... 654
Chapter 14. AppViewX Windows Gateway Setup.... 659
AppViewX Windows Gateway.... 659
Deployment Modes.... 661
Trusted Domains (2 way Trust).... 662
Independent Domains.... 665
Communication Modes.... 666
Powershell.... 666
WMI.... 672

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Native..... 673
Setting up the AppViewX Windows Gateway..... 674
Step 1: Prerequisites for Installing the AppViewX Windows Gateway..... 675
Step 2: Downloading the AppViewX Windows Gateway Installer..... 676
Step 3: Installing the AppviewX Windows Gateway..... 677
Step 4: Verifying the AppviewX Windows Gateway Installation..... 685
Step 5: Managing a Target Server..... 689
Step 6: Disabling Current Operating System Information..... 690
Installation using Non-Admin Service Account..... 690
Configuration Parameters..... 691
Allow Logon Locally Permission..... 693
WinRM HTTPS Listeners..... 694
Uninstalling the AppViewX Windows Gateway..... 695
Updating AppViewX Windows Gateway..... 696
Appendix A..... 696
General Prerequisites..... 696
Firewall Requirements..... 698
Minimum Permissions Required for Communication..... 699
Troubleshooting..... 723
FAQ: Installation and Configurations..... 736
Quick References..... 741
Chapter 15. Support..... 743
Using the AppViewX Chatbot..... 743
AVXpert AI Assistant..... 745
Chapter 16. Glossary..... 752

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Preface

Revision History

Revision	Description	Date
1.0	Initial release of the document for AppViewX v2024.3.0.0	September 2025

About this Guide

The AppViewX Setup Guide is a consolidated document with detailed instructions for setting up AppViewX for the On-prem,SaaS, and Managed Kubernetes based deployments.

Audience

The guide is for the users who want to install the AppViewX stack on managed AKS, specifically

- Platform engineers
- Implementation specialist
- Kubernetes administrators

Third-Party Software Acknowledgements

This section serves as a placeholder to document the third-party components referenced in this guide, along with their associated trademark information.

For example,

- This document includes software details developed by VMware, Inc. (www.vmware.com).

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.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Convention	Description
`codeblock`	Indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Chapter 1: AppViewX On-Prem Setup Guides

• Install, Upgrade, and Maintenance Guide

Install, Upgrade, and Maintenance Guide

This document covers the installation, maintenance, and upgrade activities for AppViewX.

• Overview
• Working with Prerequisites
• Deploying the AppViewX Virtual Appliance
• Interactive-Based Installation with Terminal UI
• Monitoring and Maintaining AppViewX
• External Certificate for Kubernetes
• Uninstalling AppViewX
• Troubleshooting
• Steps to Change Mongodb Password
• Disable Kex Algorithm Guide
• Migrating CentOS to Ubuntu/RHEL
• Upgrading RHEL to v8.10/v9.4
• Upgrading Ubuntu 20.04 to Ubuntu 22.04
• Upgrading Ubuntu 22.04 to Ubuntu 24.04
• OS Patching on Ubuntu for Multi-Node Environment

Overview

AppViewX's offering is a modular, low-code software application that enables the automation and orchestration of network infrastructure using an intuitive, context-aware, visual workflow. Leveraging a vast library of pre-built tasks and workflows, AppViewX enables the operations teams to quickly and easily translate business requirements into automation workflows that improve agility, enforce compliance, eliminate errors, and reduce cost. AppViewX is closed-loop and state-aware, capable of verifying that intent has been achieved and providing actionable insights and automated remediation.

AppViewX is a web based application that offers:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- CERT+, 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+ User Guide.

- ADC+, 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+ 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.

- SSH+, which lets you:
  - Discover and display SSH certificates alongside SSH keys, offering a more comprehensive overview of your security credentials.
  - Download keys for key-based access control, ensuring streamlined access management.
  - Specify access duration in either hours or days when requesting access to an infrastructure group, providing enhanced access management control.
  - Use a dynamic access flow that adapts to either key or certificate-based access, depending on the user's selected 'Access Mode' during host addition.
  - Rotate host certificates effortlessly, directly from the host inventory, promoting secure host certificate management.
  - Revoke SSH certificates directly, thus enhancing security control.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- Choose between 'Key' and 'Certificate' access modes during host addition, with the 'Certificate' option being pre-selected by default.
- Rotate and delete keys from hosts with multiple keys through the user and host key age report.

For details, refer the SSH+ User Guide.

- SIGN+, which lets you:
  - Simplify Code Signing Certificate enrollment and Certificate Lifecycle Management (CLM) operations.
  - Customize signing policies according to your requirements
  - Integrate with AppViewX's customized Cryptographic Service Provider (CSP) and PKCS#11 for enhanced security.
  - Manage your code signing inventory with a full suite of tools and features.
  - Sign your code effortlessly using a variety of tools including SignTool, JSign, JarSigner, APKSigner, Mage, and Nuget.
  - Ensure compatibility with third-party Timestamp Authorities (TSA) for a wider range of options.

  For details, refer the SIGN+ User Guide.
- KUBE+, which lets you:
  - Simplify Certificate Lifecycle Management for Kubernetes workloads.
  - Get real-time visibility, central audit, and governance over K8’s Certs.
  - Achieve end-to-end automated certificate enrollment process.
  - Have secure and compliant PKI across K8s workloads (secrets, pods, and service mesh).

  For details, refer the KUBE+ User Guide.

AppViewX is built on the microservice architecture. A microservice is a program that runs on a server or a virtual computing instance. The main task of this program is to respond to network requests.

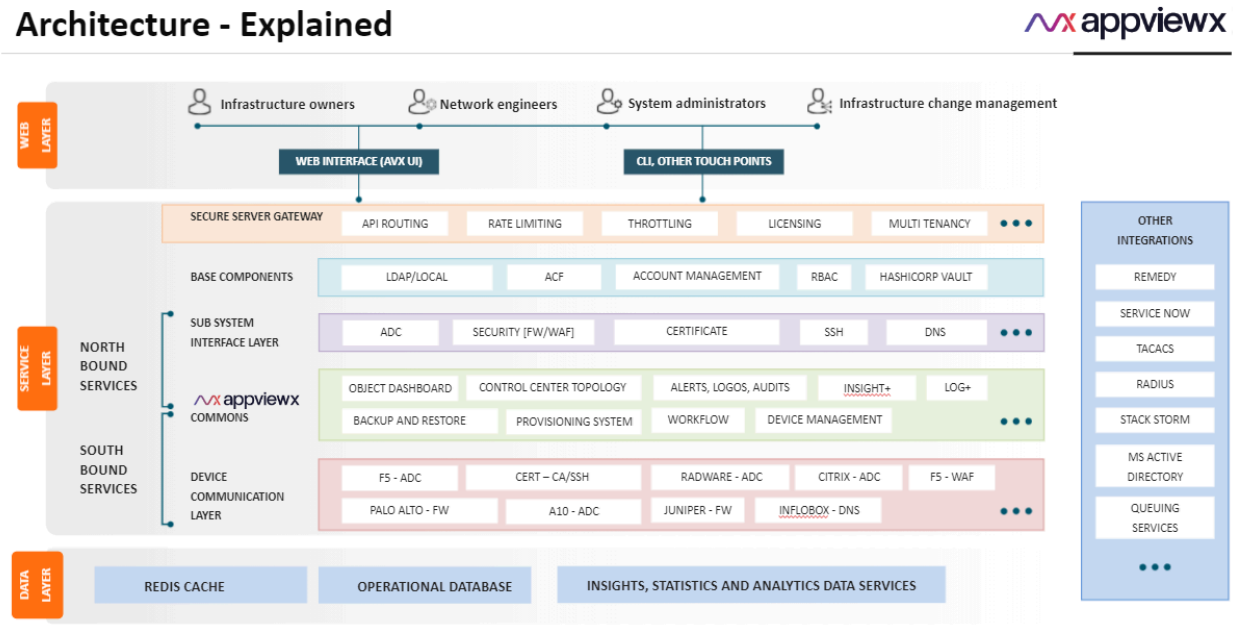
- AppViewX Architecture
- Benefits of AppViewX
- Supported Deployment Methods and Types

AppViewX Architecture

AppViewX is built on Kubernetes, an open-source platform for deploying and managing containers. It provides a container runtime, container orchestration, self-healing mechanisms, service discovery and load balancing. It’s used for the deployment, scaling, management, and composition of application containers across clusters of hosts.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

AppViewX is designed based on microservice architecture making it easier to move to containerized workloads and the containers being orchestrated using Kubernetes. The following diagram depicts the deployment architecture:



In the diagram:

- **Presentation/ Web Layer** - houses the AppViewX user interface related files and interacts with the service layer
- **Service Layer** - contains the Northbound & Southbound services that can be further classified into:
  - **Business Layer**:
    - Houses AppViewX specific business logic
    - Interacts with the Data layer for persisting the input data
  - **Device Communication Layer**:
    - Low code
    - Stateless layer
    - Routes communication to the respective vendor through APIs or SSH
    - Houses vendor specific business logic
- **Data Layer**:
  - Houses data persistence and retrieval logic
  - Redis caching is available

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Benefits of AppViewX

In order to optimally utilize the resources, AppViewX has adopted Kubernetes to achieve higher security by adopting a zero trust network model. The features of AppViewX coupled with Kubernetes are given below.

- **Auto scaling**

  AppViewX services can have a custom throttling capability based on pre-configured memory configuration per API. This will enable AppViewX services to utilize (scale up) resources optimally as the demand surges and scale down when not in use. This will help to horizontally scale the vendor components on demand and optimize the resource usage.

- **Resiliency**

  There is no guarantee that the services will run without any interruption and they are bound to failure. Kubernetes keeps deployments healthy by restarting containers that have failed, killing and replacing unresponsive containers based on health checks. This helps to mitigate the common pain point of the application upkeep process.

- **Security**

  AppViewX architecture is designed around the concept of zero trust network model to enforce tighter security within the Kubernetes cluster. This means no one is trusted by default and required verification to gain access to the services.

Supported Deployment Methods and Types

This section explains the types and methods in which you can deploy AppViewX.

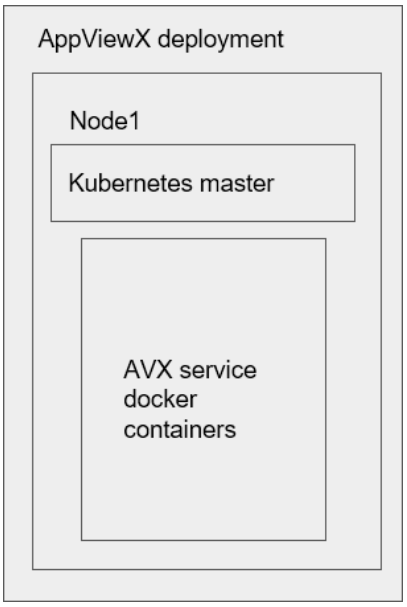
> ⚠ **Warning:** Hybrid cloud management deployment is not supported in AppViewX.

First, AppViewX can be deployed in the following modes:

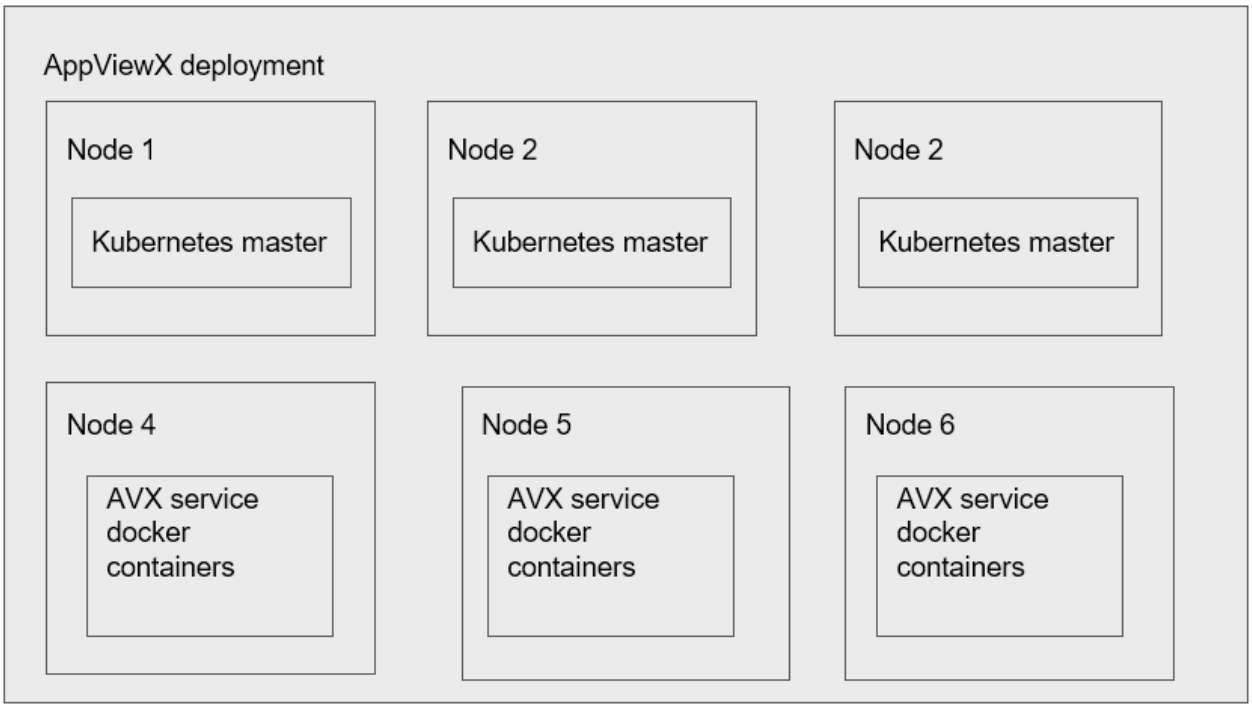
1. **Single Node** - is used to host all the services on a single setup.
   - Single-node setups may have lower performance because of a lack of resources.
   - Node resiliency and HA are not supported in single-node deployment.
2. **Multi node** - is used to host the services across multiple nodes to ensure high availability.

The following diagrams depict AppViewX deployment on a single node and a multi node mode:

Copyright © 2025 AppViewX Inc. All Rights Reserved.



Single Node Deployment



Multi Node Deployment

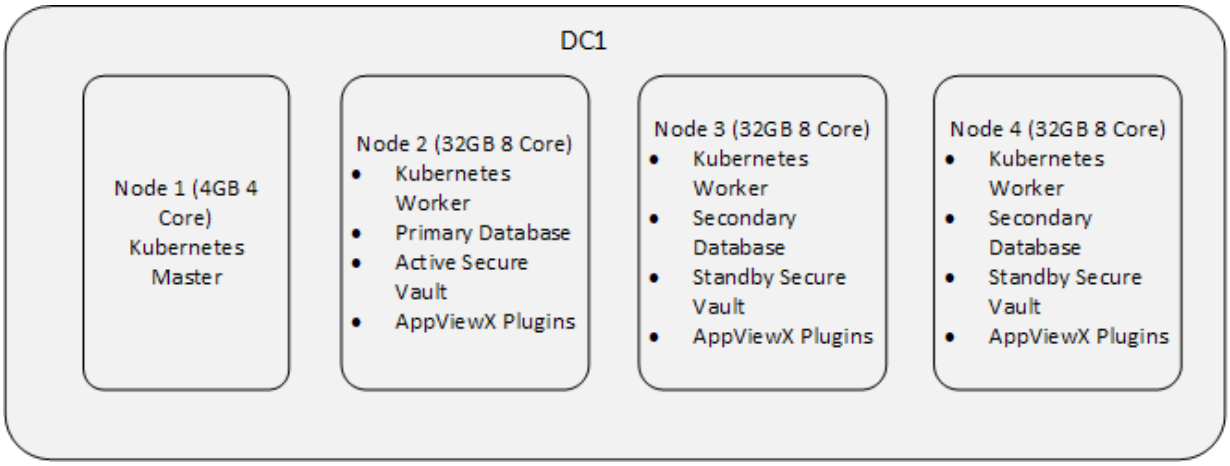
Once the deployment mode is finalized, AppViewx can be installed using any one of the following methods:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- **OVA Installation** - stands for Open Virtual Appliance that contains a compressed and installable version of a virtual machine. When you use an OVA-based installer, the installation-related artifacts are pre-bundled as part of the OVA.
- **Native Installation** - uses the standard command line interface to execute installation commands.

AppViewX supports the following deployment types/scenarios:

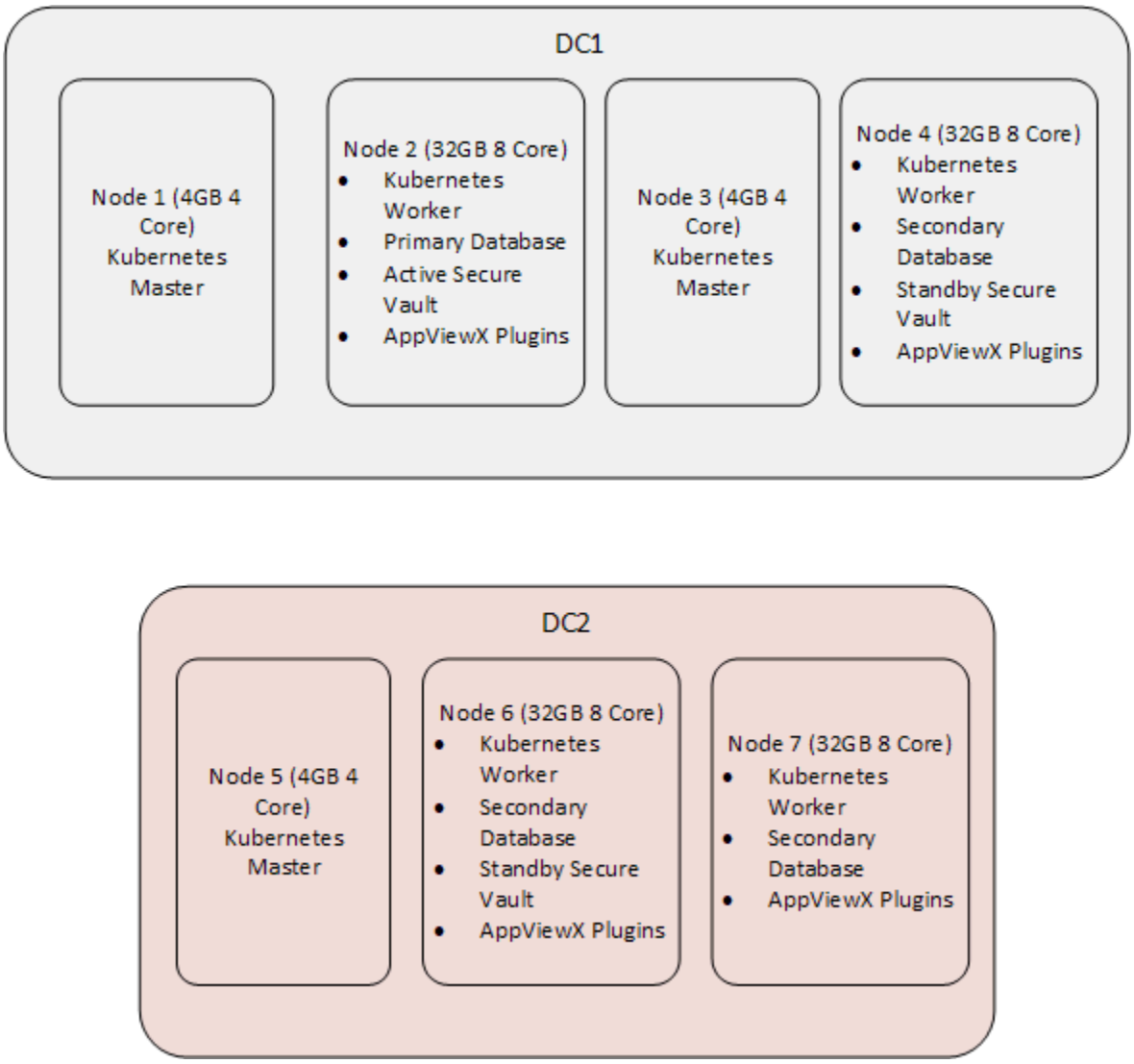
- One Data Center and Four Nodes



This deployment is recommended for customers who need only HA at the application level. This deployment does not support HA; neither for the Kube nor for the DC. This deployment is best suited for less than 50 ADC devices having a total of 100,000 objects and 10,000 certificates.

- Two Data Centers and Seven Nodes

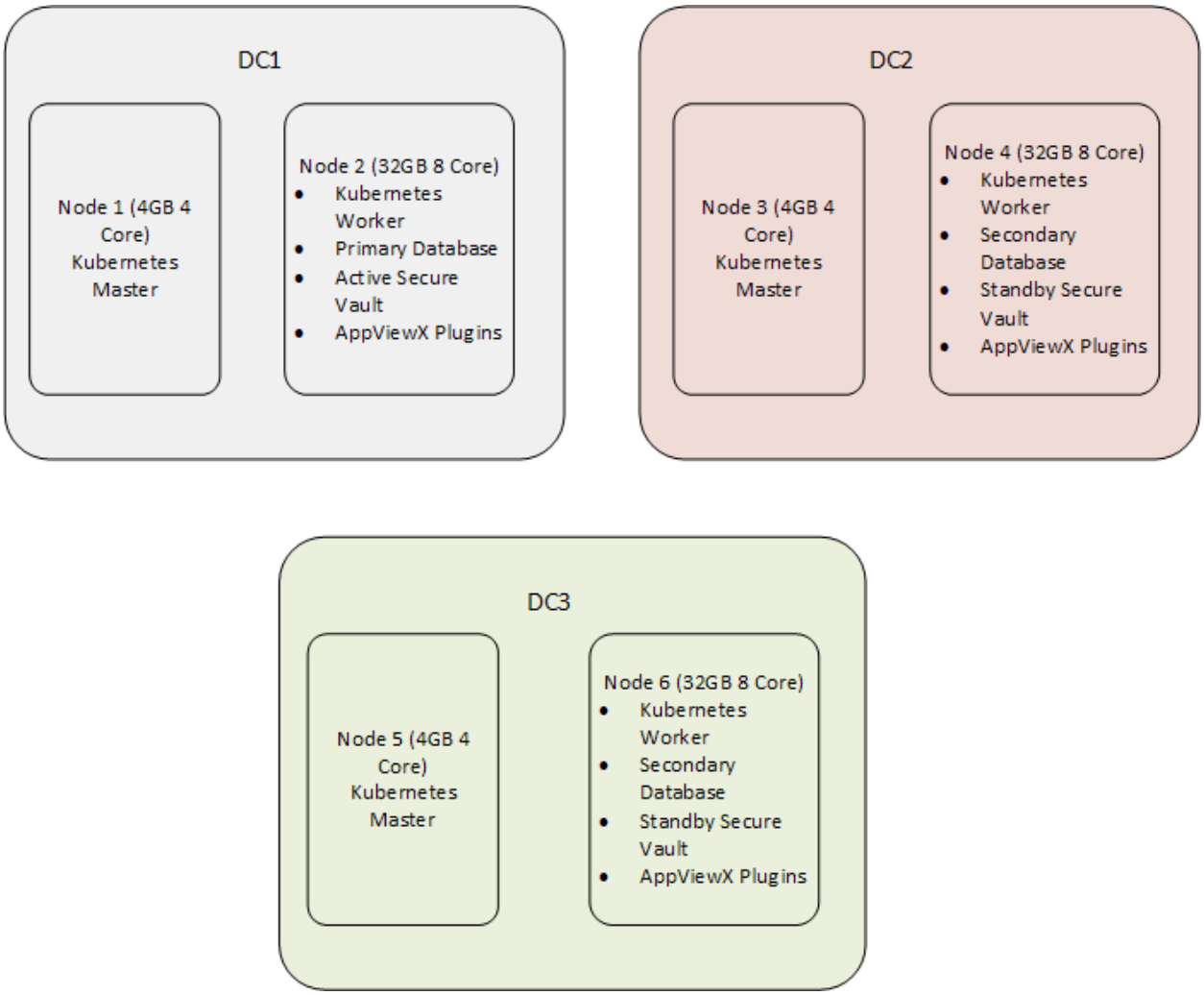
Copyright © 2025 AppViewX Inc. All Rights Reserved.



This deployment is recommended for customers who require HA at the Application, Kube, and DC level. This deployment supports HA for the Kube, Application as well as the DC. This deployment is best suited for 50 to 100 ADC devices having a total of 300,000 objects and 10,000 certificates.

• Three Data Centers and Six Nodes

Copyright © 2025 AppViewX Inc. All Rights Reserved.



This deployment is recommended for customers who require HA at the Application, Kube, and DC level. This deployment supports HA for the Kube, Application as well as the DC. This deployment is best suited for 50 to 100 ADC devices having a total of 300,000 objects and 10,000 certificates.

The table below summarizes the different deployments supported by AppViewX.

Table - Deployments supported by AppViewX

Model	Load	HA		
		Kube	DC	Application
1 DC 4N	Less than 50 ADC devices having a total of 100,000 objects and 10,000 certificates.	No	No	Yes
2 DC 7N	50 to 100 ADC devices having a total of 300,000 objects and 10,000 certificates.	Yes	Yes	Yes

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Table - Deployments supported by AppViewX (continued)

Model	Load	HA		
		Kube	DC	Application
3 DC 6N	50 to 100 ADC devices having a total of 300,000 objects and 10,000 certificates.	Yes	Yes	Yes

Note: Apart from the deployments mentioned here, AppViewX can customize the deployment based on the needs and requirements.

Working with Prerequisites

This section covers all the prerequisites required to install AppViewX on the system.

- Understanding Hardware and Software Requirements
- Configuring Elevated Access
- Configuring Firewall Ports
- Configuring YUM
- Configuring Calico before Deployment
- Configuring SELinux
- Configuring NTP
- Configuring Ulimit
- Increasing vm.max_map_count
- Enabling IP Forwarding
- Enabling Bridging
- Enabling the IP in IP Protocol
- Downloading AppViewX Packages
- Running the Prerequisite Tool

Understanding Hardware and Software Requirements

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Hardware Requirements

Before proceeding with the installation, ensure that you have, at minimum, the following hardware with the specifications given below:

- **Single Node Requirements**

Node	Bare Minimum		
	CPU	RAM	Hard Disk Space
Single Node	8	32 GB	500 GB

- **Multi-Node Requirements**

Node	Bare Minimum		
	CPU	RAM	Hard Disk Space
Multi-node (master node)	4	4 GB	100 GB
Node	**Bare Minimum**		
	CPU	**RAM**	**Hard Disk Space**
Note: One node for a single master installation and a minimum of three nodes for multi-master installation.			
Node	**Bare Minimum**		
	CPU	**RAM**	**Hard Disk Space**
Multi-node (worker node)	8	32 GB	500 GB

Note: For more information on the nodes, refer to the Supported Deployment Methods and Types section.

For deploying the OVA, ensure that you have all the prerequisites as mentioned below.

- Platform Bare Minimum Requirements

Supported Virtualization	Versions	VCPU	RAM	HDD
VM Server, VMware ESXi	5.5 or later	8v	32 GB	1 TB

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Software Requirements

The following operating systems support both single node and multi-node installations of AppViewX:

Operating System	Supported Versions
RHEL	• 9.2 • 9.3 • 9.4 • 9.5 • 9.6
Ubuntu	• 20.04 • 22.04 • 24.04
Rocky Linux	• 9.5
Oracle Linux	• 9.2 • 9.3 • 9.4 • 9.5 • 9.6

Ensure the following packages are installed before proceeding with the AppViewX installation:

For RHEL:

- nmap
- curl
- bindutils
- net-tools
- rsync
- nmapncat
- fontconfig

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- zip
- unzip
- sysstat
- tcpdump

For UBUNTU:

- nmap
- curl
- ebtables
- sysstat
- zip
- unzip
- rsync
- dnsutils
- fontconfig
- net-tools
- tcpdump

Prerequisite OS Packages and their Usage

This section outlines the essential packages required for the Linux server, which are necessary for both the installation and maintenance of AppViewX application. Each listed package is accompanied by a detailed description of its functionality and usage, ensuring a comprehensive understanding of its role in the AppViewX environment.

S. No.	Package Name	Required by Application Functionality	Usage
1	Nmap (Ubuntu)	No	This package is essential during installation and upgrades to check if required ports are open, helping verify connectivity for services before deployment. Additionally, Nmap aids in troubleshooting network-related issues within the cluster. Its usage is diagnostic, with no impact on the functionality of the running application.
2	Nmap-Ncat (RHEL)	No	This package is essential during installation and upgrades to check if required ports are open, helping

Copyright © 2025 AppViewX Inc. All Rights Reserved.

S. No.	Package Name	Required by Application Functionality	Usage
			verify connectivity for services before deployment. Additionally, Nmap aids in troubleshooting network-related issues within the cluster. Its usage is diagnostic, with no impact on the functionality of the running application.
3	Curl	No	This package is not required for application functionality but it is used during the installation and AppViewX utilities. AppViewX utilities are a set of scripts which are used for maintenance purposes.
4	IPtables-ebtables (RHEL8)	Yes	This package is crucial for Kubernetes installation and upgrades, used by the **kube-proxy** component to set up iptable rules that manage pod and service communication. As Kubernetes is the base layer for the microservice-based application, this package is essential for setup and running the application.
5	IPtables-nft (RHEL 9)	Yes	This package is crucial for Kubernetes installation and upgrades, used by the **kube-proxy** component to set up iptable-nft rules that manage pod and service communication. As Kubernetes is the base layer for the microservice-based application, this package is essential for setup and runtime reliability.
6	Ebtables (Ubuntu)	Yes	This package is crucial for Kubernetes installation and upgrades, used by the **kube-proxy** component to set up iptable rules that manage pod and service communication. As Kubernetes is the base layer for the microservice-based application, this package is essential for setup and running the application.
7	Zip	No	The zip package compresses files and directories into a single archive, optimizing storage and facilitating file sharing. This package is not required for running the application but it's useful in compressing the logs

Copyright © 2025 AppViewX Inc. All Rights Reserved.

S. No.	Package Name	Required by Application Functionality	Usage
			file collected for troubleshooting. It is used during the installation of AppViewX.
8	Unzip	No	The unzip package extracts files from a .zip archive, allowing you to access the original files and directories. This package is not required for running the application but it's useful in compressing the logs file collected for troubleshooting. It is used during the installation of AppViewX.
9	Rsync	No	The rsync package synchronizes files and directories between two servers. This package is not required for running the application but it is used during the application patching activities.
10	OpenSSL	No	OpenSSL manages secure communications by generating, managing, and verifying cryptographic keys and certificates. Although it’s not required for running the main application, it is used in AppViewX utilities. It is used during the installation of AppViewX.
11	Systat	No	Sysstat provides system performance monitoring tools like **sar**, **iostat**, and **mpstat** to report on CPU, memory, disk, and network usage. It’s mainly for performance analysis and is not directly required for application functionality.
12	Bind-utils (RHEL)	No	The bind-utils package includes DNS troubleshooting tools like dig, nslookup, and host, which allow you to query DNS records, test DNS configurations, and perform domain lookups to resolve or debug DNS issues. It is used during the installation of applications but it's not mandatory for running the application.
13	Dnsutils (Ubuntu)	No	The dnsutils package provides tools like nslookup, dig, and host to perform DNS queries, troubleshoot DNS issues, and retrieve information about domain names and IP addresses. It is used during the

Copyright © 2025 AppViewX Inc. All Rights Reserved.

S. No.	Package Name	Required by Application Functionality	Usage
			installation of applications but it's not mandatory for running the application.

Installation Prerequisites

These prerequisites are applicable to both single-node and multi-node setups.

Operating System Consistency

All nodes must run the same Operating System (OS) and the same version. (Refer to the section Software Requirements for the list of supported OS).

Default User Shell

The default shell for the installation user must be set to **bash**.

SSH and SSH Algorithm

SSH access (either single-node or multi-node) must be functional with **PubKeyAuthentication** enabled. Additionally, the **ssh-rsa** algorithm must be supported by the SSH daemon during installation (*This can be disabled post-installation*).

Filesystem Type at Installation Path

The target installation path must reside on an **xfs** or **ext4** filesystem.

Permissions on /tmp Directory

The installation user must have read, write, and execute permissions on **/tmp**.

Backup Path Access

The installation user must have read and write access to the backup path directory.

Network Subnet Conflict Check

The **SERVICE_SUBNET** and **POD_SUBNET** specified in appviewx.conf must not conflict with any node IPs or existing network subnets in the environment.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Summary

All checks listed above must be successfully validated across **all target nodes** before proceeding with the installation. The installer (./install.sh) will perform these checks:

- Any **critical failure** among the above items will cause the installer to **exit immediately**.
- Remaining validations will be **collected and presented in a tabular format**, allowing the user to review and resolve issues before continuing with the application installation.

Configuring Elevated Access

AppViewX is installed on top of a Kubernetes engine and to install the underlying Kubernetes engine and other dependent packages like docker, we would require the user to have sudo access and executable permission for the tmp folder. Refer to the Understanding Commands Executed during Installation section to get the details on the commands that the sudo user needs access to.

Note: If you are using an OVA-based installer, a user named "appviewx" is already available with Super user privileges.

Configuring Firewall Ports

The following ports must be opened between the nodes to install AppViewX. Users can configure it in a firewall device, firewalld, or using iptables.

Table - Firewall Ports

Sr No	Source		Destination		Protocol Used	TCP/ UDP	Type of Information Communicated
	IP	Port	IP	Port			
1	All Nodes	Any	All Nodes*	22	SSH	TCP	Required for AppViewX installation and prerequisite checks.
2	All Nodes	Any	All Nodes*	179	BGP	TCP	To establish a common routing table for the overlay network.
3	All Nodes	Any	All Nodes*	6443	HTTPS	TCP	Kubernetes API server for communication between Kubernetes master and worker nodes.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Table - Firewall Ports (continued)

Sr No	Source		Destination		Protocol Used	TCP/ UDP	Type of Information Communicated
	IP	Port	IP	Port			
4	All Nodes	Any	All Nodes*	10250	HTTPS	TCP	Used by Kubelet Agent which exposes Rest endpoints for the Kubernetes API Server.
5	Load Balancer (for ex, F5, GCP, etc.)	Any	ISTIO Ingress Proxy IP (Kube Worker)	31443	HTTPS	TCP	To access the AppViewX web user interface. **Note:** This port must be open in a **Single-Node** setup.
6	Load Balancer (for ex, F5, GCP, etc.)	Any	Kube Master IP	6443		TCP	To allow communication between the F5 load balancer and the pool members (master nodes).
7	All Nodes	Any	F5 VIP	6443		TCP	To allow all the nodes to communicate with the Kube Master for Kubernetes Control plane traffic.
8	AppViewX Admin network #	Any	ISTIO Ingress Proxy IP (Kube Worker)	30190	HTTPS	TCP	To access the AppViewX management console. **Note:** This port must be open in a **Single-Node** setup.
9	All Nodes	-	All Nodes*	-	IP-IP IP Protocol 4	NA	Overlay network established with IP-IP tunnels. Information over this tunnel is encrypted using mTLS. **Note:** This protocol must be accessible in a **Single-Node** setup.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Table - Firewall Ports (continued)

Sr No	Source		Destination		Protocol Used	TCP/ UDP	Type of Information Communicated
	IP	Port	IP	Port			
10	Master	Any	Kube Master	2379	HTTPS	TCP	Required for etcd client communication from worker nodes to master node.
11	Master	Any	Kube Master	2380	HTTPS	TCP	Required for etcd peer communication in a multi-master setup.
12	All Nodes	Any	All Nodes*	9100	HTTP	TCP	Required for monitoring the node metrics.
13	All Nodes	Any	All Nodes*	4789	VXLAN	UDP	To establish a common routing table for the overlay network.
14	All Nodes	Any	All Nodes*	30022 (SCEP) 30021 (EST)	HTTPS	TCP	To access the Auto-Enrollment Protocols (SCEP & EST)

* (asterisk) indicates all the nodes present in the cluster i.e. master nodes, secondary master nodes, and worker nodes.

- indicates the network/machines/nodes of users who want to manage AppViewX Infra using the management console (actions include create, delete pods, and/or services.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Note:

- The system will require 1 IP per node.
- The externally exposed services will all use the nodes IP address to communicate within the network.
- Port 22 is used for administration of the node for example to log into the linux CLI. Need SSH access the nodes to other nodes.
- We would need an external Load Balancer to distribute user/API traffic to all Kube master nodes. We can open firewall ports depending on the network setup.
- Ensure that the external endpoints that you want to access from the AppViewX worker nodes are accessible., e.g. Microsoft CA. Ensure that the corresponding ports and URLs are opened for communication.

Configuring Firewall Ports for External Integrations

Table - Firewall Ports for External Integrations

S. No	Source		Destination		Protocol Used	TCP/ UDP	Type of Information Communicated
	IP	Port	IP	Port			
1	AppViewX Worker Nodes	Any	ADC		SSH		
2	AppViewX Worker Nodes	Any	ADC		HTTPS		To execute REST APIs
3	AppViewX Worker Nodes	Any	MSCA Agent		HTTPS		AppViewX to MSCA agent communication
4	AppViewX Worker Nodes	Any	CA		HTTPS		To execute REST APIs

Configuring YUM

This section guides users to configure AppViewX nodes to the YUM repository hosted by AppViewX. Yum will sync only AppViewX repositories to get the OS package updates. This task is required to update the OS security patching on AppViewX supplied OVAs.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Note: For information regarding the best practices on rebooting the operating system after security patching, refer to the Graceful Reboot Procedure.

Warning: This will remove all the other repositories configured in the system.

Before you configure yum, ensure that:

1. AppViewX nodes have access to the following URL https://repos.appviewx.com
2. The user has root/sudo access to configure yum.

To configure YUM:

1. Download the **appviewx.repo** file from the release portal.
2. Login as a root user.
3. To take a backup of existing yum repositories, execute the following command:

   ```
   mv /etc/yum.repos.d /etc/yum.repos.d_backup
   ```

   This is to ensure that we have a backup of the existing yum repository configurations.
4. To create a yum repository, execute the following command:

   ```
   mkdir -p /etc/yum.repos.d
   ```
5. To copy the appviewx.repo to yum.repos.d, execute the following command:

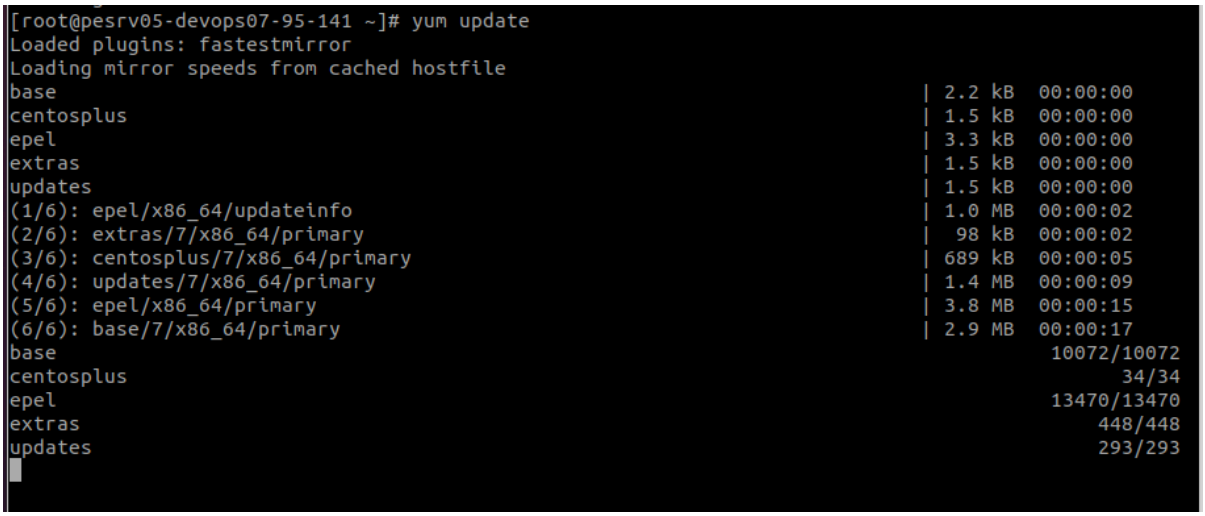
   ```
   cp appviewx.repo /etc/yum.repos.d/
   ```
6. To clean the yum repository, execute the following command:

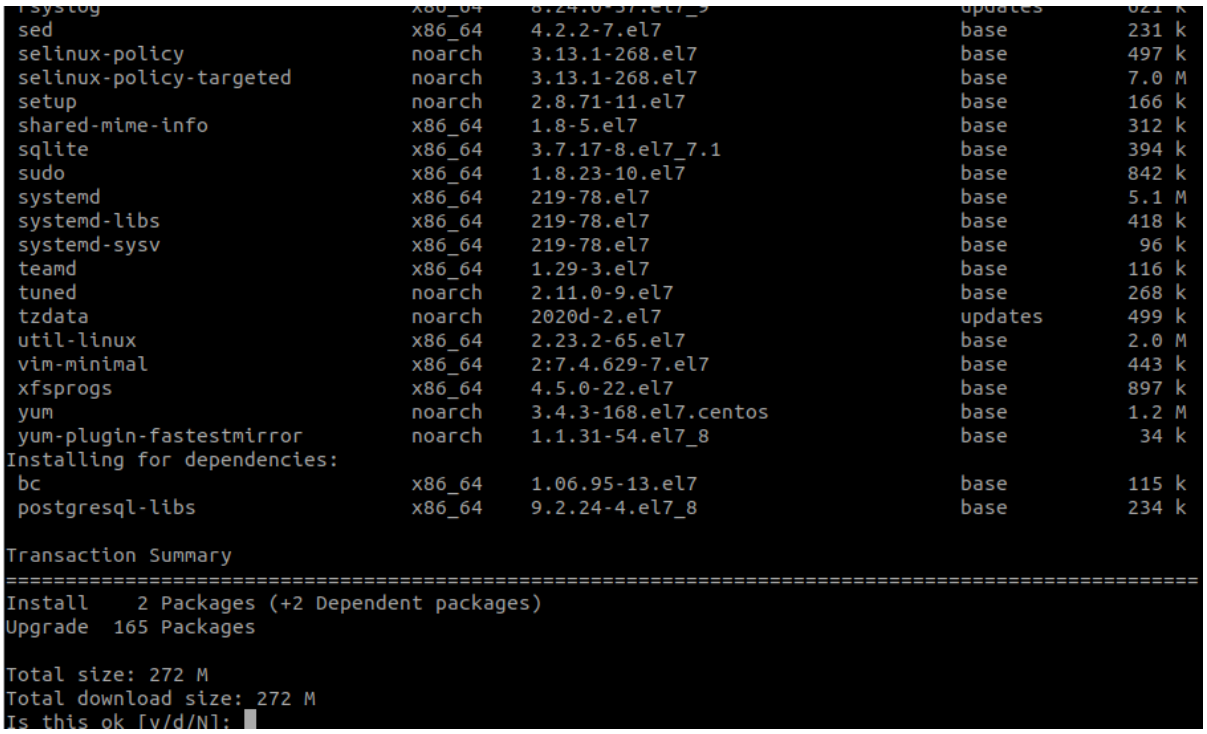
   ```
   yum clean all
   ```
7. To get the latest updates from repos.appviewx.com, execute the following command:

   ```
   yum update
   ```

   The command will connect to the AppViewX repository and update the packages. Reference images are given below:

Copyright © 2025 AppViewX Inc. All Rights Reserved.





Configuring Calico before Deployment

This section provides instructions on configuring calico before deploying AppViewX on Azure.

> **Warning:** Follow these instructions ONLY if you are deploying AppViewX on Azure.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

1. Navigate to the /home/appviewx/appviewx_kubernetes/configs/kube directory.
2. Open the **calico.yaml** file in edit mode.
3. Change the value of the **CALICO_IPV4POOL_VXLAN** parameter from **CrossSubnet** to **Always**.
4. Change the value of the **CALICO_IPV4POOL_IPIP** parameter from **Always** to **Never**.
5. Save the changes to the **calico.yaml** file.
6. Close the editor.

Configuring SELinux

To configure SELinux

1. Open the file /etc/selinux/config file
2. Configure the parameters, **SELINUX=permissive** or **SELINUX=disabled**
3. Reboot the node by the command

```
sudo reboot
```

4. Verify that the command output below should be permissive

```
getenforce
```

Configuring NTP

To configure NTP

1. Install the NTP service by the command

```
sudo yum install ntp
```

2. Update the NTP server details in **/etc/ntp.conf** or **/etc/chrony.conf**
3. Restart the NTPD/chronyd service by the command

```
sudo systemctl start ntpd
```

4. Verify the NTP status using command

```
ntpstat
```

Configuring Ulimit

To set or verify the ulimit values on Linux:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

1. Edit the **/etc/security/limits.conf** file and specify the following values:
   - **<USERNAME> soft nofile 65536**
   - **<USERNAME> hard nofile 65536**
2. Exit and login again to verify the changes.
3. Verify the Ulimit using the command

```
ulimit -n
```

Increasing vm.max_map_count

To increase the vm.max_map_count

1. Execute the command

```
sudo sysctl -w vm.max_map_count=262144
```

2. Verify the value using the command

```
cat /proc/sys/vm/max_map_count
```

Enabling IP Forwarding

1. In the **/etc/sysctl.conf** file, add the parameter **net.ipv4.ip_forward=1**
2. Execute the command

```
sudo sysctl -p
```

3. Verify the IP_Forwarding using the command

```
sysctl net.ipv4.ip_forward
```

Enabling Bridging

To enable bridging

1. In **/etc/sysctl.conf** file, add the following parameters:
   - **net.bridge.bridge-nf-call-ip6tables = 1**
   - **net.bridge.bridge-nf-call-iptables = 1**
2. Execute the following commands:

```
sudo modprobe br_netfilter
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
sudo sysctl -p
```

3. Verify the bridging using the command

```
sysctl net.bridge.bridge-nf-call-iptables
```

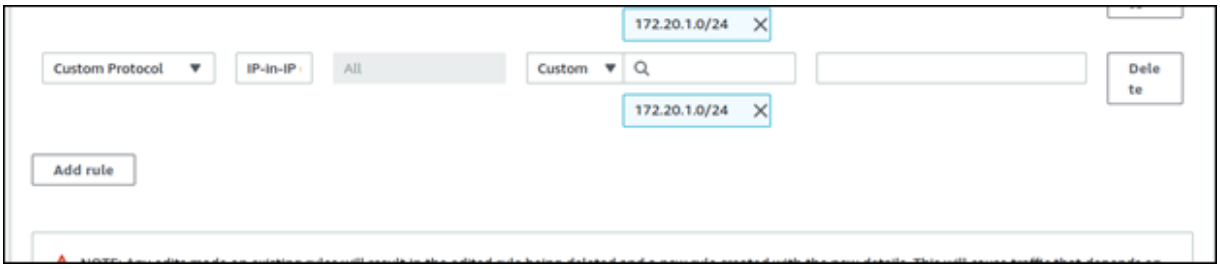
```
sysctl net.bridge.bridge-nf-call-ip6tables
```

Enabling the IP in IP Protocol

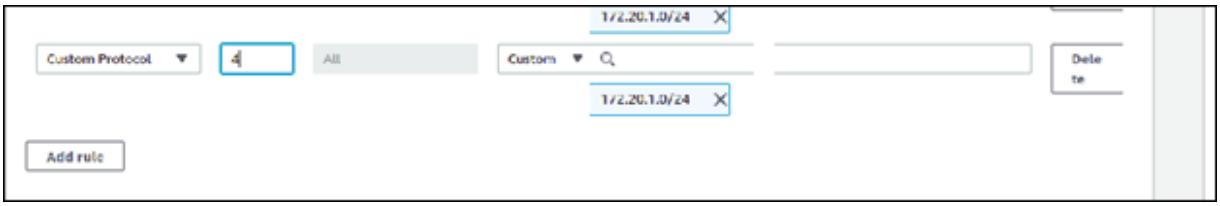
> **Warning:** Follow these steps ONLY if you want to deploy AppViewX on AWS.

You must enable the IP in IP protocol between the nodes in the AWS security group before deploying AppViewX.

1. Log in to the AWS console.
2. Navigate to the security group that needs to be modified.
3. Click **Edit inbound rules**.
4. Click **Add rule**.
5. From the **Add rule** list, select **Custom Protocol**.



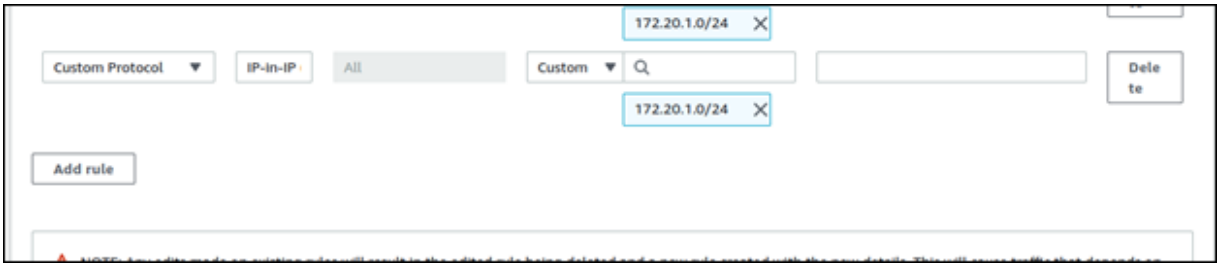
6. Enter the protocol value as **4**.



7. Enter the subnet across which IP in IP needs to be enabled.
8. Click **Save rule**.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

The protocol automatically changes to IP in IP.



Downloading AppViewX Packages

To install AppViewX, download the following packages from the AppViewX Release Portal.

Note: To get the release portal credentials, contact help@appviewx.com.

File Name	Mandatory	Description	Purpose
appviewx_kubernetes_2024.3.0.0.tar.gz	Yes	AppViewX core installer	Core installer that has the AppViewX package from which the installation is triggered.
appviewx_kubernetes_addons_2024.3.0.0.tar.gz	Yes	To install AppViewX addons	Additional software to support the functionalities of AppViewX. This is mandatory for the installation.
appviewx_kubernetes_elk_2024.3.0.0.tar.gz	Optional	ELK stack to monitor logs	Additional package to install a GUI-based log collector to troubleshoot and Graffana-based UI to monitor the application performance.
appviewx_kubernetes_insight_2024.3.0.0.tar.gz	Optional	Insight for AppViewX Insight module	The insight package is an additional package to enable AppViewX to collect the statistical information of devices

Copyright © 2025 AppViewX Inc. All Rights Reserved.

File Name	Mandatory	Description	Purpose
			managed by AppViewX and generate it as a report.
upgrade.tar.gz		To upgrade from the existing version	This package is required to upgrade from older versions of AppViewX to 2024.3.0.0
prerequisite_utils.tar.gz		To check whether all the components are available.	The tool checks whether all the required prerequisites are present on the system.

Note: All OVA related updates are maintained by AppViewX and are available on the release site.

Running the Prerequisite Tool

The prerequisite tool

- **validates** for environment readiness and also **configures** some of the system configurations to ensure a smooth and successful AppViewX deployment.
- is a secure and reliable way to avoid any adverse effects on the environment.
- logs all changes made to the environment configuration for auditing and troubleshooting purposes.

Sudo permissions are required to execute the tool. This utility can be executed from any of the nodes; either worker or master. It is available in github - prerequisite utility.

The table below lists all the validations and the possible configurations performed by the prerequisite tool.

System Configuration	Validation	Configuration	Reason
Validating the system architecture - x86_64	Yes	No	System level configuration

Copyright © 2025 AppViewX Inc. All Rights Reserved.

System Configuration	Validation	Configuration	Reason
Validating RAM in worker nodes and master nodes	Yes	No	System level configuration
Validating CPU cores in worker nodes and master nodes	Yes	No	System level configuration
Validating disk space	Yes	No	System level configuration
Verifying ports communication	Yes	No	System level configuration
Cleaning up the process running on ports	Yes	Yes	NA
Validating OS	Yes	No	OS - RHEL/ CentOS/ Ubuntu
Validating OS version	Yes	No	Validating version of OS
Installing rpm/ deb dependency packages	Yes	Yes	NA
Validation for chrony and chrony sync status	Yes	Yes	NA
Validating NTP and NTP sync status	Yes	Yes	NA
Validate Runc version	Yes	Yes	NA
Validating Selinux status	Yes	Yes	NA
Validating IP of node	Yes	No	Validating the provided Ip of the nodes
Validating user id value	Yes	No	User id check
Validating umask value	Yes	Yes	NA
Validating ulimit value	Yes	Yes	NA
Validating Openssl version	Yes	No	Openssl version validation
Validating time difference between the servers	Yes	Yes	NA
Validating ftype in xfs_info	Yes	No	Checking supported file system

Copyright © 2025 AppViewX Inc. All Rights Reserved.

System Configuration	Validation	Configuration	Reason
noexec for /tmp	Yes	No	Checking permission for the /tmp directory
Validate IPV6 service	Yes	Yes	NA
Check vm.max_map_count value	Yes	Yes	NA
Server latency check	Yes	No	Latency check in and between the servers
Validate loopback IP - 127.0.0.1	Yes	Yes	NA
Validate Firewalld service	Yes	Yes	NA
Network interface lookup	Yes	No	Validating the network interface
Packet analyser	Yes	No	capture, inspect, and analyze network traffic
Analysing received packets	Yes	No	capture, inspect, and analyze network traffic
Validation of Crontab of user	Yes	Yes	NA
Validate puppet status	Yes	Yes	NA
Validate IP Tables	Yes	Yes	NA
Validate ip_forwarding	Yes	Yes	NA
Validate bridging	Yes	Yes	NA
Validating proxy variable values	Yes	Yes	NA
Cross validating SSH communication	Yes	No	Validation of ssh to nodes
Validating the GID	Yes	No	Verify that the GID we're validating corresponds to an existing group on the system
Validating User Id between the user of all servers	Yes	No	Ensuring user id assignment across servers is same

Copyright © 2025 AppViewX Inc. All Rights Reserved.

System Configuration	Validation	Configuration	Reason
Validating group Id between the user of all servers	Yes	No	Ensuring GID assignment across servers is same

To run the prerequisite tool:

1. Create a folder to download the **prerequisite.tar.gz** Eg:

   ```
   mkdir -p <folder_name>
   ```

2. Download and extract the **prerequisite.tar.gz** file to the created folder. To extract the file use the command below.

   ```
   tar -xvf prerequisite.tar.gz
   ```

3. To update the **hosts_template** execute the command below. Specify the AppViewX IP address of the VMs (master and worker nodes), DNS servers and gateway address, and users in the file.

   ```
   vi hosts_template
   ```

   Add the following values in the host_template parameters below.

Parameters	Description
NODES	It is the IP addresses of all the nodes. You can enter multiple comma separated values. Example:`NODE = 192.111.222.333`
USERS	The username for the nodes and must have sudo privileges. Example:`USERS = appviewx` In case of multi-node setup, enter the sudo username with comma separated value the number of times equivalent to number of AppViewX nodes (including masters & workers). for example, for a 2 DC 7 node setup, we need to enter the username 7 times - appviewx,appviewx,appviewx,appviewx,appviewx,appviewx,appviewx.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Parameters	Description
	Note: Usually for a multi-node setup the username for all nodes (master and slave) must be the same.
MASTER_NODES	IP address of the master node Example:`MASTER_NODES = 192.111.222.444`
WORKER_NODES	IP address of the master node Example:`WORKER_NODES = 192.111.222.555`
NEW_INSTALLATION_PATH	Example:`NEW_INSTALLATION_PATH = /home/appviewx/appviewx_cluster`
USER_GENERATED_PEM	If you require to do a password-less installation of AppViewX, enter the value as True. If set to False, then you will be prompted to enter the password after you execute the `./prerequisite` command Example:`USER_GENERATED_PEM = TRUE`
PRIVATE_KEY_FILE_PATH	Specify the path where the pem file is saved. Example:`PRIVATE_KEY_FILE_PATH = /tmp/user_generated_private.pem`
CHRONY_SYNC	If you require the time sync in the nodes enter the value as true. Example:`CHRONY_SYNC = TRUE`
CHRONY_SERVER	Specify the chrony server to be used for time sync. Example:`CHRONY_SERVER = abcs.appviewx.net`
HTTP_PROXY	If internet is not present in the nodes then use the http proxy to install the rpm/deb packages in the nodes. Example:`HTTP_PROXY = https://192.999.888.777:1234`

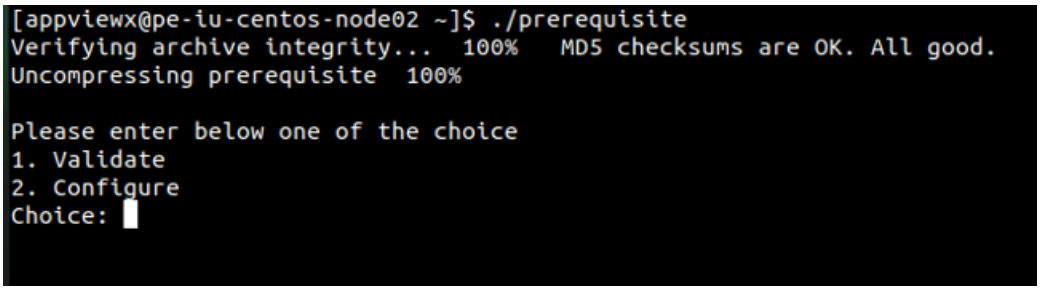
Copyright © 2025 AppViewX Inc. All Rights Reserved.

Parameters	Description
HTTPS_PROXY	If internet is not present in the nodes then use the http proxy to install the rpm/deb packages in the nodes. Example:`HTTPS_PROXY = https://192.999.888.777:1234`

4. Execute the following command:

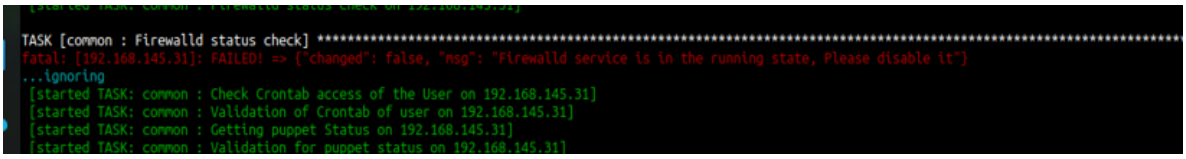
```
./prerequisite
```

The following options are displayed.



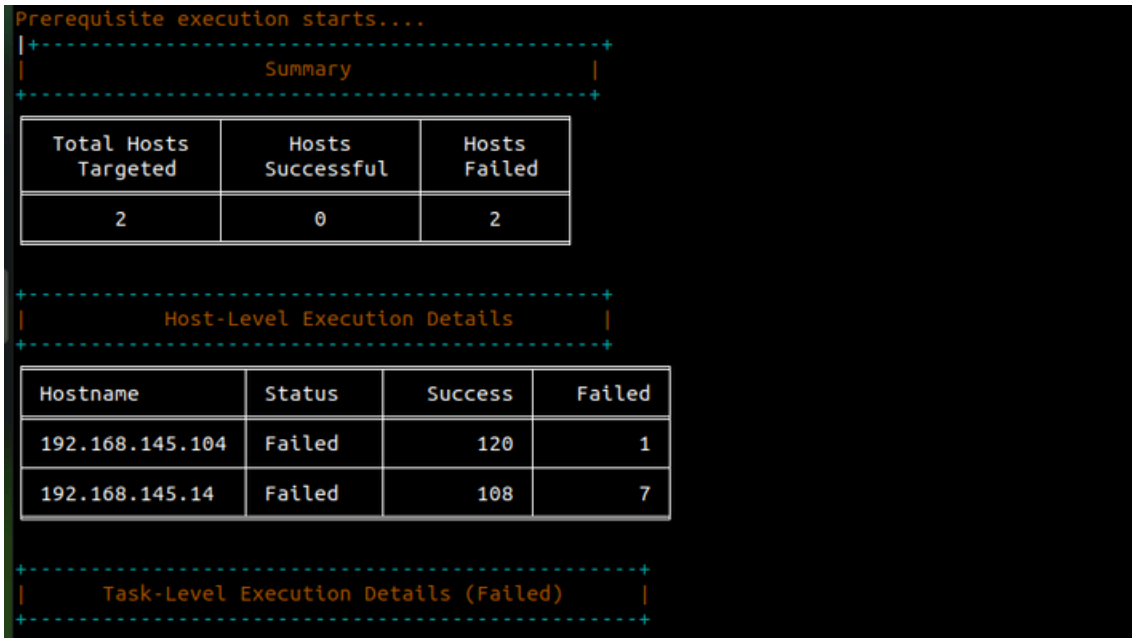
a. Enter the choice as 1 or 2

b. If you enter 1, then the tool validates all the system configurations specified in the table. If the configurations are not as per expectations the errors/failures will be displayed as shown below.



Post the validation the results are displayed as report in the format below. It shows the Summary, Host-Level Execution Status, and the Task-Level Execution Status (Failed). (*The screenshots shown are for a multi-node setup.*)

Copyright © 2025 AppViewX Inc. All Rights Reserved.



Prerequisite execution starts....

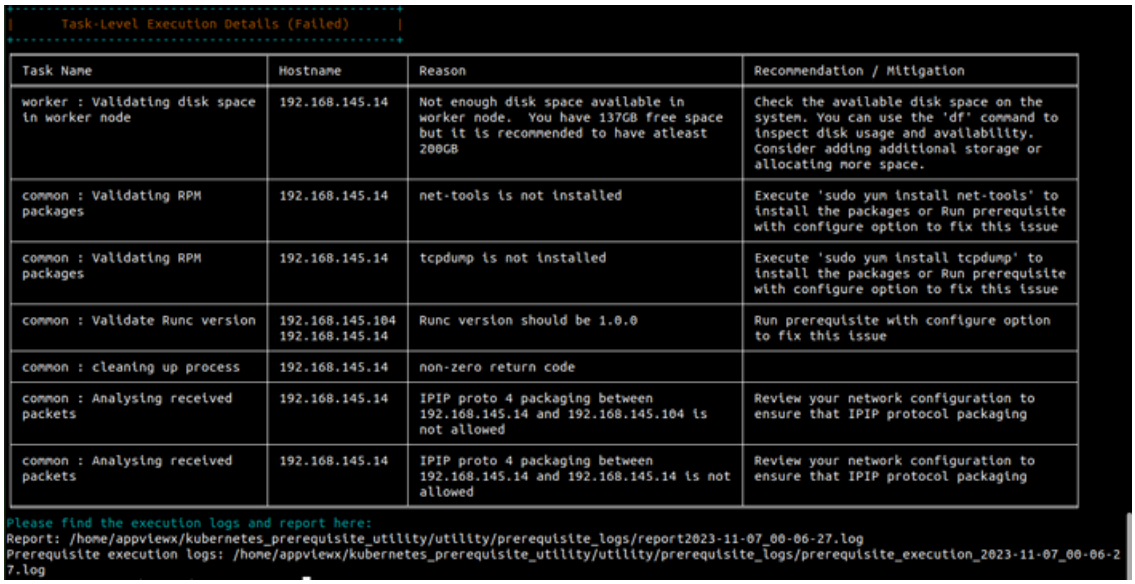
Summary

Total Hosts Targeted	Hosts Successful	Hosts Failed
2	0	2

Host-Level Execution Details

Hostname	Status	Success	Failed
192.168.145.104	Failed	120	1
192.168.145.14	Failed	108	7

Task-Level Execution Details (Failed)



Task-Level Execution Details (Failed)

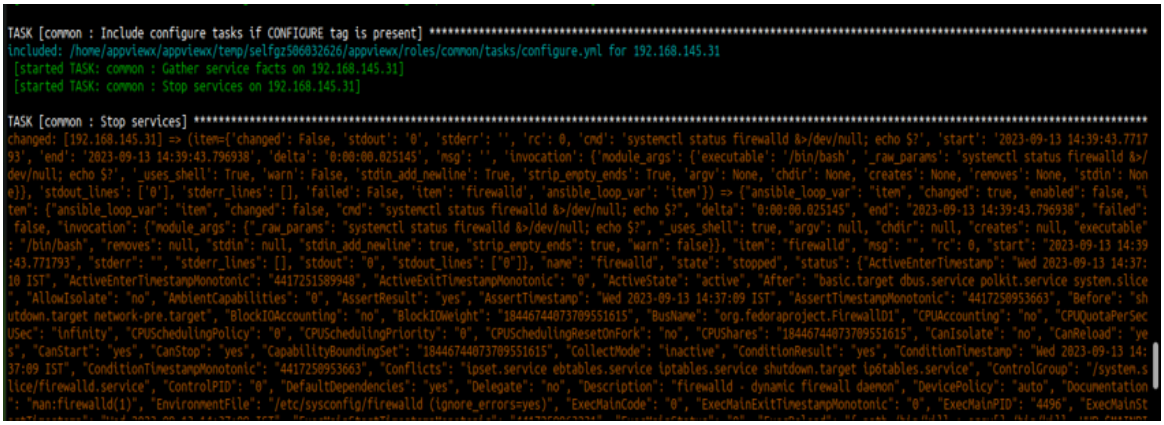
Task Name	Hostname	Reason	Recommendation / Mitigation
worker : Validating disk space in worker node	192.168.145.14	Not enough disk space available in worker node. You have 137GB free space but it is recommended to have atleast 200GB	Check the available disk space on the system. You can use the 'df' command to inspect disk usage and availability. Consider adding additional storage or allocating more space.
common : Validating RPM packages	192.168.145.14	net-tools is not installed	Execute 'sudo yum install net-tools' to install the packages or Run prerequisite with configure option to fix this issue
common : Validating RPM packages	192.168.145.14	tcpdump is not installed	Execute 'sudo yum install tcpdump' to install the packages or Run prerequisite with configure option to fix this issue
common : Validate Runc version	192.168.145.104 192.168.145.14	Runc version should be 1.0.0	Run prerequisite with configure option to fix this issue
common : cleaning up process	192.168.145.14	non-zero return code	
common : Analysing received packets	192.168.145.14	IPIP proto 4 packaging between 192.168.145.14 and 192.168.145.104 is not allowed	Review your network configuration to ensure that IPIP protocol packaging
common : Analysing received packets	192.168.145.14	IPIP proto 4 packaging between 192.168.145.14 and 192.168.145.14 is not allowed	Review your network configuration to ensure that IPIP protocol packaging

Please find the execution logs and report here:
Report: /home/appviewx/kubernetes_prerequisite_utility/utility/prerequisite_logs/report2023-11-07_00-06-27.log
Prerequisite execution logs: /home/appviewx/kubernetes_prerequisite_utility/utility/prerequisite_logs/prerequisite_execution_2023-11-07_00-06-27.log

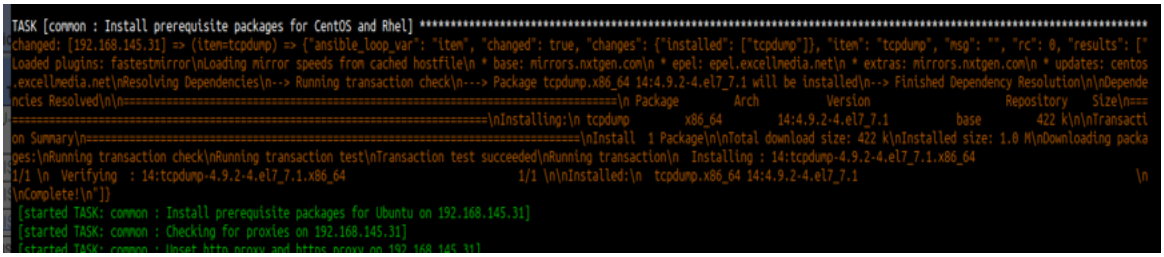
c. If you enter 2, then the tool will install the configurations as specified in the table and also validates all the system configurations.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- It stops the firewall service.



- It installs the rpm/deb package if not already installed.



d. To view the execution logs for both validate and configure, refer the **prerequisite_execution.log** file that is present in the location where the **prerequisite** file is present.

Deploying the AppViewX Virtual Appliance

An OVA file is an open virtualization appliance that contains a compressed and installable version of a virtual machine. If you are using an OVA-based installer, the installation-related artifacts are pre-bundled as part of the OVA.

The following packages are pre-bundled as part of the OVA:

- **appviewx_kubernetes_2024.3.0.0.tar.gz**
- **appviewx_kubernetes_elk_2024.3.0.0.tar.gz**
- **appviewx_kubernetes_insight_2024.3.0.0.tar.gz**
- **appviewx_kubernetes_addons_2024.3.0.0.tar.gz**
- **upgrade.tar.gz**

To deploy an OVA file,

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- Download the Release Package
- Install the AppViewX OVA

Refer the sections below.

- Downloading the Release Package
- Installing the AppViewX OVA

Downloading the Release Package

To download the release package,

1. Visit the AppViewX download URL at https://release.appviewx.com.
2. Download the release package in <.ova> format into the Downloads folder or the Desktop in your environment.
   For example, setting up a VM for a master or worker required either of the OVA
   - **appviewx_2024.3.0.0_prod_ubuntu_master.ova**
   - **appviewx_2024.3.0.0_prod_ubuntu_1TB_Worker.ova**
   - **appviewx_2024.3.0.0_prod_ubuntu_500GB_Worker.ova**
3. Validate the md5sum of the downloaded file
4. Open a terminal window.
5. To display the md5sum value of the downloaded file, execute the command:
   a. To display the md5sum value of the downloaded file, execute the following command:

   ```
   md5sum <filename>
   ```

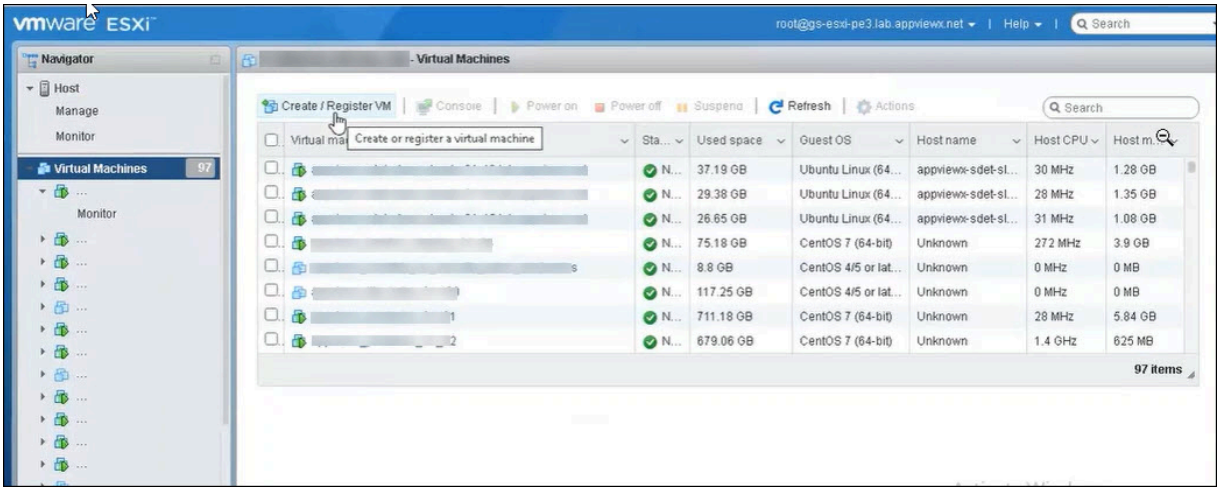
   b. Match the displayed value against the original value from the release portal.

Installing the AppViewX OVA

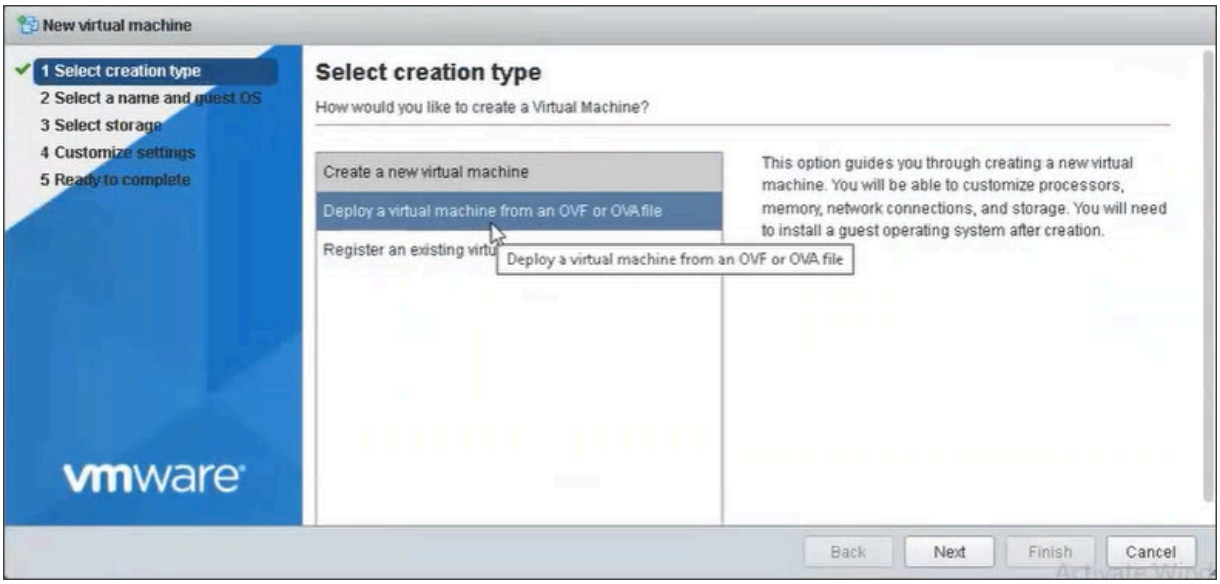
This section covers the procedures for installing the AppViewX master and worker OVA.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

1. Log into the ESXI Server, go to **Virtual Machines > Create/Register VM**.

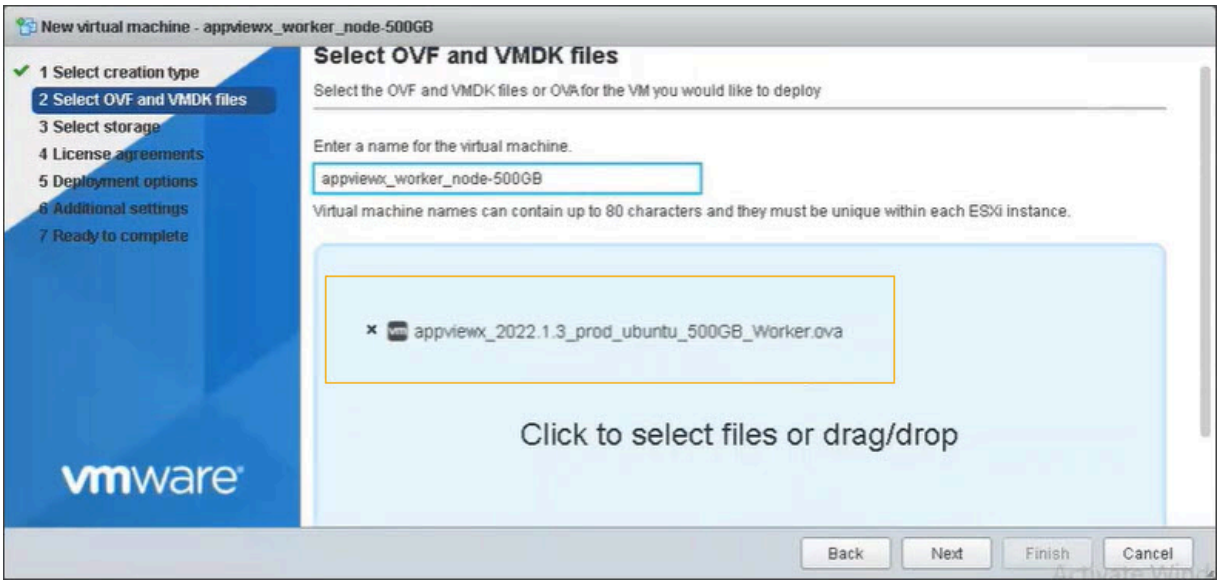


2. Select **Deploy a virtual machine from an OVF or OVA file** and click **Next**

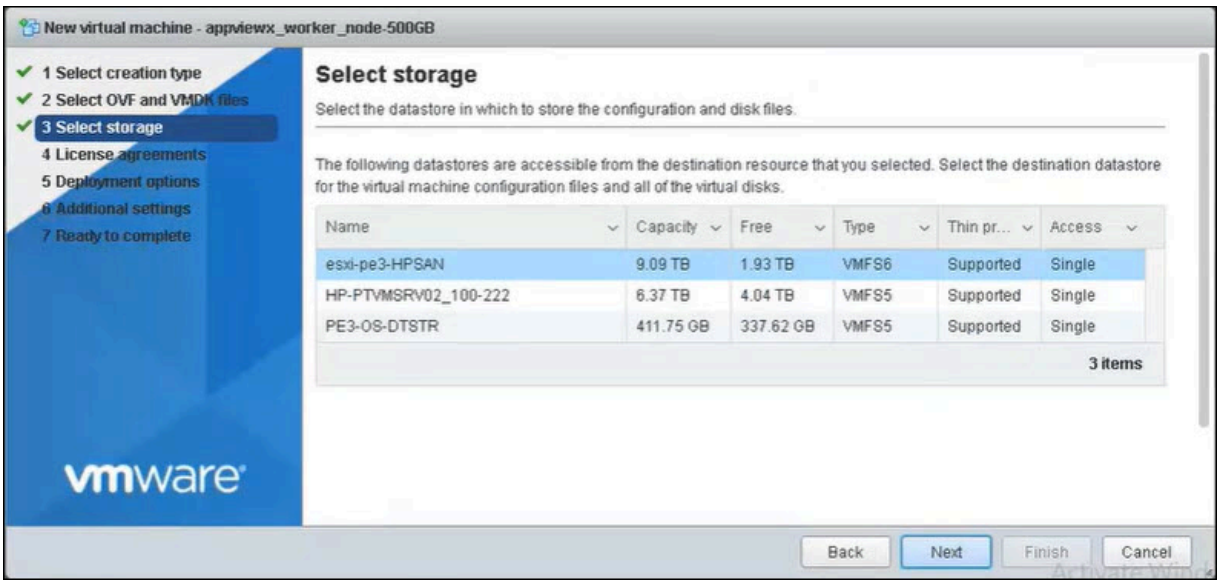


3. On the Source screen, *Enter a name for the virtual machine* in the text field provided and select **Click to select files or drag/drop option**. Click **Next**

Copyright © 2025 AppViewX Inc. All Rights Reserved.

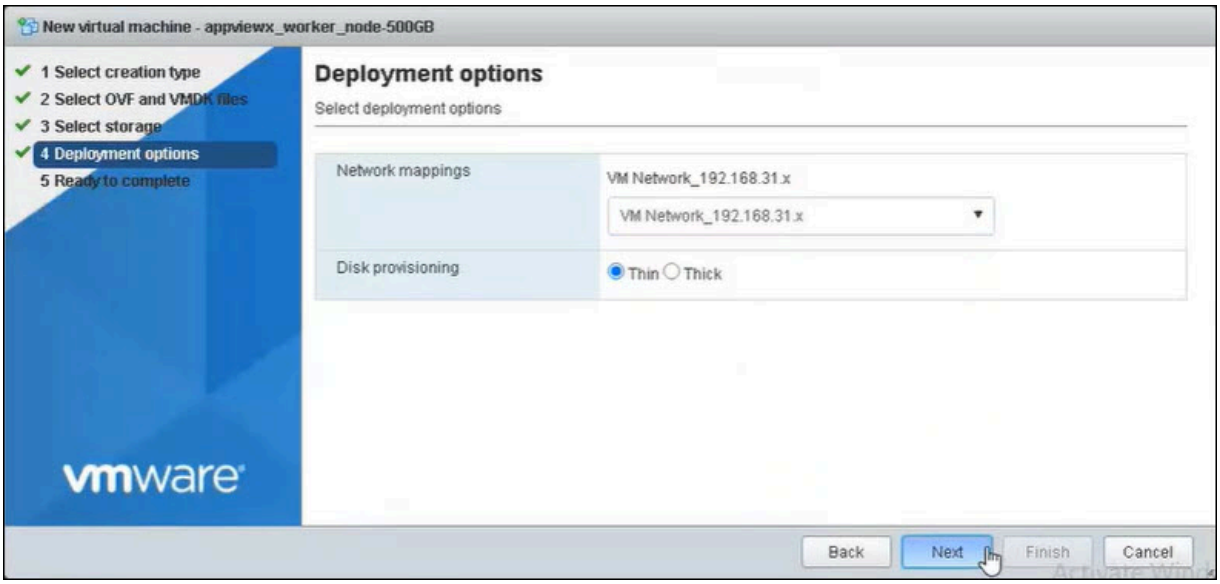


4. On the Select Storage screen, select a storage location and click **Next**.

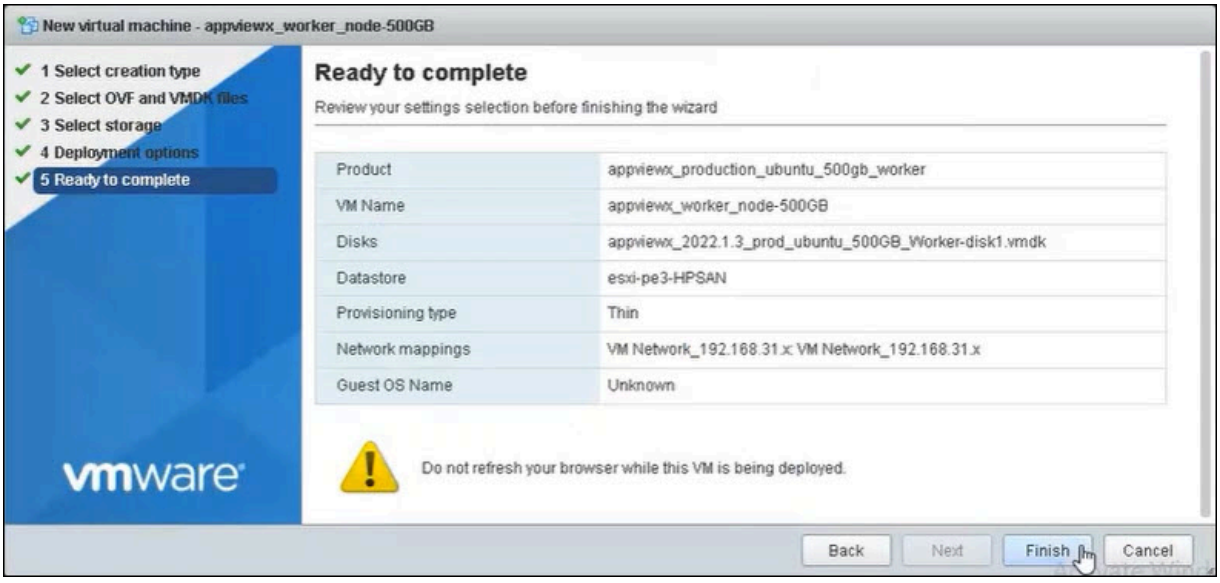


5. On the Deployment options screen, choose a network adapter in the *Network mappings* dropdown list and select the *Disk provisioning* as **Thin**. Click **Next**.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

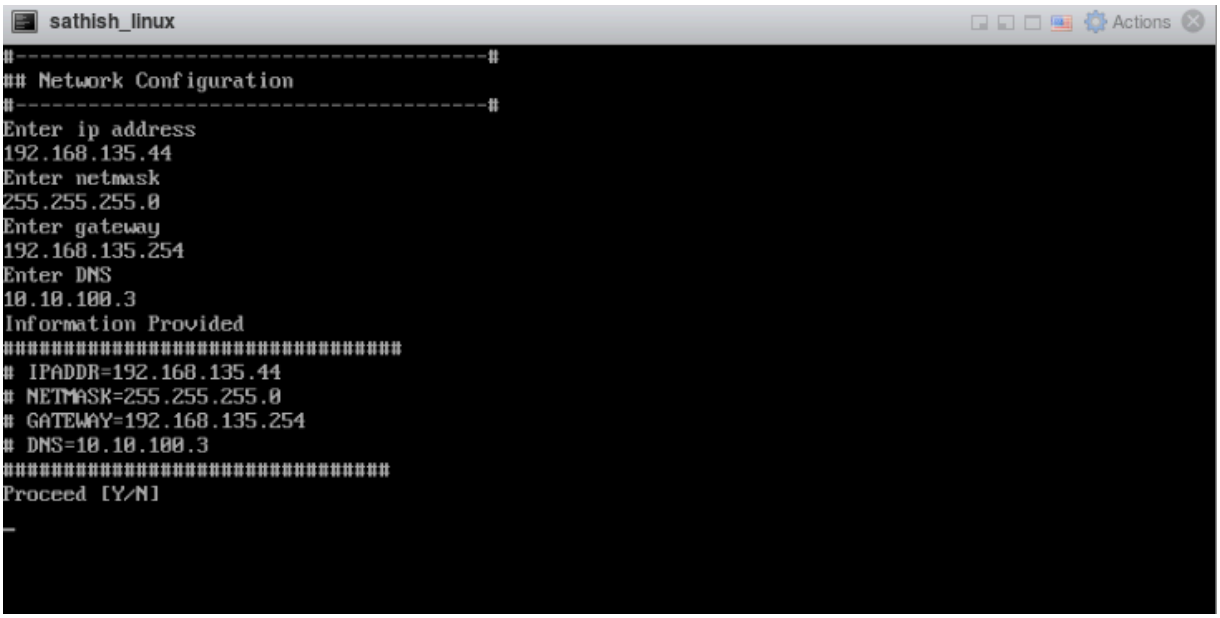


6. On the Ready to Complete screen click **Finish** to complete the OVA deployment process.



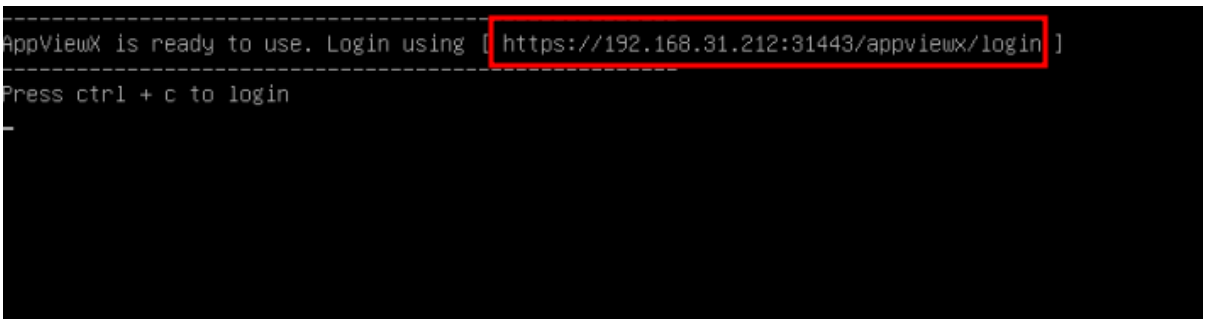
When the deployment wizard finishes, the AppViewX user interactive provisioning console opens within the ESXI server console section. You can use this console to set up your basic network configuration.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



- Type **Y** on the console screen to proceed with the network configuration.

7. After the basic network configuration process finishes, the installation starts automatically. Once the installation process completes, you can access the application by opening the browser on the host machine and entering: https://<**ip:31443/appviewx/login**>.



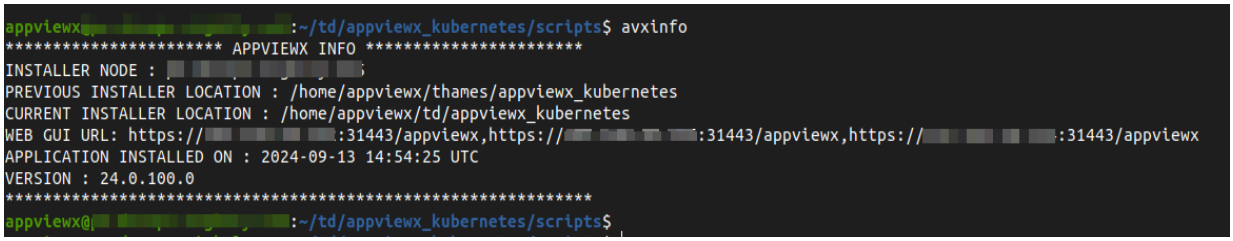
Interactive-Based Installation with Terminal UI

This section outlines several new features designed to simplify the AppViewX installation process and enhance the overall user experience. The most notable addition is the introduction of an interactive terminal-based installation UI, which enables users to generate the *appviewx.conf* file without manual steps. Other key improvements include enhanced prerequisite configuration, encrypted backup support, and better log management.

Key Features

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- *User-Friendly Installation*: The interactive terminal UI guides users through the installation process step-by-step, making it easier for users of all technical backgrounds.
- *Automated Prerequisite Configuration*: A prerequisite check and configure option helps users avoid common dependency issues, creating a more seamless installation experience.
- *Secure Backup*: MongoDB and Vault backups now feature encryption, ensuring sensitive data remains secure during transfers to external servers such as Windows SFTP. Additionally, backup success or failure alerts will be visible in the AppViewX infrastructure alerts section, with options to configure email notifications.
- *Enhanced Log Management*: Configurable *logrotate* has been introduced for MongoDB and plugin logs, streamlining log management and preventing disk space overuse through automatic log cleanup.
- *Avxinfo*: A new command, avxinfo, has been introduced to retrieve details such as the installer node and installation location. This command can be executed from any node within the cluster.



- *Auto Remediation*: This feature automatically conducts checks on the Kubernetes platform, identifies and resolves issues, and collects log files to determine the root cause of each problem. It empowers customer success teams and customers to rectify issues themselves without the need of reaching out to the engineering team. Newly identified issues will be incorporated into this tool for future handling.
- *Collect Logs*: This feature automates the process of collecting and archiving critical data, such as application logs, system logs, and TCP dump files.
- *Apply Patch*: This feature facilitates updating the existing AppViewX setup by applying plugin patches, add-on patches, or both, ensuring the system stays up-to-date and functional.

Note: If you do not have a deployment model defined yet, contact help@appviewx.com

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Warning:

- It is critical that you execute the prerequisite tool before installing AppViewX.
- Before you start the installation, ensure that the node password does not contain special characters such as single quote (**'**), double quote (**"**), and back slash (****), ampersand (**&**), comma (**,**) semicolon (**;**) or a combination of special characters such as **%{** and **${**.
- Upgrading from earlier versions is not supported in v2021.1.0. A new install is the only option.

- Prerequisites
- Adding Third-party Libraries
- Fresh Installation
- Upgrade
- Data Restore
- Collect Logs (for Troubleshooting)
- Auto-Remediation
- Apply Patch
- Compare Configuration
- Points to Remember

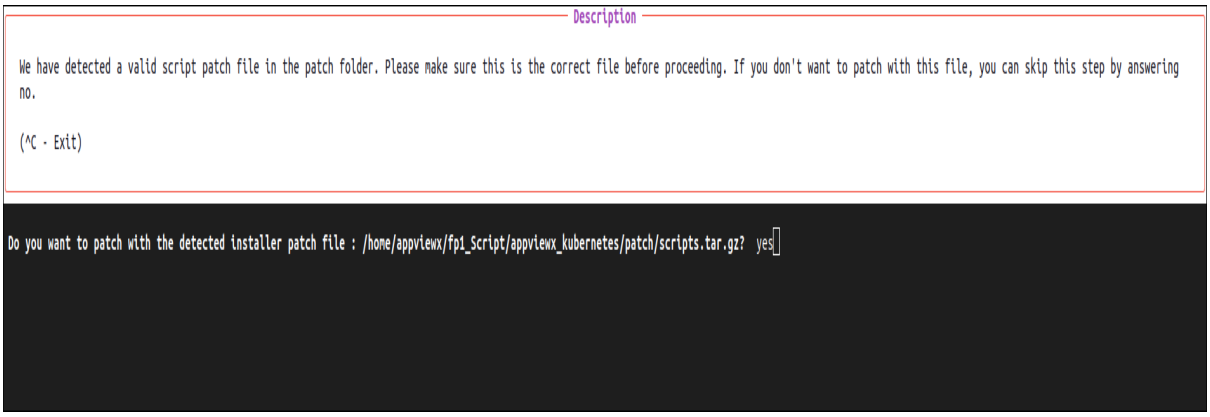
Prerequisites

Before beginning the installation, ensure that the following are prepared:

- If any operation is being carried out using the interactive tool, these steps should be followed.
  - If a script patch is released for the version, make sure to download the patch.
  - Move the patch to the <installer_path>/patch directory.(e.g.: /home/appviewx/appviewx_kubernetes/patch)

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- Trigger the `install.sh` script and respond with '**yes**' to any prompts that appear, as shown in the image below.



These steps are essential to ensure that the tool operates with the latest fixes.

- Sudo access to the configured for the appviewx installation user
- If the user opts for a key-based installation, passwordless sudo must be configured. As the root user, add the following lines to the **/etc/sudoers** file:

```
appviewx ALL=(ALL) NOPASSWD:ALL
```

- Load balancer (LB) for the AppViewX GUI (*optional - refer note below*)
- Kubernetes master L4 load balancer (*optional - refer note below*)
- A **.p12** certificate for the AppViewX Web GUI - An external signed certificate is typically issued by a trusted Certificate Authority (CA) and is used to authenticate and encrypt communication between the web server and clients, ensuring data integrity and security. (*optional - refer note below*).
- NTP server details
- Nameserver details
- Proxy or internet access is required on all the nodes if any of the following OS prerequisite packages mentioned below are to be installed during the prerequisite check. Ignore, if the packages are already installed on the nodes.
  - **Ubuntu** - curl, net-tools, nmap, zip, unzip, sysstat, rsync, tcpdump, chrony, bind9-utils, dnsutils, ebtables, netcat, netcat-openbsd
  - **RHEL** - curl, net-tools, nmap, zip, unzip, sysstat, rsync, tcpdump, chrony, bind-utils, nmap-ncat, (ebtables, iptables-ebtables, iptables-nft) any one based on OS version
- If you wish to enable Mongo/Vault backups to the SFTP server, you must specify the server where the Database and Vault backups will be securely stored. Ensure that the IP address or hostname is

Copyright © 2025 AppViewX Inc. All Rights Reserved.

accurate, accessible from your network, and has enough storage space to avoid any connectivity or capacity issues during the backup process.

- The HSM client must be configured on the HSM node. For a **single node**, the configuration is done on the node itself. For a **multi-node** setup, the HSM nodes are selected and configured accordingly.

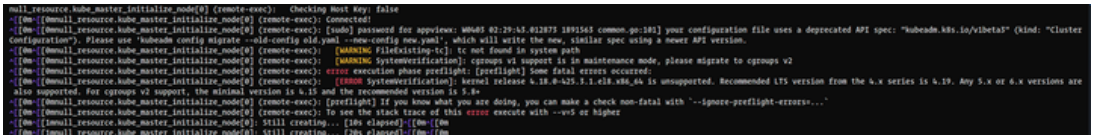
Note:

- **The Load Balancers** and the **.p12** certificate prerequisites are optional during the installation process but are recommended to be enabled for High Availability (HA). This will help complete the installation or upgrade smoothly in one attempt.
- If a custom workflow is being used, please contact AppViewX technical support before proceeding with the upgrade.

Updating the Kernel version

Following the upgrade to **Kubernetes v1.32.2**, the `kubeadm init` setup is no longer compatible with **older Linux kernel versions**. This is primarily due to enhanced reliance on **cgroups** (control groups), a Linux kernel feature that manages resource isolation and allocation. This incompatibility typically affects nodes running older operating system versions, particularly those in the **RHEL 8 series** (e.g., 8.5, 8.6, or even 8.10), depending on the specific kernel version present on the node.

Figure 1. Error log from kubernetes



Cgroups Overview:

Control Groups (cgroups) are a kernel-level feature that enables the limitation, prioritization, and isolation of resource usage (CPU, memory, I/O, etc.) among process groups. Kubernetes leverages cgroups extensively for container orchestration.

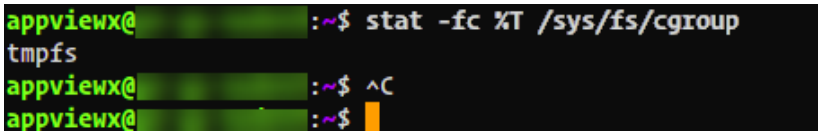
Determining the Cgroups Version in Use:

To identify the active cgroups version on a Linux system, run the following command:

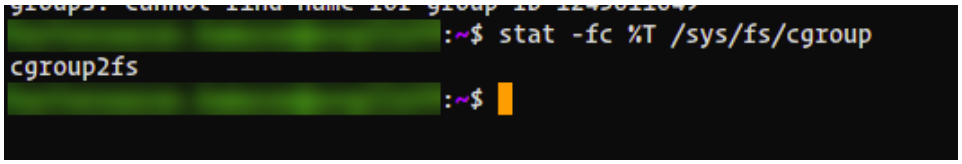
```
stat -fc %T /sys/fs/cgroup
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- If the output is `tmpfs`, the system is using **cgroups v1**.



- If the output is `cgroup2fs`, the system is using **cgroups v2**.



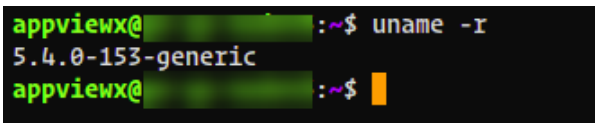
Kernel Version Requirements Based on Cgroups Version

Cgroups Version	Minimum Kernel Version	Recommended Kernel Version
v1	4.19+	5.x or 6.x series
v2	4.15+	5.8 or later

To ensure compatibility with Kubernetes 1.32.2, it is recommended to validate and, if necessary, upgrade the Linux kernel version in accordance with the cgroups configuration of the host system.

To verify the kernel version, execute the command:

```
uname -r
```



Solution provided:

Since the issue is not OS-specific, the prerequisite scripts now contain the generic checks during both the installation and application upgrade processes.

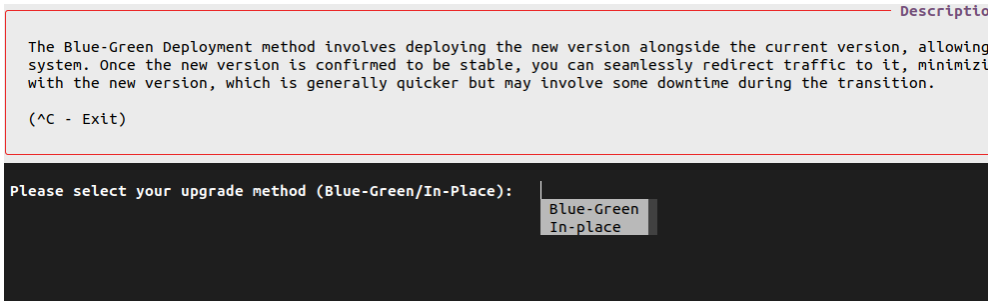
- If the system does not meet the minimum required kernel version, the installation will be blocked from proceeding.
- Additionally, if the system is using cgroups v1, a warning message will be displayed recommending an upgrade to cgroups v2 for improved compatibility and performance.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Keyboard Shortcuts for Install and Upgrade Operations

- *Ctrl* + *Z*: Go to the previous question
- *Ctrl* + *L*: Collect the logs
- *Ctrl* + *W*: Scroll up
- *Ctrl* + *S*: Scroll down
- *Ctrl* + *U*: Unpause scrolling
- *Ctrl* + *C*: Cancel the installation or upgrade
- *Ctrl* + *T*: Show or hide the password
- *Ctrl* + *E*: Show the error message
- *Ctrl* + *P*: To rollback during apply patch failures
- *Ctrl* + *R*: To resume
- *Ctrl+W*: To scroll up
- *Ctrl+S*: To scroll down
- *Ctrl+A*: To scroll left
- *Ctrl+D*: To scroll right
- *V* : View verbose logs
- *Tab*: Press the tab key for suggestions to questions

Example below shows options for upgrade when the Tab key is pressed.



Adding Third-party Libraries

AppViewX requires specified libraries to manage and control the devices. These libraries are specified by the manufacturers of the devices. AppViewX will be able to communicate with the devices only when these libraries are installed.

Please follow the steps in this section to add external proprietary jars in AppViewX.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- If the customer wants to use any third party integrations with earlier versions of AppViewX, ensure that the **.jar** files for these integrations are downloaded and extracted in the Installer/external_lib directory before the migration/installation process.
- If the customer wants to use any third party integrations with earlier versions of AppViewX after migration or installation, ensure that the corresponding **.jar** files are downloaded and extracted to the /home/appviewx/appviewx/external_libs directory.

• iControl F5 Integration
• Thales
• Safenet/Gemalto

iControl F5 Integration

iControl is an open API that enables applications to work in sync with the network based on the software integration. iControl uses SOAP/XML to ensure an open communication between dissimilar systems. It helps F5 customers, independent software vendors (ISVs), and solution providers leverage efficiency in automation and management of network objects and devices.

Users who want to use third party integrations to control their devices can integrate the required .jar file. The process begins with the user downloading the .jar file from the respective vendor. After downloading, the contents of the .jar file must be extracted into the external_libs directory. Finally, the plugin must be restarted for the changes to take effect.

1. To integrate the iControl library into the required project, copy the library and paste it into the <user_home_dir>/installer/external_libs/ directory (create a directory if it does not exist).
2. Visit devcentral f5 download page URL: iControl Library For Java With Source.
3. Download the latest iControl integration library file from the list of libraries.
4. Extract the downloaded zip file to: iControlAssembly_13_1_0-Java.
5. Download the axis.jar from the axis library.
6. Download the **javax.xml.soap-api-1.4.0.jar** from the file library.
7. Copy the **iControl.jar**, **axis.jar**, and **javax.xml.soap-api-1.4.0.jar** files from the extracted package to the **external_libs** directory.
8. If AppViewX is already installed or upgraded from an earlier version of AppViewX, move the **iControl-13.1.0.jar**, **axis.jar**, and **javax.xml.soap-api-1.4.0.jar** file to external_libs directory using command below

```
cp -r /lib/iControl-13.1.0.jar <user_home_dir>/appviewx_dependencies/external_libs/ directory
```

```
cp -r axis-1.4.jar <user_home_dir>/appviewx_dependencies/external_libs/ directory
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
cp -r javax.xml.soap-api-1.4.0.jar <user_home_dir>/appviewx_dependencies/external_libs/ directory
```

9. If AppViewX is not installed, move the **iControl-13.1.0.jar**, **axis.jar**, and **javax.xml.soap-api-1.4.0.jar** file to external_libs directory using command

```
cp -r /lib/iControl-13.1.0.jar /home/appviewx/Installer/external_libs
```

```
cp -r axis-1.4.jar /home/appviewx/Installer/external_libs
```

```
cp -r  javax.xml.soap-api-1.4.0.jar /home/appviewx/Installer/external_libs
```

10. In case of a multi node environment, copy the **iControl-13.1.0.jar**, **axis.jar**, and **javax.xml.soap-api-1.4.0.jar** file to all the servers where the **avx_vendors** plugin is running.

> **Note:** To restart the avx_vendors plugin followed by the gateway plugin, refer to the Restarting a plugin section.

Thales

Users who want to use third party integrations to control their devices can integrate the required .jar file. The process begins with the user downloading the .jar file from the respective vendor. After downloading, the contents of the .jar file must be extracted into the **external_libs** directory. Finally, the plugin must be restarted for the changes to take effect.

> **Note:** Install the Thales client only on the node where AppViewX is installed.

1. To navigate to the directory where Thales client is installed, execute the following command:

```
cd /opt/nfast/java/classes
```

2. Copy the jutils, kmjava, and nfjava jars from the directory and paste it to the external libs directory in AppViewX.
   - If AppViewX is already installed /migrated, execute the following command:

```
cp <jar_name>.jar <user_home_dir>/external_libs/
```

   - If AppViewX is not installed/migrated, to copy the jar in the installer directory, execute the following command:

```
cp <jar_name.jar /home/appviewx/Installer/external_libs
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

3. Restart the **avx_vendors** plugin followed by the gateway plugin.

> **Note:** For more information on how to restart the plugin, refer to the Restarting a plugin section.

Safenet/Gemalto

Users who want to use third party integrations to control their devices can integrate the required .jar file. The process begins with the user downloading the .jar file from the respective vendor. After downloading, the contents of the .jar file must be extracted into the external_libs directory. Finally, the plugin must be restarted for the changes to take effect.

> **Note:** Install the Safenet client only on the node where AppViewX is installed.

1. To navigate to the directory where Safenet is installed, execute the following command:

```
cd /usr/safenet/lunaclient/jcprov/lib
```

2. Copy the **jcprov jar** from the directory and paste it to the **external_lib** directory in AppViewX.
   - If AppViewX is already installed /migrated:

```
cp jcprov.jar <user_home_dir>/appviewx_dependencies/external_libs/
```

   - If AppViewX is not installed/migrated, copy the jar in the installer directory:

```
cp jcprov.jar /home/appviewx/Installer/external_libs
```

3. Restart the **avx_vendors** plugin followed by the gateway plugin.

> **Note:** For more information on how to restart the plugin, refer to the Restarting a plugin section.

Fresh Installation

1. Copy all the downloaded packages to the server.

> **Note:** The AppViewX installation must start from the node that is selected for the primary MongoDB host. For example, the first node specified under the MONGODB_HOST property in the **appviewx.conf** file.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

2. SSH to the server in which packages are copied.
3. Open the terminal.
4. To extract the contents of the **appviewx_kubernetes_2024.3.0.0.tar.gz** file, execute the following command:

```
tar -xvf appviewx_kubernetes_2024.3.0.0.tar.gz
```

5. To move the **appviewx_kubernetes_addons_2024.3.0.0.tar.gz** file to the **appviewx_kubernetes** folder, execute the following command:

```
mv appviewx_kubernetes_addons_2024.3.0.0.tar.gz appviewx_kubernetes/
```

6. To navigate to the <InstallerLocation>/appviewx_kubernetes/scripts directory, execute the following command:

```
cd <InstallerLocation>/appviewx_kubernetes/scripts
```



7. To start the installation process, run the command:

```
./install.sh
```

The following welcome screen is displayed.



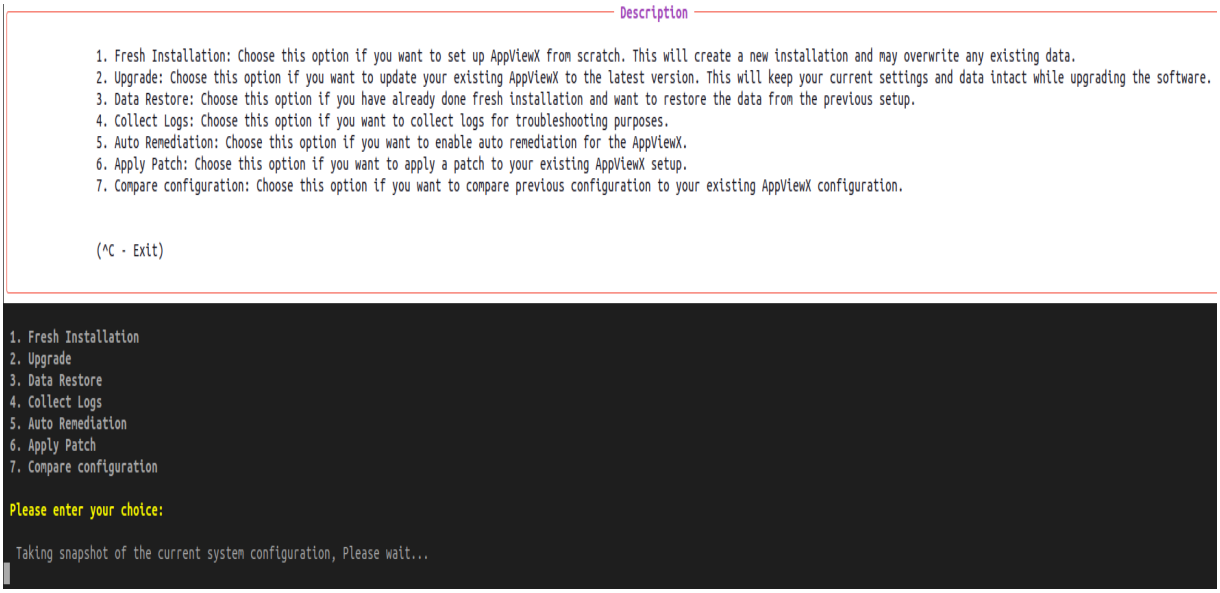
8. Select **Continue**.

The screen shows the following options:

- 1. Fresh Installation
- 2. Upgrade
- 3. Data Restore
- 4. Collect Logs
- 5. Auto Remediation

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- 6. Apply Patch
- 7. Compare Configuration



9. To install the AppViewX application from scratch, Enter **1** (Fresh Installation).

> **Note:** In case the application is already installed, you will be prompted to uninstall and the proceed with the installation.
>
> 
>

10. Follow the interactive steps provided by the Terminal UI to configure the settings for generating the appviewx.conf file.



Copyright © 2025 AppViewX Inc. All Rights Reserved.

> **Note:** If an appviewx.conf file already exists, you will be prompted to continue with the existing file. However, validations will occur for each configuration step.

11. You will be prompted with the following questions:
    a. *Do you have an external Signed certificate for Appviewx Web UI?* If you have such a certificate (.p12) for AppViewX Web GUI, please respond with **Yes**.
    b. *Would you like to enable HSM plugin ?* If you have already installed HSM as per the prerequisites, respond with **Yes**.
12. Ensure all required inputs are provided, and the system will automatically generate the configuration file, ready for use.
13. Once all the questions are answered, you will be prompted with a final table displaying the provided details. Review the information carefully, and type '**y**' to proceed with the prerequisites validation.



Description	Value
Installation Type	Multi Node
AppViewX will be installed using this username	appviewx
SSH port	22
Authentication method for SSH connection	password
Total number of data centers	2
All Data Center Name(s)	dc1,dc2
Hostname(s) / IP address(es) of master nodes	[redacted]
Names of Datacenter(s) with Master VM(s)	dc1, dc2
Master VM(s) in dc1	[redacted]
Master VM(s) in dc2	[redacted]
Worker VM(s) in dc1	[redacted]
Worker VM(s) in dc2	[redacted]
All VM(s) in dc1	[redacted]
All VM(s) in dc2	[redacted]
Hostname / IP address of the TCP Load Balancer	[redacted]
TCP Load balancer port	6443
Hostname(s) / IP address(es) of all Database nodes	[redacted]
Hostname(s) / IP address(es) of all Vault nodes	[redacted]
Hostname(s) / IP address(es) in which the AppViewX GUI will be accessible	[redacted]
External certificate for the AppViewX web interface	/home/appviewx/on-premise.p12
Hostname(s) / IP address(es) of all HSM nodes	[redacted]
The directory where AppViewX will be installed	/home/appviewx/appviewx_app
Remote Server IP or Hostname for SFTP Transfer for Database and Vault backup	[redacted]

you wish to continue (y/n)?

^S to scroll down and ^W to scroll up

14. **Prerequisites validation** - The prerequisites validation will be performed based on the information provided.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



15. **Configuring Prerequisites** - If any prerequisites are missing, the Prerequisite Configure option allows the installer to automatically install or configure the necessary packages. Press '**y**' to continue with the configuration.



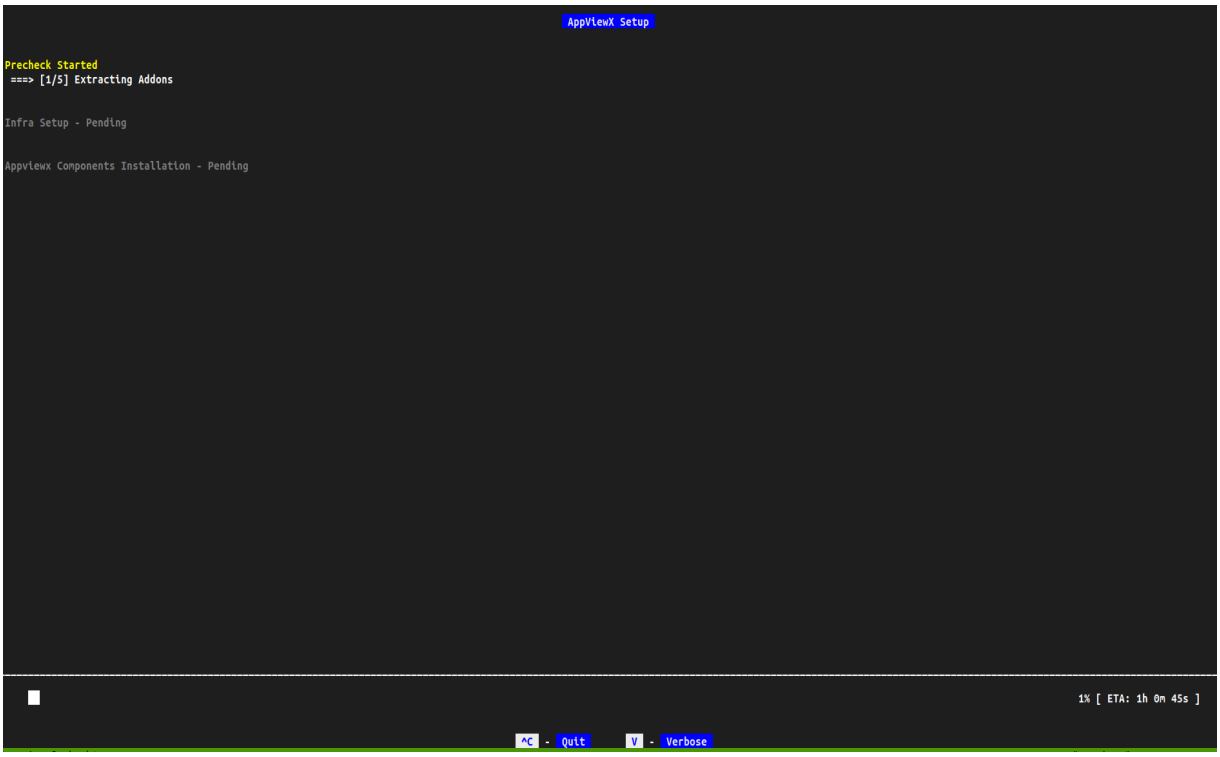
AppViewX Prerequisite Report

Validation	Nodes	Reason	Mitigation
Validating Disk Space: [redacted] ⚠(Warning)	[redacted]	Insufficient disk space on installation path (103 GB available). Minimum recommended: 250GB.	Insufficient disk space available on worker node Recommended minimum: 250GB
Validating Disk Space: [redacted] ⚠(Warning)	[redacted]	Insufficient disk space on installation path (124 GB available). Minimum recommended: 250GB.	Insufficient disk space available on worker node Recommended minimum: 250GB
Validating Disk Space: [redacted] ⚠(Warning)	[redacted]	Insufficient disk space on installation path (105 GB available). Minimum recommended: 250GB.	Insufficient disk space available on worker node Recommended minimum: 250GB

> **Note:** Installing these prerequisite packages requires internet access or proxy access to download and install the missing packages.

16. Once all the prerequisites are validated, the AppViewX installation will be triggered.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



17. To view the verbose logs, press '**v**.'



Copyright © 2025 AppViewX Inc. All Rights Reserved.

Upgrade

Note: Ensure that a valid **other_user_internal.pem** file is present in the <appviewx installer directory>/scripts directory of the old installer for the in-place and blue-green deployments, where the existing setup is live.

1. Copy all the downloaded packages to the server.

   Note: The AppViewX installation must start from the node that is selected for the primary MongoDB host. For example, the first node specified under the MONGODB_HOST property in the **appviewx.conf** file.

2. SSH to the server in which packages are copied.
3. Open the terminal.
4. To extract the contents of the **appviewx_kubernetes_2024.3.0.0.tar.gz** file, execute the following command:

   ```
   tar -xvf appviewx_kubernetes_2024.3.0.0.tar.gz
   ```

5. To move the **appviewx_kubernetes_addons_2024.3.0.0.tar.gz** file to the **appviewx_kubernetes** folder, execute the following command:

   ```
   mv appviewx_kubernetes_addons_2024.3.0.0.tar.gz appviewx_kubernetes/
   ```

6. To navigate to the <InstallerLocation>/appviewx_kubernetes/scripts directory, execute the following command:

   ```
   cd <InstallerLocation>/appviewx_kubernetes/scripts
   ```

7. To start the upgrade process, use the installer command:

   ```
   ./install.sh
   ```

   A welcome screen is displayed with options to Continue with installation or Exit.

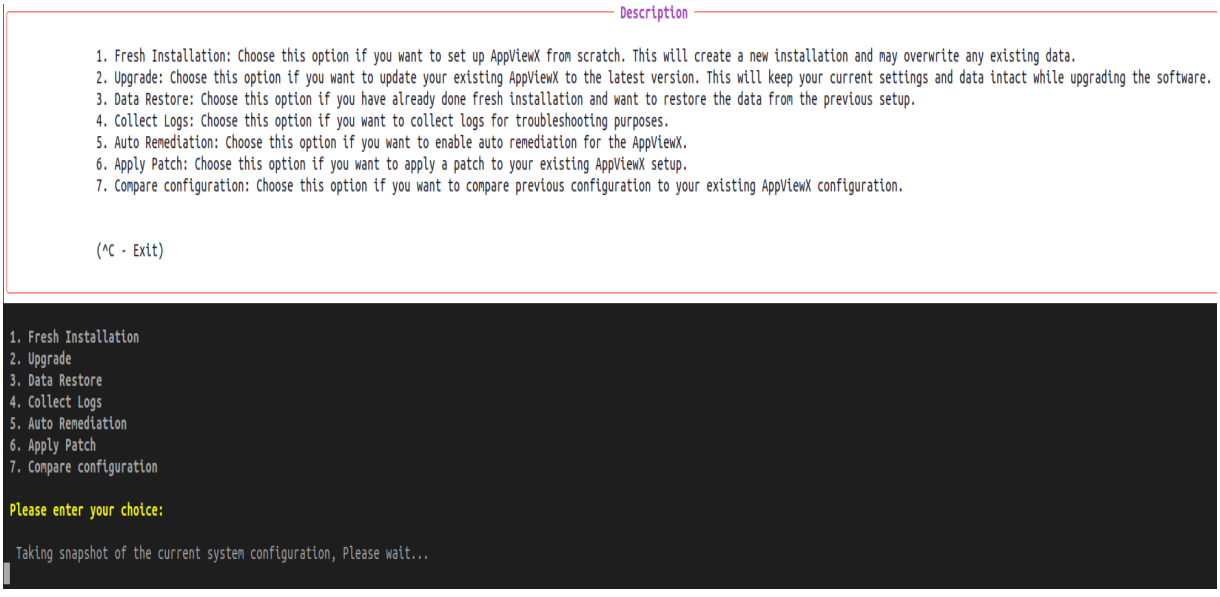
Copyright © 2025 AppViewX Inc. All Rights Reserved.



8. Select **Continue**.

   The screen shows the following options:

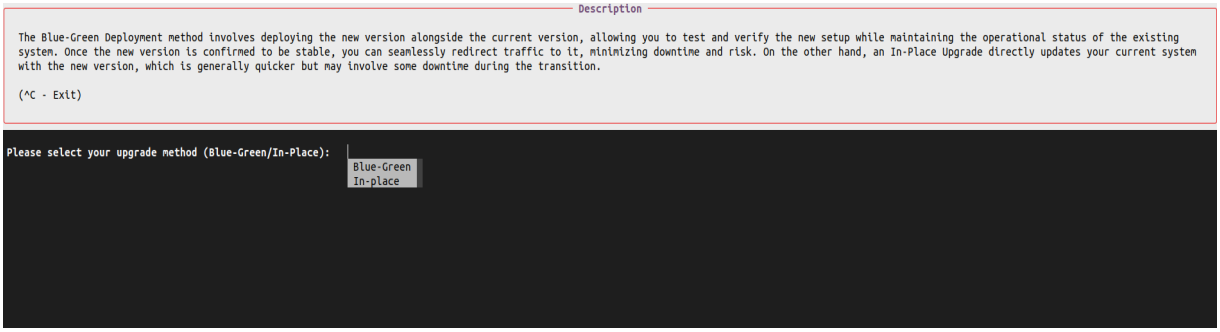
   - 1. Fresh Installation
   - 2. Upgrade
   - 3. Data Restore
   - 4. Collect Logs
   - 5. Auto Remediation
   - 6. Apply Patch
   - 7. Compare Configuration



9. To upgrade the AppViewX application, choose option **2**.

   The screen displays a prompt "*Please select your upgrade method (Blue-Green/In-Place):*"

Copyright © 2025 AppViewX Inc. All Rights Reserved.



10. Now, choose between two types: Blue-Green Upgrade and In-Place Upgrade.
    Choose the upgrade option based on the current transition type:

    Note: All prerequisite validations will be carried out during the upgrade, and the `appviewx.conf` file can be modified during the interactive question phase.

    - **Blue-Green**: If this option is selected, prepare the previous installer node details, installer path, and decide whether to take new hot backups or use an existing backup for the migration. There are two options
      - Select **Yes** if you are retaining the same design, where both the existing and new clusters have an identical number of nodes, to migrate without modifying the configuration.
      - Select **No** to allow the system to automatically generate the `appviewx.conf` file based on the inputs provided.
    - **In-Place**: This option involves downtime, so ensure that the necessary downtime is planned and approved before proceeding with the in-place upgrade. The in-place upgrade retains the existing configuration, but validations will take place during the interactive questions and confirmations. You may add or remove nodes during the interactive questions phase.
11. You will be prompted with the following questions:
    a. *Do you have an external Signed certificate for Appviewx Web UI?* If you have such a certificate (.p12) for AppViewX Web GUI, please respond with **Yes**.
    b. *Would you like to enable HSM plugin ?* If you have already installed HSM as per the prerequisites, respond with **Yes**.
12. After the upgrade completes, to clean up the temporary data run the command below:

```
source appviewx.conf
```

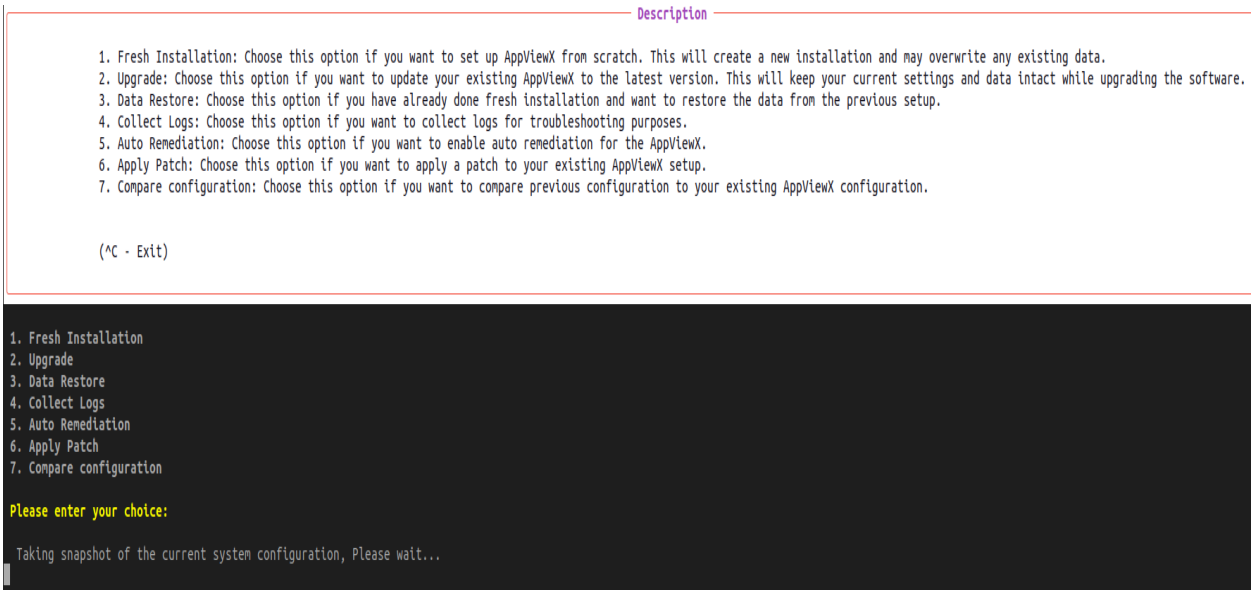
```
./appviewx.sh --run-on-all "rm -rf $INSTALLATION_PATH/appviewx_dependencies/seed/*"
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Data Restore

This option should be used when the customer is ready to proceed with the final database and vault restoration in preparation for the cut-over and go-live phase. Choose option 3 for Data Restore.

Note: Ensure that the **appviewx_kubernetes_addons_2024.3.0.0.tar.gz** file is present in the **appviewx_kubernetes** directory.



After the data restore is complete, run the command below from the installer location:

```
source appviewx.conf
```

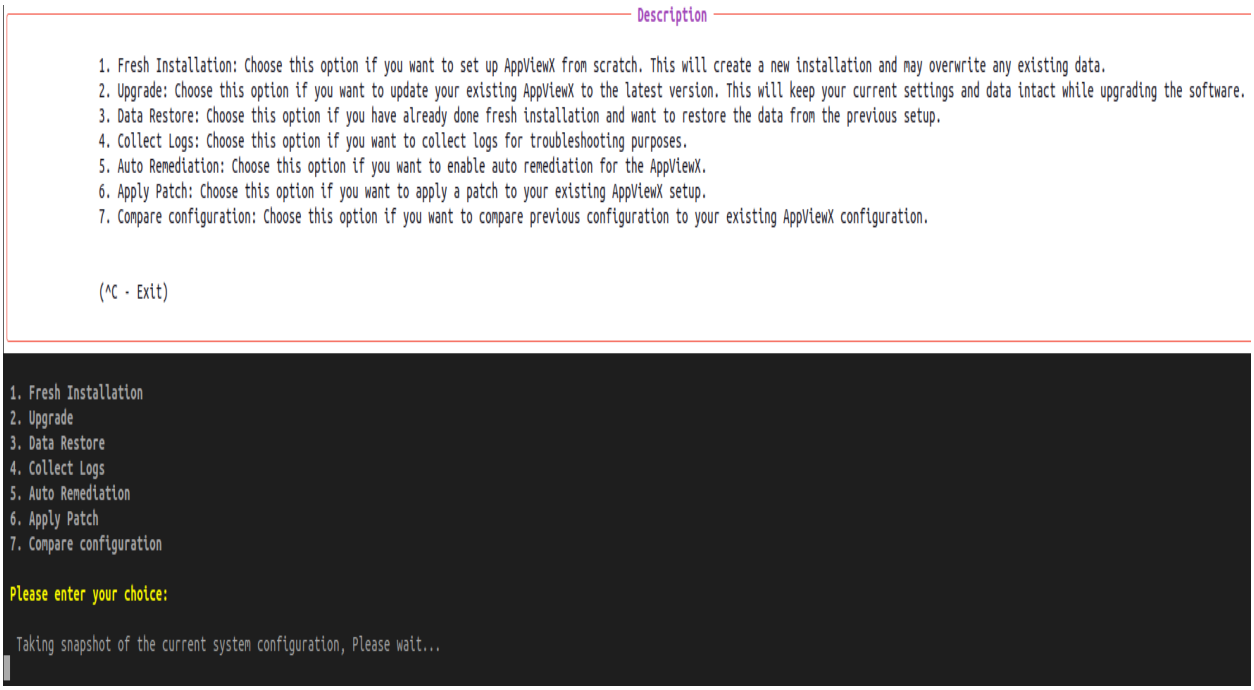
```
sudo ./appviewx.sh --run-on-all "rm -rf $INSTALLATION_PATH/logs/mongo_backup_*"
```

Collect Logs (for Troubleshooting)

This option should be used to gather application logs, system logs, and other relevant logs for troubleshooting in case of any issues with the application. Choose option 4 for collecting logs.

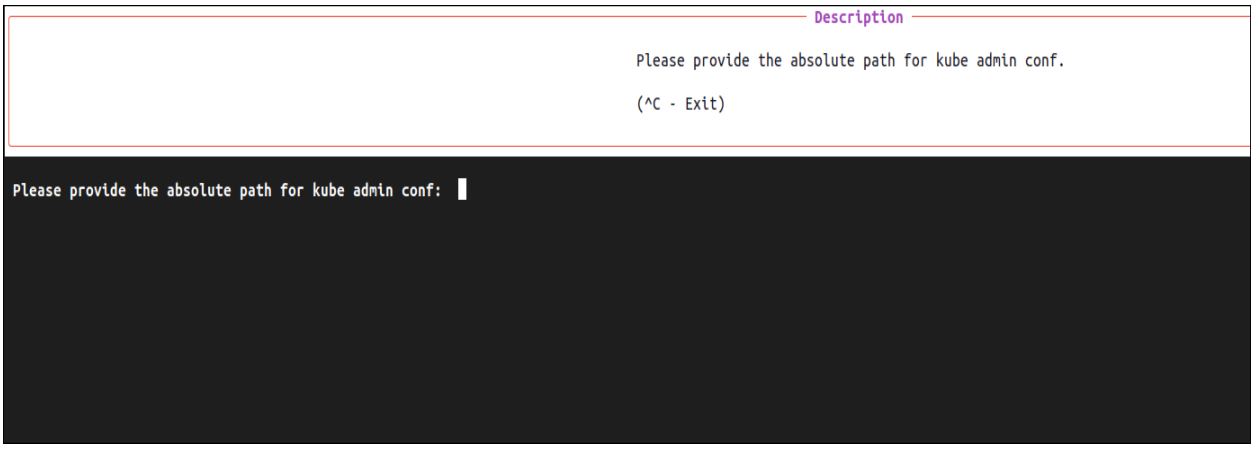
Note: Ensure that the **appviewx_kubernetes_addons_2024.3.0.0.tar.gz** file is present in the **appviewx_kubernetes** directory.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



By default, the cluster has read-only access. To perform administrative tasks, admin access must be enabled. A configuration file located at <Home Directory>/.kube/backup/config provides the required admin privileges. When prompted, ensure you specify the correct path to this file.

Note: If the file has been moved or backed up to a secure location, provide the updated file path accordingly.

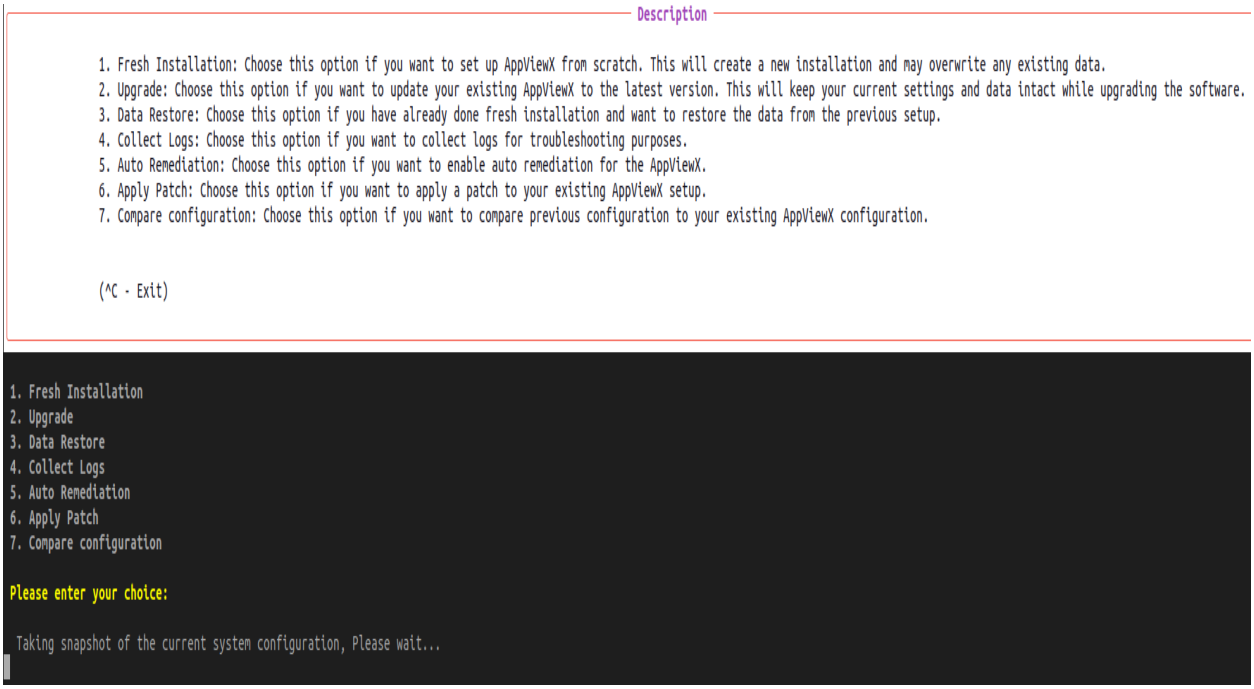


Copyright © 2025 AppViewX Inc. All Rights Reserved.

Auto-Remediation

This option should be used during an outage in the customer environment. It scans the entire cluster, performs remediation if any issues are detected, and collects logs for root cause analysis (RCA). Choose option 5 for Auto-Remediation.

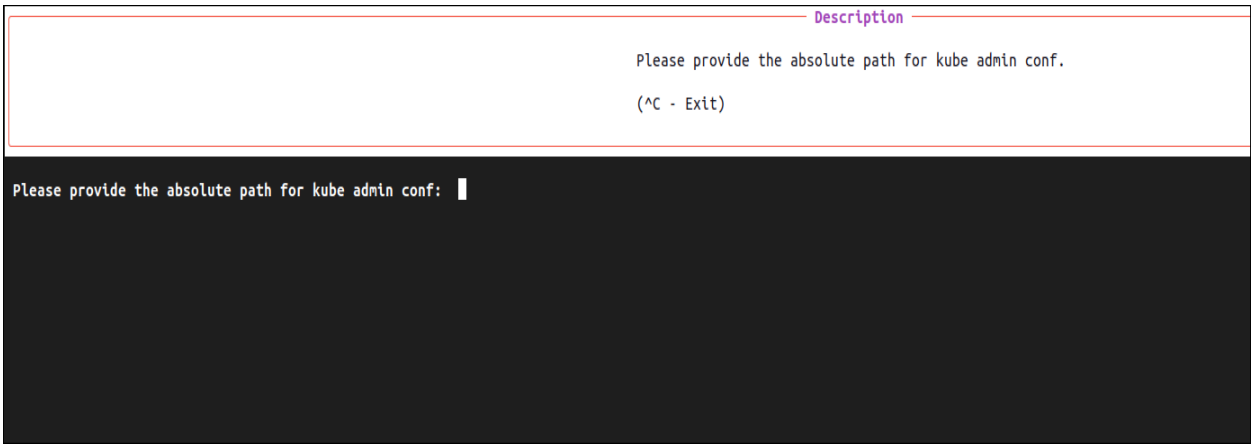
Note: Ensure that the **appviewx_kubernetes_addons_2024.3.0.0.tar.gz** file is present in the **appviewx_kubernetes** directory.



By default, the cluster has read-only access. To perform administrative tasks, admin access must be enabled. A configuration file located at <Home Directory>/.kube/backup/config provides the required admin privileges. When prompted, ensure you specify the correct path to this file.

Note: If the file has been moved or backed up to a secure location, provide the updated file path accordingly.

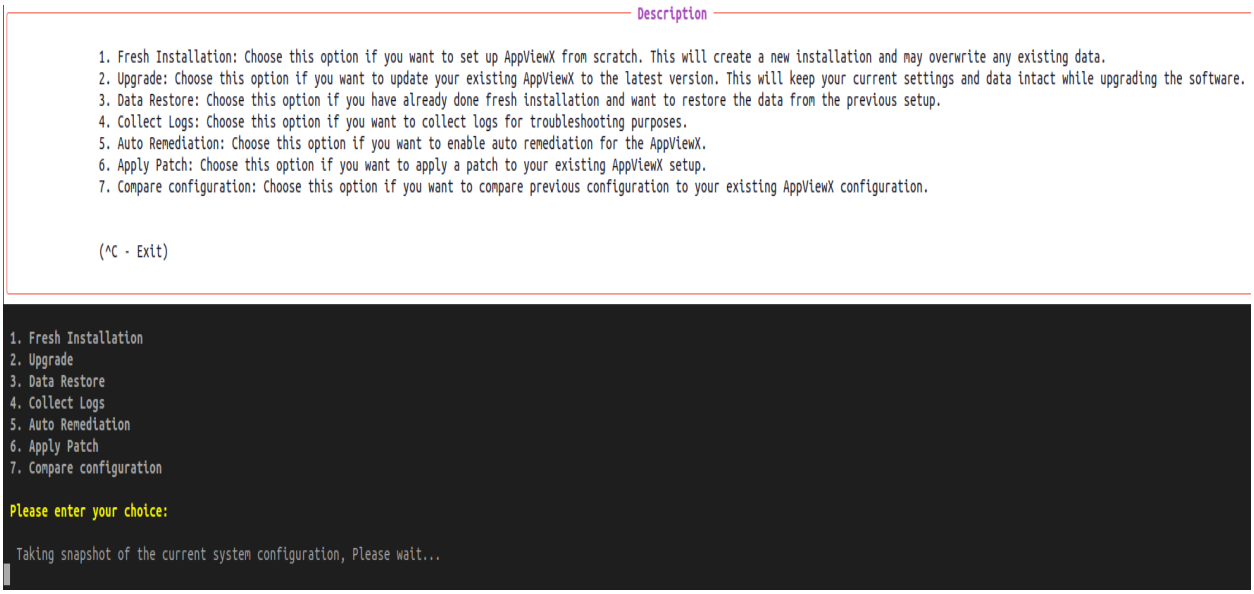
Copyright © 2025 AppViewX Inc. All Rights Reserved.



Apply Patch

This option should be selected when the customer is ready to apply the latest plugins and addons patch, incorporating the newest bug fixes and application enhancements. Choose option 6 for applying the patch.

Note: Ensure that the **appviewx_kubernetes_addons_2024.3.0.0.tar.gz** file is present in the **appviewx_kubernetes** directory.



Copyright © 2025 AppViewX Inc. All Rights Reserved.

Skipping Backup During apply patch:

Attention: Ensure that node snapshots are taken, as the rollback option in the Terminal UI will not restore data if the backup is skipped.

To skip the mongoDB backup:

1. Run `./install.sh`.
2. Select option 6.
3. Enter the password and respond to the prompts.
4. When the table screen appears, create a duplicate session and follow these steps:
   a. Navigate to the directory <appviewx_installer_directory>/appviewx_kubernetes/scripts/
   b. Open the **application_upgrade.json** file.
   c. To disable backups, find the **take_backup** parameter and set its value from `true` to `false`.
   d. Save the file.

Compare Configuration

In customer environments, configuration changes often lead to issues, making it challenging to identify what has changed. This troubleshooting process can also be time-consuming. To streamline this and save time, AppViewX has developed a utility that compares the current configuration with previous versions when the system was functioning correctly. By default, the utility automatically captures snapshots whenever an event—such as an installation, upgrade, patch, or data restore—is successfully completed.

Important: Admin conf is required for running this utility.

Note: Ensure that the **appviewx_kubernetes_addons_2024.3.0.0.tar.gz** file is present in the **appviewx_kubernetes** directory.

Steps to perform a compare configuration:

1. To start the installation process, run the command:

   ```
   ./install.sh
   ```

   The following welcome screen is displayed.

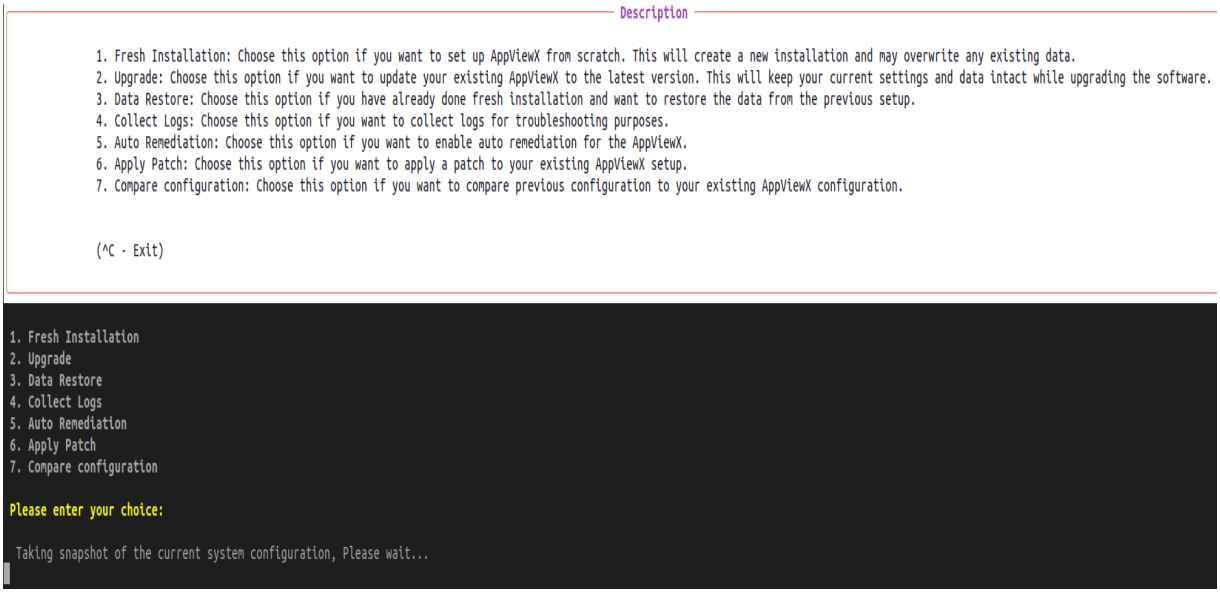
Copyright © 2025 AppViewX Inc. All Rights Reserved.



2. Select **Continue**.

   The screen shows the following options:

   - 1. Fresh Installation
   - 2. Upgrade
   - 3. Data Restore
   - 4. Collect Logs
   - 5. Auto Remediation
   - 6. Apply Patch
   - 7. Compare Configuration

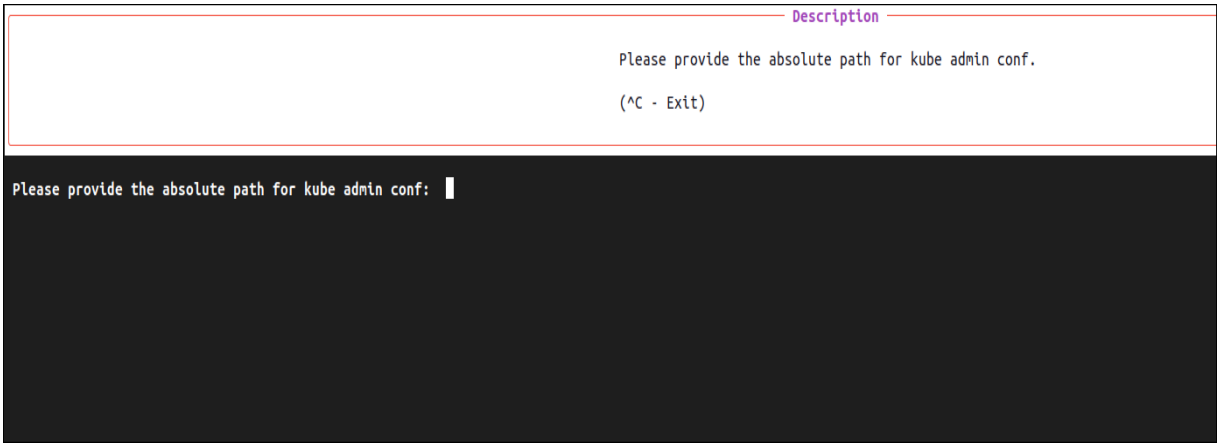


3. Choose option 7, Compare Configuration.

   By default, the cluster has read-only access. To perform administrative tasks, admin access must be enabled. A configuration file located at <Home Directory>/.kube/backup/config provides the required admin privileges. When prompted, ensure you specify the correct path to this file.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Note: If the file has been moved or backed up to a secure location, provide the updated file path accordingly.

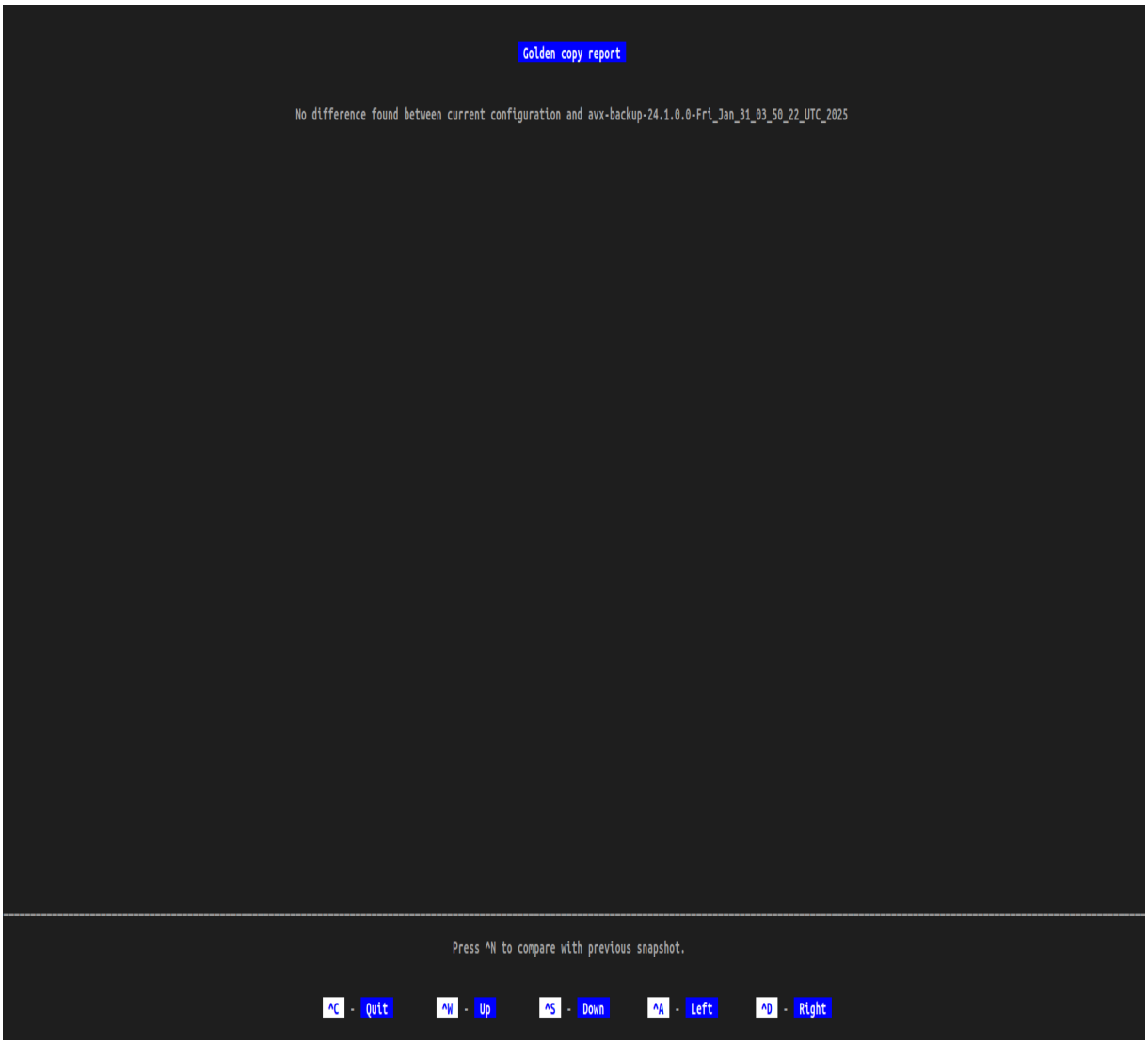


4. The utility captures a snapshot of the current configuration and compares it with the previous one. The resulting snapshots highlight areas where changes have occurred and where no modifications have been made.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



Copyright © 2025 AppViewX Inc. All Rights Reserved.



5. To compare with the next snapshot, press **Ctrl+N**. This will compare it with the subsequent snapshot. If the report is too large, use the following short cut keys:
   - To scroll up, Ctrl+W
   - To scroll down, Ctrl+S
   - To scroll left, Ctrl+A
   - To scroll right, Ctrl+D

Copyright © 2025 AppViewX Inc. All Rights Reserved.

The resulting snapshots highlight areas when there are no more snapshots to compare and when the report is too large.

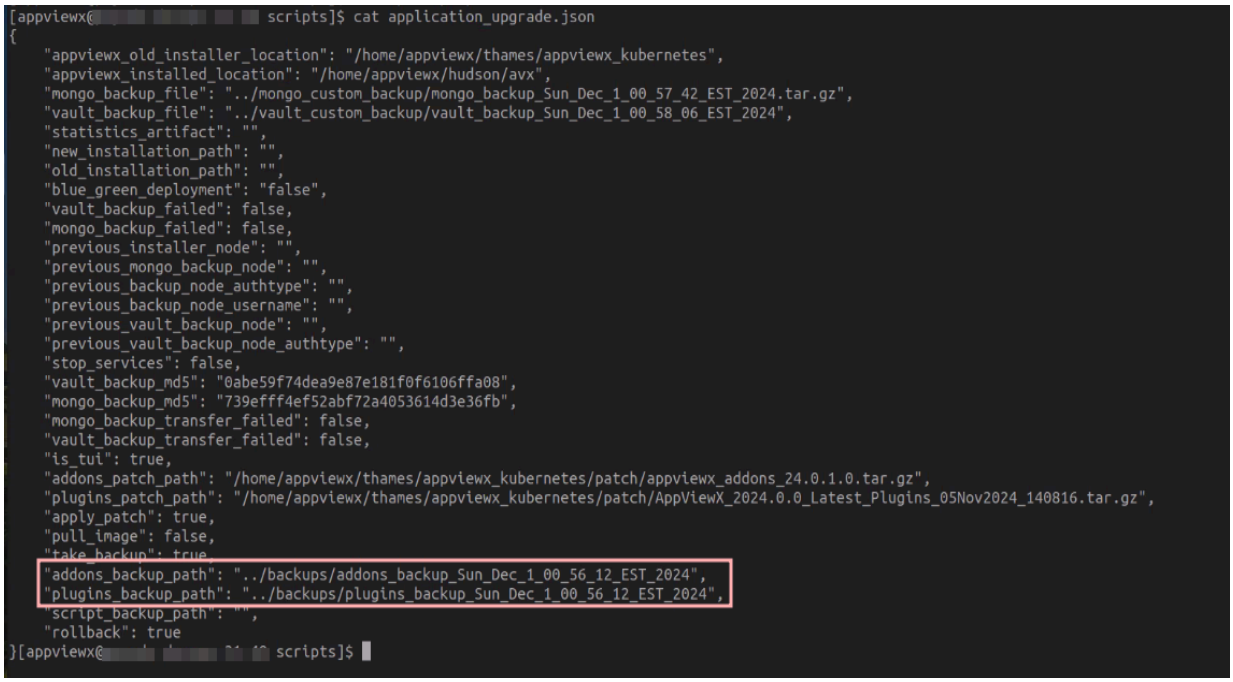


Points to Remember

- Do not use the appviewx_kubernetes installer if it was previously used during the upgrade process. During the upgrade, the initial seed dump file is replaced with the user-provided backup file, which could lead to issues if the same installer is reused for a fresh installation. To ensure a smooth process, always use a new appviewx_kubernetes installer for fresh installations.
- The file <installer path>/interactive-iu/resources/requirements.txt located inside the **interactive-iu** folder is just a reference for prerequisites validation during the upgrade. Do not use it for package installation.

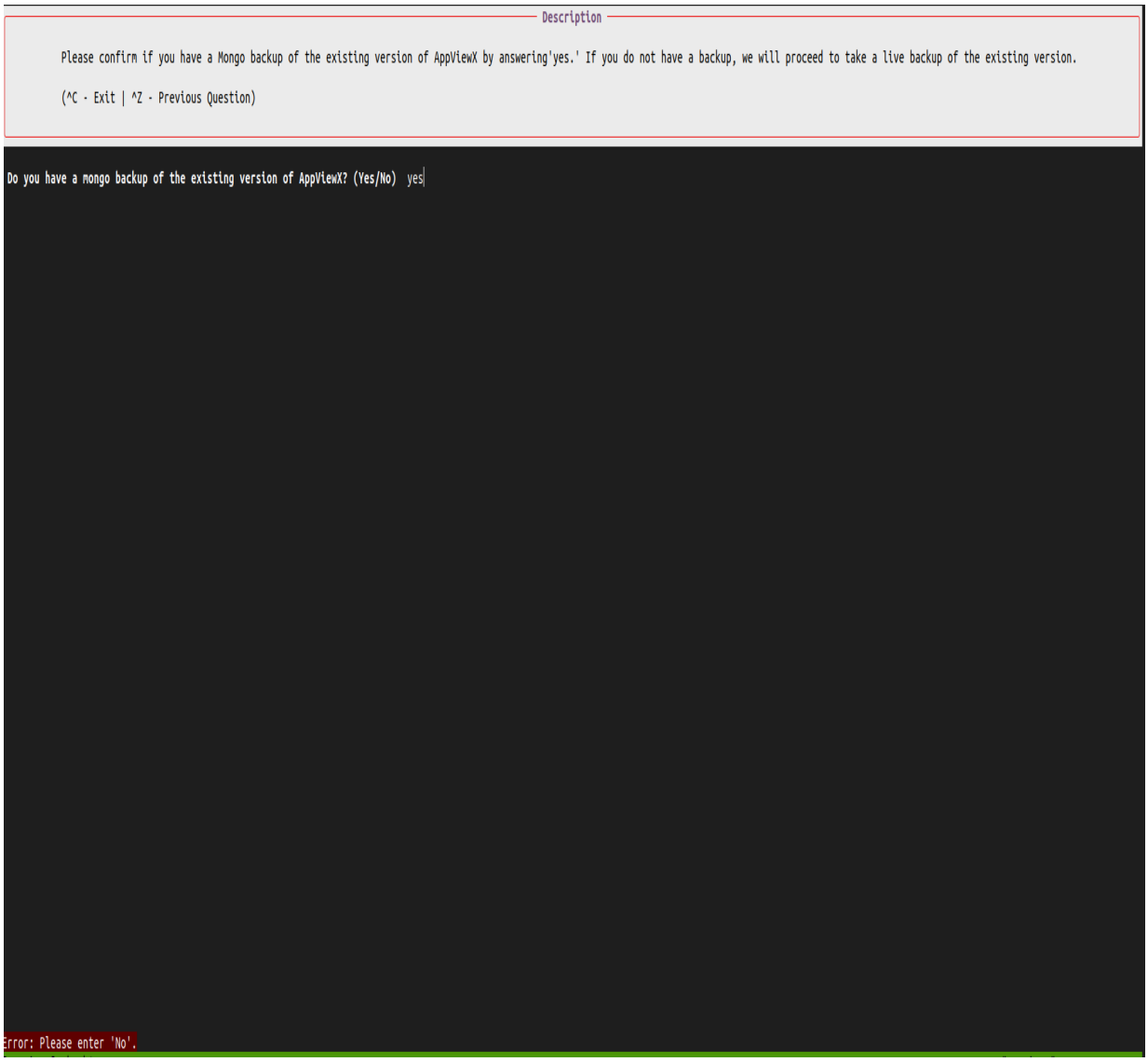
Copyright © 2025 AppViewX Inc. All Rights Reserved.

- If auto-enrolment plugins are selected during the installation phase, the external gateway must be installed manually.
- Navigating back to previous questions after reaching the additional questions requires exiting and restarting the process.
- Latency warnings during prerequisite checks can be ignored. However, it is recommended to manually verify latency using the `ping` command for confirmation.
- For a patch rollback, update the `addons_backup_path` and `plugins_backup_path` in the `application_upgrade.json` file located in the `<appviewx_installer>/scripts` folder if multiple backups are present. Backup paths can be found in the `<appviewx_installer>/backups` directory.



Copyright © 2025 AppViewX Inc. All Rights Reserved.

- Additional questions have been introduced during the production cut-over; however, they are consistently restricted to "No," ensuring they do not impact the cut-over process.



Monitoring and Maintaining AppViewX

A system monitor is a component that is used to gauge resources and performance. It enables users to gather data and manage the health of the system. Although monitoring does not fix issues, it ensures that the system is stable and reliable.

In the case of AppViewX, during installation, the ELK stack is installed. This stack is a combination of three open-source products; Elasticsearch, Logstash, and Kibana. These are used to analyse the log files and ensure a stable system performance.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- Installing ELK Components
- Executing Commands for Maintenance
- Installing Trusted Certificate for GUI/API Access
- Enabling Strict Data Center Routing
- Enabling the Load Balancer for the Kube API Server
- Enabling Device Syslog Processing
- Enabling the Insight Module
- Understanding Commands Executed during Installation
- Enabling Sudo Access
- Graceful Reboot Procedure
- Adding a Node to the Cluster
- Working with Alerts
- Working with Backup and Restore
- Working with Logs
- Working with Plugins
- Working with the Management Console
- Applying Custom Pod Configurations
- Remote Transfer of AppViewX Backups

Installing ELK Components

Elasticsearch, Logstash, and Kibana (ELK) provide centralized logging to identify problems in servers or applications. It allows you to search all the logs and find issues that occur in multiple servers by connecting their logs during a specific time frame. To enable log management of the present and historical CLI logs of AppViewX from a GUI, an ELK-based utility is utilized.

To install ELK:

1. Download the **appviewx_kubernetes_elk_2024.3.0.0.tar.gz** file.
2. To extract the contents of the file, execute the following command:

   ```
   tar -xvf appviewx_kubernetes_elk_2024.3.0.0.tar.gz
   ```

3. Move the contents of the extracted file to <InstallerLocation>/appviewx_kubernetes/yaml/appviewx_monitoring directory using the command,

   ```
   cp -r appviewx_monitoring/* <InstallerLocation>/appviewx_kubernetes/yaml/appviewx_monitoring
   ```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

4. Edit the **appviewx.conf** file.
5. Update the following parameters with their corresponding values:
   a. **ELK=TRUE**
   b. /appviewx_kubernetes/yaml/appviewx_monitoring directory
6. Navigate to <InstallerLocation>/appviewx_kubernetes/scripts/ directory.
7. To run the script, execute the following command:

```
./elk_install.sh
```

You can find the URLs to access the ELK stack in the command prompt after the installation is complete.

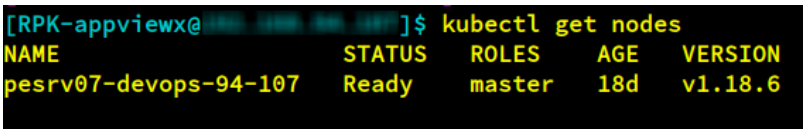


Executing Commands for Maintenance

AppViewX is installed based on the Kubernetes engine. We can use a few of the basic Kubernetes commands to manage the components in AppViewX. The list below contains the basic commands to manage the Kubernetes cluster.

1. View all the nodes in the cluster

```
kubectl get nodes
```



2. View all the pods of AppViewX services

```
kubectl get pods -n avx -o wide
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
[RPK-appviewx@ ]$ kubectl get pods -n avx -o wide
NAME                                    READY   STATUS              RESTARTS   AGE
IP              NODE                    NOMINATED NODE   READINESS GATES
avx-platform-gateway-586cdccd79-s7fl9   0/2     Pending             0          14d
<none>          <none>                  <none>           <none>
avx-platform-gateway-586cdccd79-xlxcd   1/2     Terminating         793        18d
<none>          pesrv07-devops-94-107   <none>           <none>
avx-platform-web-5c4595b87-cd2ml        2/2     Running             0          12d
                pesrv07-devops-94-107   <none>           <none>
mongo-configdb-0                        2/2     Running             0          12d
                pesrv07-devops-94-107   <none>           <none>
mongo-configdb-1                        2/2     Running             0          12d
                pesrv07-devops-94-107   <none>           <none>
mongo-configdb-2                        2/2     Running             0          12d
                pesrv07-devops-94-107   <none>           <none>
```

3. View all the services

```
kubectl get services -n avx
```

```
[RPK-appviewx@ ]$ kubectl get services -n avx
NAME                    TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)              AGE
avx-platform-gateway    ClusterIP                    <none>        5300/TCP             18d
avx-platform-web        ClusterIP                    <none>        5004/TCP,5555/TCP    18d
mongo-configdb-service  ClusterIP                    <none>        27017/TCP            18d
mongo-routerdb-service  ClusterIP                    <none>        27017/TCP            18d
mongo-shareddb-service  ClusterIP                    <none>        27017/TCP            18d
vault                   ClusterIP                    <none>        8200/TCP,8201/TCP    18d
vault-internal          ClusterIP                    <none>        8200/TCP,8201/TCP    18d
```

4. Log in to a particular container of the pod

```
kubectl exec -it avx-platform-web-5c4595b87-cd2ml -n avx /bin/sh
```

```
[RPK-appviewx@ ]$ kubectl exec -it avx-platform-web-5c4595b87-cd2ml -n avx -- /bin/sh
Defaulting container name to avx-platform-web.
Use 'kubectl describe pod/avx-platform-web-5c4595b87-cd2ml -n avx' to see all of the containers in this pod.
sh-4.2#
sh-4.2#
```

5. List all the namespaces

```
kubectl get namespaces
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
[RPK-appviewx@ ]$ kubectl get namespaces
NAME                   STATUS   AGE
absecon                Active   18d
avx                    Active   18d
avx-jobs               Active   18d
default                Active   18d
external-system        Active   18d
istio-operator         Active   18d
istio-system           Active   18d
kube-node-lease        Active   18d
kube-public            Active   18d
kube-system            Active   18d
kubernetes-dashboard   Active   18d
lens-metrics           Active   16d
```

6. List all the configuration maps. This is used to view configuration related details.

```
kubectl get configmaps -n avx
```

```
[RPK-appviewx@ ]$ kubectl get configmaps -n avx
NAME                           DATA   AGE
avx-common-config              6      18d
avx-platform-gateway-config    2      18d
avx-platform-web-config        1      18d
avx-vault-configmap            3      18d
istio-ca-root-cert             1      18d
vault-config                   1      18d
```

7. List all the deployments

```
kubectl get deployments -n avx
```

```
[RPK-appviewx@ ]$ kubectl get deployments -n avx
NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
avx-platform-gateway  0/1     1            0           18d
avx-platform-web      1/1     1            1           18d
```

8. Stop a deployment

```
kubectl scale --replicas=0 deployment/avx-vendor-haproxy -n avx
```

```
[RPK-appviewx@ ]$ kubectl scale --replicas=0 deployment/avx-platform-gateway -n avx
deployment.apps/avx-platform-gateway scaled
```

9. Start a deployment

```
kubectl scale --replicas=1 deployment/avx-vendor-haproxy -n avx
```

```
[RPK-appviewx@ ]$ kubectl scale --replicas=1 deployment/avx-platform-gateway -n avx
deployment.apps/avx-platform-gateway scaled
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

10. Edit a configuration

```
kubectl edit configmaps -n avx avx-common-config
```

```
# Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this file will be
# reopened with the relevant failures.
#
apiVersion: v1
data:
  APS_MONGO_ENCRYPTED_PASSWORD: wnfzZa0MAvfOR/ULgeCMNA==
  DATA_CENTER: avx
  DEPENDENCY_PATH: /appviewx/dependencies
  MONGO_ENCRYPTED_PASSWORD: vault:v1:JY4O+YyoCLfcfUys7T84zWGAB/Vr9sNSk/8h9VYFNIA+jazThhggPeH49ZM5
  MONGO_KEY: t6ehrofmwa59g3hoakjh4d79s
  appviewx.properties: "#Below Vault are replaced in vault helm chart\nAPP_ROLE_ID=a5e54859-3304-13b3-3d74-6
\n#RELEASE_INFO\nRELEASE_DATE=2019-18-12_17-24-00\nBUILD_NUMBER=416\nRELEASE_DESCRIPTION=appviewX2020.1.0\n\
alhost:$APPVIEWX_SERVICE_PORT/services/\n\n#CERT_DELAY\nCERT_DISC_BATCH_AND_DELAY_IN_MILLISECONDS=220/2000\r
```

11. Describe the pods

```
kubectl describe pods -n avx <plugin name>
```

```
[RPK-appviewx@ ]$ kubectl describe pods -n avx avx-platform-web-5c4595b
87-cd2ml
Name:         avx-platform-web-5c4595b87-cd2ml
Namespace:    avx
Priority:     0
Node:
Start Time:   Thu, 25 Feb 2021 04:53:59 +0000
Labels:
Annotations:
```

12. Log in to the database

```
kubectl exec -it mongo-routerdb-0 -n avx -- /bin/sh
```

```
[RPK-appviewx@ ]$ kubectl exec -it mongo-routerdb-0 -n avx -- /bin/sh
Defaulting container name to mongo-routerdb-container.
Use 'kubectl describe pod/mongo-routerdb-0 -n avx' to see all of the containers in this pod.
#
#
```

Installing Trusted Certificate for GUI/API Access

The steps to install a trusted certificate for GUI/API access is follows:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

1. To create a secret external-tls-credential of type tls, execute the following command:

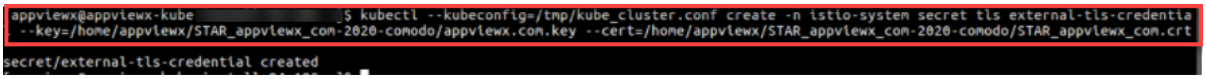
```
kubectl --kubeconfig=~/.kube/config create -n istio-system secret tls external-tls-credential --key=/etc/qualys/ssl/appviewx.com.key
 --cert=/etc/qualys/ssl/ssl-bundle.crt
```

For example:

```
kubectl --kubeconfig=~/.kube/config create -n istio-system secret tls external-tls-credential --key=/etc/qualys/ssl/appviewx.com.key
 --cert=/etc/qualys/ssl/ssl-bundle.crt
```

where:

- *~/.kube/config* should be present in each node
- *~/.kube* will be present in the home folder of the installing user

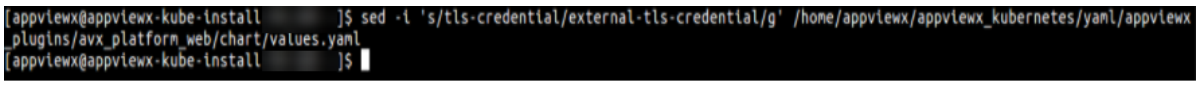


2. Replace secret name tls-credential with external-tls-credential in the `values.yaml` file.

> **Note:** The `values.yaml` file is available at installerLocation/appviewx_kubernetes/yaml/appviewx_plugins/avx_platform_web/chart/

- To replace, execute the following command:

```
sed -i 's/tls-credential/external-tls-credential/g' <installerLocation>/appviewx_kubernetes/yaml/appviewx_plugins/avx_platform_web/chart/values.yaml
```



3. Update the Gateway to consume the latest changes:

a. To navigate to the <installerLocation>/appviewx_kubernetes/yaml/appviewx_plugins/avx_platform_web directory, execute the following command:

```
cd <installerLocation>/appviewx_kubernetes/yaml/appviewx_plugins/avx_platform_web
```

b. To upgrade the avx-platform-web package to reflect changes, execute the following command:

```
helm upgrade avx-platform-web ./chart
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
[appviewx@appviewx-kube-install                                    ]$ helm upgrade avx-platform-web ./chart
Release "avx-platform-web" has been upgraded. Happy Helming!
NAME: avx-platform-web
LAST DEPLOYED: Wed Sep  9 10:49:09 2020
NAMESPACE: default
STATUS: deployed
REVISION: 2
TEST SUITE: None
[appviewx@appviewx-kube-install                                    ]$
```

4. Verify the application URL to check SSL is enabled.
5. Verify the certificate by launching the Appviewx portal.

   The URL is `https://<Service URL>:Port>/appviewx`



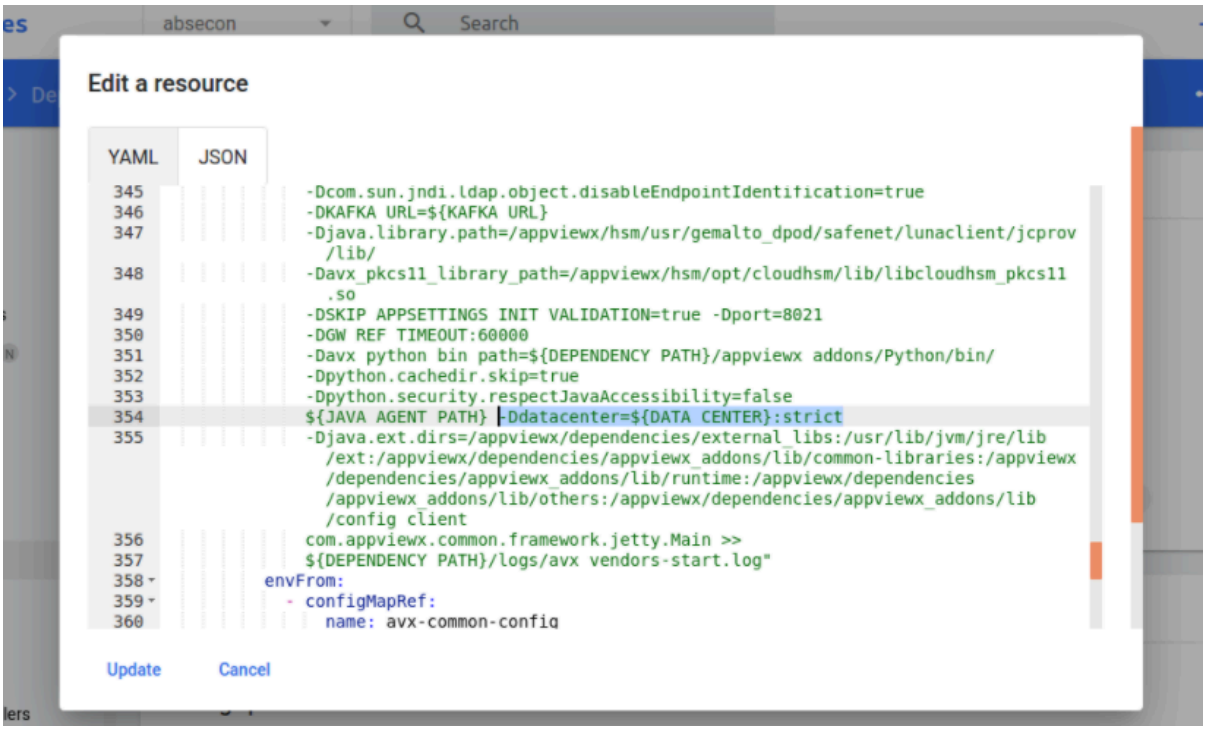
Enabling Strict Data Center Routing

Strict data center routing is used to ensure that calls from AppViewX to a plugin/device through one data center are not routed to any other data center if there are no plugins available to serve traffic in the same data center.

To enable strict data center routing:

1. Log in to the Kubernetes dashboard of AppViewX.
2. On the left pane, under **Workloads**, click **Deployments**.
3. Search for the respective deployment to modify it.
4. Click **Edit**.
5. Add the argument **:strict** in **-Ddatacenter** jvm argument as shown below:

Copyright © 2025 AppViewX Inc. All Rights Reserved.



6. Click **Update**.

Enabling the Load Balancer for the Kube API Server

Given below is an example configuration done on F5 devices and is needed only when we need to balance the load between multiple kube api servers in the case of multi DC support.

Prerequisite:

Create the TCP load balancer for Kube master API Server.

> **Note:** This section is applicable only when the load balancer for the kube apiserver is not installed during the installation.

Sample Configuration:

Load balancer Configuration for Kube Master:

```
ltm virtual vs-appviewxmasterapi {

destination <IP Address>:sun-sr-https

ip-protocol tcp

mask XXX.XXX.XXX.XXX
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
pool pool-avxmasterapi
profiles {
fastL4 { }
}
serverssl-use-sni disabled
source 0.0.0.0/0
source-address-translation {
type automap
}
translate-address enabled
translate-port enabled
}
```

Pool Member Configuration for Kube Master

```
ltm pool pool-avxmasterapi {
 members {
 <Master Node IP Address>:sun-sr-https {
 address XXX.XXX.XXX.XXX
 session monitor-enabled
 state up
 }
 <Master Node IP Address>:sun-sr-https {
 address XXX.XXX.XXX.XXX
 session monitor-enabled
 state up
 }
 <Master Node IP Address>:sun-sr-https {
 address XXX.XXX.XXX.XXX
 session monitor-enabled
 state up
 }
 }
 monitor gateway_icmp
 }
```

To enable the load balancer for Kube Master:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

1. To verify whether the load balancer is functioning normally, execute the `curl` command with endpoint `/healthz` or `/version` as shown below:

```
curl -k https://loadbalancer-ip:6443/healthz
```

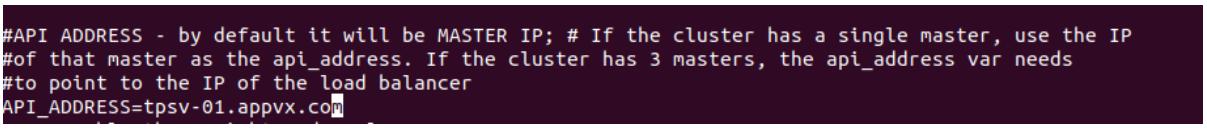
> **Note:** The `/healthz` endpoint should return an "OK" status, indicating that the Load Balancer is functioning properly and the API Server is reachable.

```
curl -k https://loadbalancer-ip:6443/version
```

> **Note:** The `/version` endpoint may require an auhorizaiton. Using `curl` to access endpoints like `/version` without an authentication token will result in a 403 Forbidden error, as unauthenticated requests are not permitted.



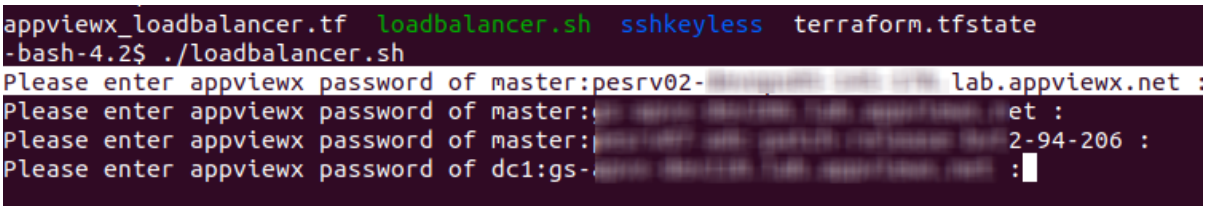
2. Apply the latest script patch from the release portal.
3. Navigate to the <installerLocation>/appviewx_kubernetes/scripts/ directory.
4. Open the **appviewx.conf** file.
5. Search for the **API_ADDRESS** parameter.
6. Modify the value of the **API_ADDRESS** parameter to reflect the IP Address or the FQDN of the load balancer.



7. Navigate to the <installerLocation>/appviewx_kubernetes/scripts/loadbalancer/ directory.
8. To run the load balancer script, execute the following command:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
./loadbalancer.sh
```



9. Enter the password of the nodes when prompted.
10. To verify the changes, execute the following command:

```
kubectl cluster-info
```

The output should contain the updated load balancer URL (IP Address or FQDN) of the kube API server.

Note: Once the LoadBalancer has been verified, proceed to delete the backup of the older control plane certificates using the following command:

```
rm -rf <actual_directory_path>/kubernetes_API_LB_backup_*
```

Replace `<actual_directory_path>` with the actual directory path.

Rollback Master LB

1. Modify the following parameters in the appviewx.conf file with the values as specified.
   - API_ADDRESS = <IP of the Master Node>
   - API_ADDRESS_LISTNER_PORT = 6443
2. Run the load balancer script, using the following command:

```
./loadbalancer.sh
```

3. Once the script completes successfully, execute the command below to confirm that the cluster is now pointing to the master IP.

```
kubectl cluster-info
```

Enabling Device Syslog Processing

The Syslog module in AppViewX is used to receive syslogs from the device and update the necessary changes made in the device into the AppViewX database.

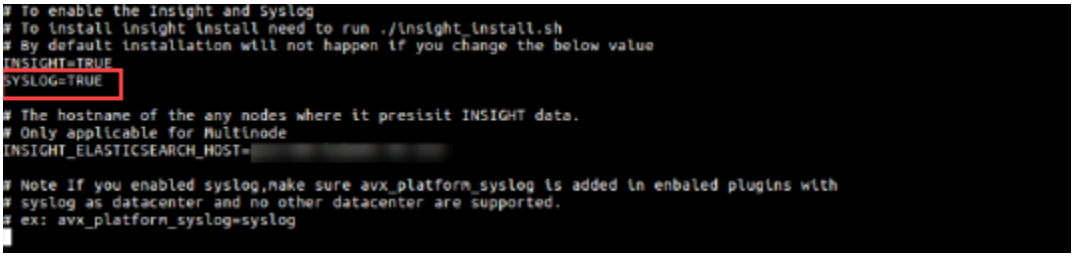
Copyright © 2025 AppViewX Inc. All Rights Reserved.

To enable Syslog parsing for the devices managed by AppViewX:

1. Navigate to the /home/appviewx/appviewx_kubernetes/scripts directory.
2. To open the **appviewx.conf** file, execute the following command:

```
vi appviewx.conf
```

3. Search for the SYSLOG parameter.
4. Set the value of the SYSLOG parameter to TRUE.



5. Search for **Enabled Plugins**.
6. Add the following plugins:
   - **appviewx_dependencies**
   - **avx_platform_syslog**
   - **avx_platform_gateway**

> **Note:** Gateway must be added to register the new APIs from the plugins that are installed.

7. Update the data center as **syslog** for the parameter **avx_platform_syslog** plugin.



8. Save and exit the **appviewx.conf** file.
9. From the /home/appviewx/appviewx_kubernetes/scripts directory, execute the following command:

```
./insight_install.sh
```

10. Execute the following command:

```
./plugins_install.sh
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

11. Execute the following command:

```
kubectl get services -n syslog
```

It displays the results as shown in the image below. Fetch the Syslog port from the service logstash-syslog-service. Here, the Syslog port is 30336.



This Syslog port changes for every installation/upgrade.

12. Connect to the MongoDB and open the **avx_app_metadata** collections. Edit this file by searching the parameter SYSLOG_RECEIVER_ENABLED and set it to TRUE. Save the file and move out of the DB.
13. To configure Syslog as TRUE, execute the following command:

```
kubectl edit configmaps -n "data center name" Set SYSLOG_RECEIVER_ENABLED=True,SYSLOG_HOST=192.168.XXX.XXX (Node IP where Syslog is installed)),SYSLOG_PORT=30047 (fetch the ports from point 8)
```

14. Save and exit the **<configmaps>** file.
15. To get the Pod name, execute the following command:

```
kubectl get pods -n "data center name"
```

16. To restart subsystems and vendors, execute the following command:

```
kubectl delete pods “Pod name” -n ‘‘data center name’’
```

For example: You may restart multiple pods and the config servers by entering the name of the pod and the config server in the command below with space.

```
kubectl delete pods avx-subsystems-7666cfb459-6q4rn avx-vendors-99c69cd69-jtr4w avx-config-server-85ff9dd46d-h5qnr-n ''data center name’’
```

Enabling the Insight Module

The Insight module allows you to collect statistics from the devices that are managed by AppViewX. Also, it displays historical statistics on demand for users.

To install Insight for statistics collection:

1. Open the terminal.
2. Navigate to the /home/appviewx/appviewx_kubernetes/yaml directory.
3. Download the **appviewx_kubernetes_insight_2024.3.0.0.tar.gz** file.
4. To extract the file, execute the following command:

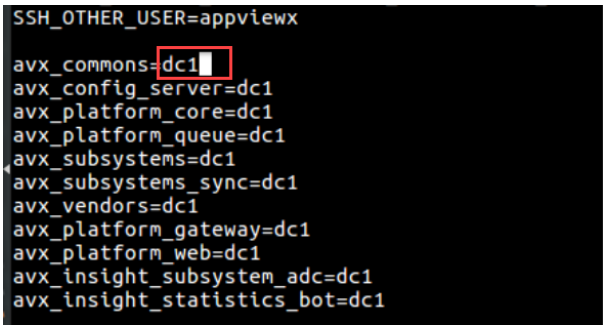
```
tar -xvf appviewx_kubernetes_insight_2024.3.0.0.tar.gz
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

5. Navigate to the /home/appviewx/appviewx_kubernetes/scripts directory.
6. To open the **appviewx.conf** file in the editor mode, execute the following command:

```
vi appviewx.conf
```

7. Set the following parameters as follows:
   - INSIGHT = TRUE
   - ELASTICSEARCH_BACKUP_HOST = <node1-hostname>,<node2-hostname>
   - ELASTIC_BACKUP_PATH = <path to store backup e.g.: /home/appviewx/elastic_backup>
8. Search for **Enabled Plugins** and add the following plugins:
   - **appviewx_dependencies**
   - **avx_insight_subsystem_adc**
   - **avx_insight_statistics_bot**
   - **avx_platform_gateway**
9. Update the data center for insight plugins as shown in the image below:



10. Save and exit the **appviewx.conf** file.
11. To install Insight, navigate to the /home/appviewx/appviewx_kubernetes/scripts directory.
12. Execute the following command:

```
./insight_install.sh
```

13. Execute the following command:

```
./plugins_install.sh
```

14. To restart subsystems and vendors, execute the following command:

```
kubectl delete pods "Pod name" -n "datacenter name"
```

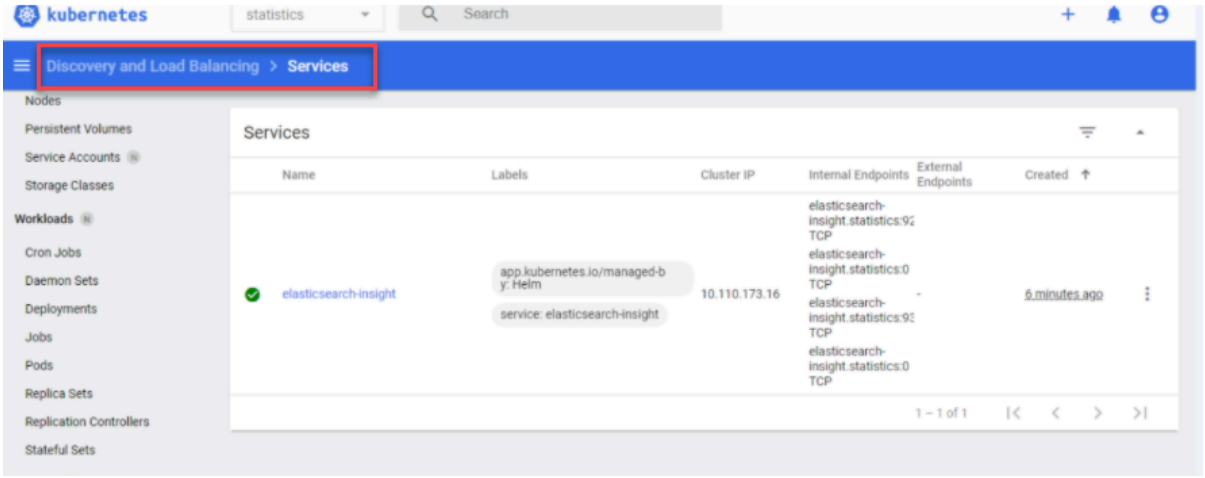
    For example:

```
kubectl delete pods avx-insight-statsitics-bot-3499c69cd6-4sdfs,avx-insight-subsystem-adc-4399c69ed6-4sdfs,avx-subsystems-7666cfb459-6q4rn -n
absecon
```

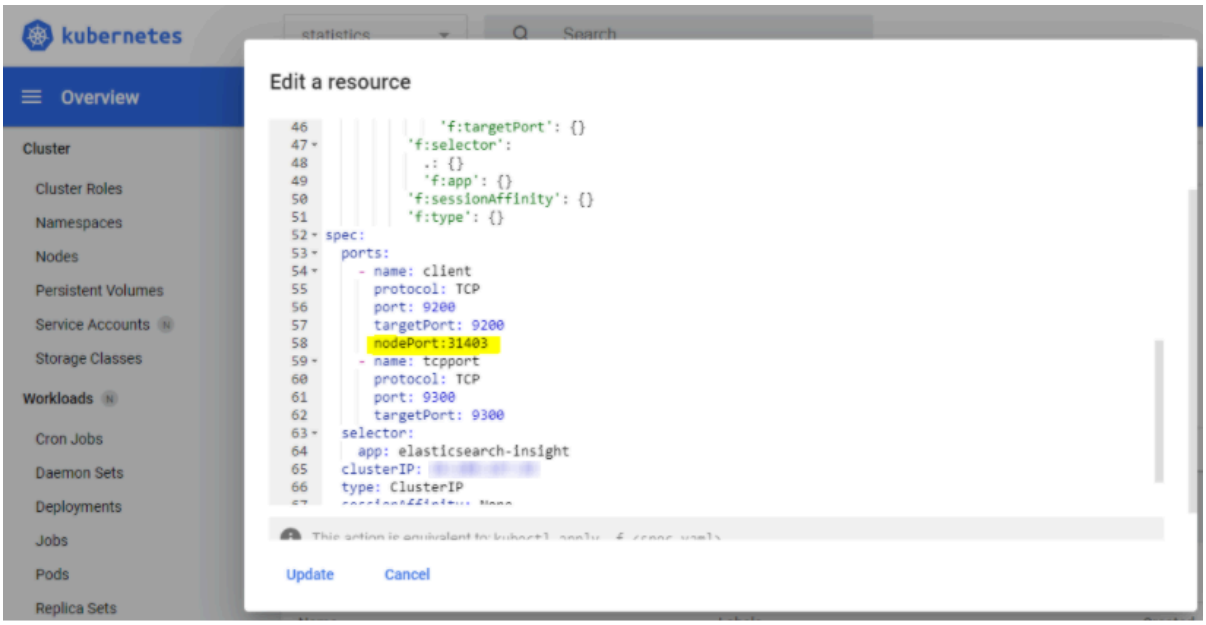
    To restart multiple Pods, enter the name of the pod in the above command with space.
15. In the case of Insight migration, continue till point 11.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

16. Log in to the Kubernetes dashboard, enter statistics as the namespace.
17. Select services and search for **elasticsearch-insight**.

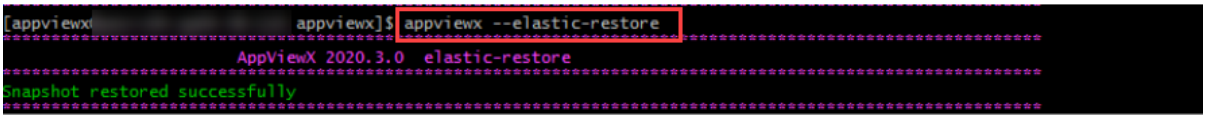


18. On the Pod, click **Edit**.
19. Enter the port details as **nodePort: 31403** and save as shown in the image below:



20. To restore the elastic data, go to the old installation path of AppViewX, and execute the following command:

```
appviewx --elastic-restore
```

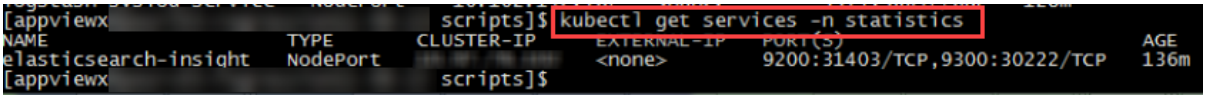


Copyright © 2025 AppViewX Inc. All Rights Reserved.

21. To connect to the elastic database, execute the following command:

```
kubectl get services -n statistics
```

It displays the results as shown in the image below:



Understanding Commands Executed during Installation

The section lists the commands executed by the AppViewX installer that requires Sudo access.

To restrict the commands that a Sudo user has access to, enable the following commands:

- `sudo kubeadm`
- `sudo kubectl`
- `sudo yum remove`
- `sudo yum install`
- `sudo systemctl daemon-reload`
- `sudo rpm -ivh --force *.rpm`
- `sudo modprobe br_netfilter`
- `sudo swapoff -a`
- `sudo iptables -F`
- `sudo date --set`
- `sudo -S setenforce 0`
- `sudo -S sysctl -w net.bridge.bridge-nf-call-iptables=1`

Apart from the above commands, Sudo user must be able to read/write/execute in the following directories:

- `/etc/`
- `/root/`
- `/var/lib`
- `/tmp`
- `/usr/local/bin`
- `/home/SSH_OTHER_USER` (Other user is user-defined in /scripts/appviewx.conf)

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Enabling Sudo Access

To enable Sudo access and grant access to all commands:

1. Log in as an Administrator.
2. Log in to the node with root credentials.

• Creating a New Sudo User

• Adding Users to the Sudo Group

• Verifying if the Wheel Group is Enabled

• Adding a User to the Wheel Group

• Switching to the Sudo User

• Sudo Privilege Access for RHEL OS

Creating a New Sudo User

1. Open the terminal.
2. Execute the following command:

```
adduser <UserName>
```

> **Note:** Replace the UserName with the new user's name.

3. To create a password for the new user, execute the following command:

```
passwd <Password>
```

The system prompts you to set and confirm a password for your new user account. If successful, the system responds with “all authentication tokens updated successfully.”

> **Note:** A strong secure password has more characters and a few special characters (such as numbers, symbols, or capitals). Ensure that you are choosing an appropriately strong password for your system.

Adding Users to the Sudo Group

For CentOS

Copyright © 2025 AppViewX Inc. All Rights Reserved.

By default, CentOS 7 has a user group called the Wheel group. Members of the wheel group are automatically granted Sudo privileges. Adding a user to this group grants Sudo privileges to the user.

To add a user to the wheel group, refer to Adding a User to the Wheel Group.

For Ubuntu

In Ubuntu OS, a user in “sudo” group is granted sudo permissions.

To add a user to the sudo group:

Execute the following command:

```
usermod –aG scudo UserName
```

Note: Replace the UserName with the new user's name to grant Sudo privileges.

Verifying if the Wheel Group is Enabled

To verify whether CentOS 7 installation has the wheel group enabled or disabled:

1. To open the configuration file, execute the following command:

   ```
   vi sudo
   ```

2. Search for the following entry in the configuration file:

   ```
   ## Allows people in group wheel to run all commands %wheel ALL=(ALL) AL
   ```

   If the second line begins with the # sign, it indicates that the line is marked as a comment and the feature is disabled.

3. Delete the # sign at the beginning of the second line as given below.

   ```
   %wheel ALL=(ALL) ALL
   ```

4. Save the file and exit the editor.

Note: If there is no # sign at the beginning of the line, do not make any changes. The wheel group is already enabled.

Adding a User to the Wheel Group

Adding a user to the wheel group is applicable for CentOS.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

To add a user to the wheel group:

Execute the following command:

```
usermod –aG wheel UserName
```

Note: Replace the UserName with the new user's name to grant Sudo privileges.

Switching to the Sudo User

To switch to the new (or newly-elevated) user account with the su (substitute user):

1. Execute the following command:

   ```
   su - UserName
   ```

2. Enter the password if prompted.
3. To list the contents of the /root directory, execute the following command:

   ```
   sudo ls -la /root
   ```

4. Enter the password if prompted.
   The terminal displays the list of directories. Since listing the contents of the /root directory requires Sudo privileges, this is an easy way to prove that the new user can use the Sudo command.

Sudo Privilege Access for RHEL OS

AppViewX is installed on a Kubernetes engine. To install the underlying Kubernetes engine and other required dependencies, such as containerd, sudo access and executable permissions for the **/tmp** mount point are required.

Adding Commands in the Sudoers List

Below are the commands that must be added to the sudoers list to perform essential tasks such as installation, upgrade, Master Load Balancer configuration, prerequisite validation, and uninstallation of AppViewX. These commands are mandatory for any upgrade activities, and no alternative methods are supported.

Note: The package manager commands may vary depending on whether you are using RHEL/ CentOS or Ubuntu. Please add them to the sudoers list accordingly.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

S NO	Commands to add in sudoers list
1	mkdir
2	cp
3	chmod
4	sed
5	systemctl
6	rm
7	umask
8	*kubeadm (path: /usr/bin/kubeadm)
9	chown
10	yum
11	echo
12	modprobe
13	sysctl
14	rpm
15	swapoff
16	setenforce
17	bash
18	mv
19	tee
20	whoami
21	mkfs.xfs
22	grep
23	mount
24	umount
25	sleep
26	dd

Copyright © 2025 AppViewX Inc. All Rights Reserved.

S NO	Commands to add in sudoers list
27	tcpdump
28	cat
29	sh
30	*crictl (path: /usr/bin/crictl)
31	timedatectl
32	hwclock
33	ntpq
34	chronyc
35	ifconfig
36	ip
37	nc
38	sshd
39	*istioctl (path: <INSTALLER_PATH>/bin/istioctl)
40	*calicoctl (path: <INSTALLER_PATH>/bin/calicoctl)
The commands with the asterix (*) do not require separate installation but should be included in the sudo privileges list. These packages are included as part of the installer. (Installation path: up to appviewx_kubernetes, e.g., /home/appviewx/appviewx_kubernetes).	

Apart from the above commands, Sudo user must be able to read/write/execute in the following directories:

- /etc/ (read/write/)
- /var/ (read/write/)
- /tmp/ (read/write/execute)
- /usr/ (read/write/)
- /home/SSH_OTHER_USER (Other user is user-defined in /scripts/appviewx.conf)

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Graceful Reboot Procedure

A graceful reboot ensures that AppViewX services and Kubernetes workloads remain stable and recover seamlessly after system restarts. This procedure describes the steps to safely stop services, reboot nodes, and verify that all components return to a healthy and balanced state across the cluster.

Operating system updates (such as using `apt update`) can be performed without stopping application services. However, before initiating a server reboot, follow the sequence below:

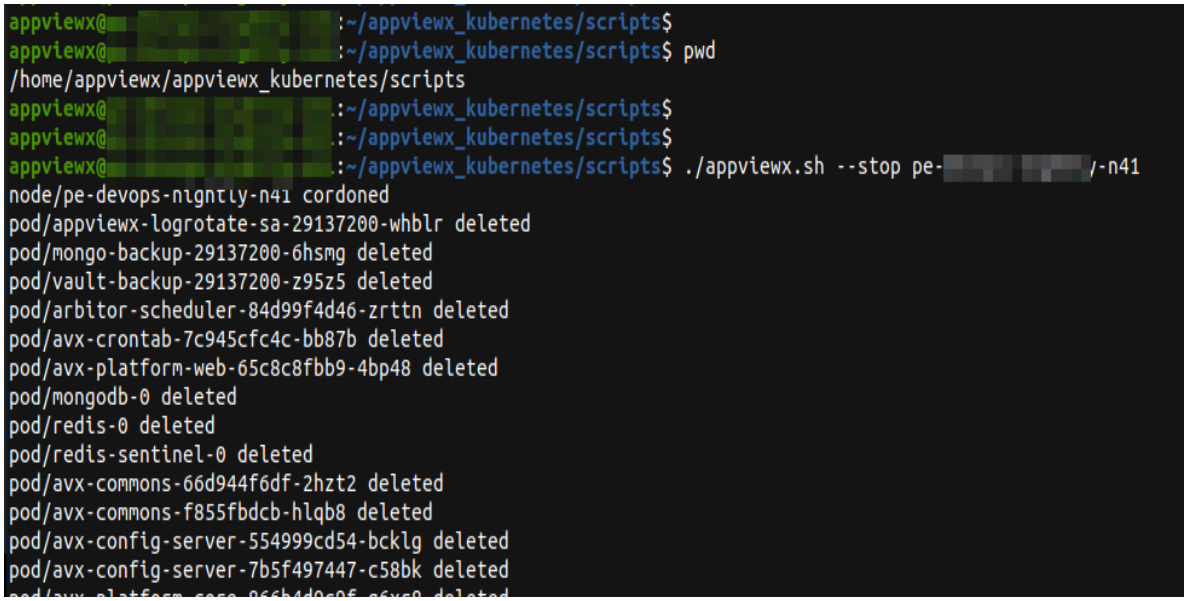
1. Log in to the AppViewX worker node where the installation was originally initiated.
2. Go to the **scripts** directory inside the installer directory.

```
cd <installer_directory_path>/appviewx_kubernetes/scripts
```

3. If the application is deployed in a single node setup:

   a. Stop the application services running on the server.

```
./appviewx.sh --stop <nodename>
```



   b. Once the components are stopped, initiate the reboot.

```
reboot
```

   c. Start the application services running on the server.

```
./appviewx.sh –start <nodename>
```

4. If the application is deployed in a multinode setup for high availability, there are two approaches:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

a. Stop and restart all cluster services; this process involves minimal downtime.

i. Stop all application services.

```
./appviewx.sh --stop -all
```

ii. Reboot all servers in the cluster.

```
reboot
```

iii. Start all application services.

```
./appviewx.sh --start -all
```

b. Stop and restart services sequentially on each node to avoid downtime in a failover setup.

Note:

- No downtime applies only to multi-node clusters (Workers and Masters). For multi-master setups, ensure that the master nodes are load-balanced to prevent downtime.
- Run the following command sequence from the *scripts* directory on the installer node, starting with the worker nodes, followed by the master nodes; one server at a time.

i. Stop the application services.

```
./appviewx.sh --stop <nodename>
```

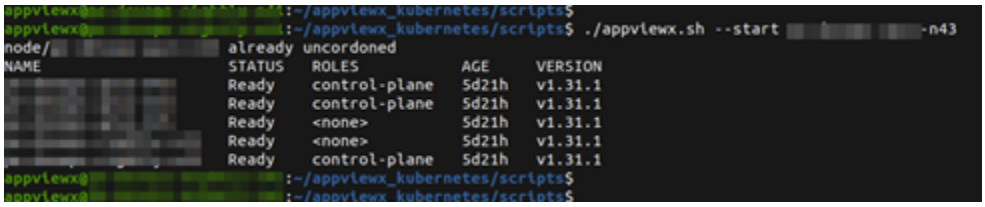
ii. Reboot the server.

```
reboot
```

iii. Start all application services.

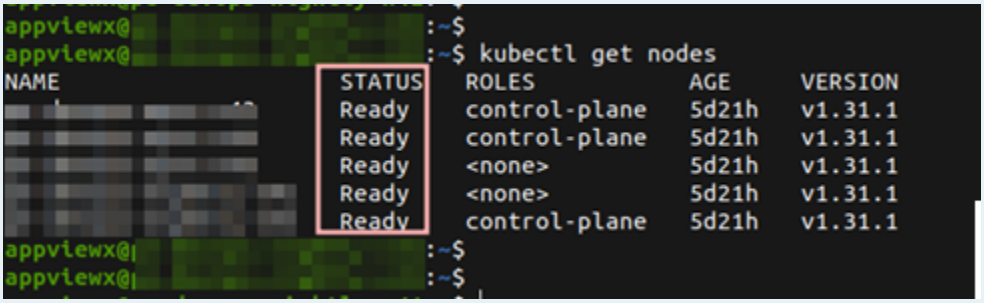
Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
./appviewx.sh --start <nodename>
```



Note: Before proceeding to the next server, ensure that the restarted node is in the “Ready” state by running the following command:

```
kubectl get nodes
```

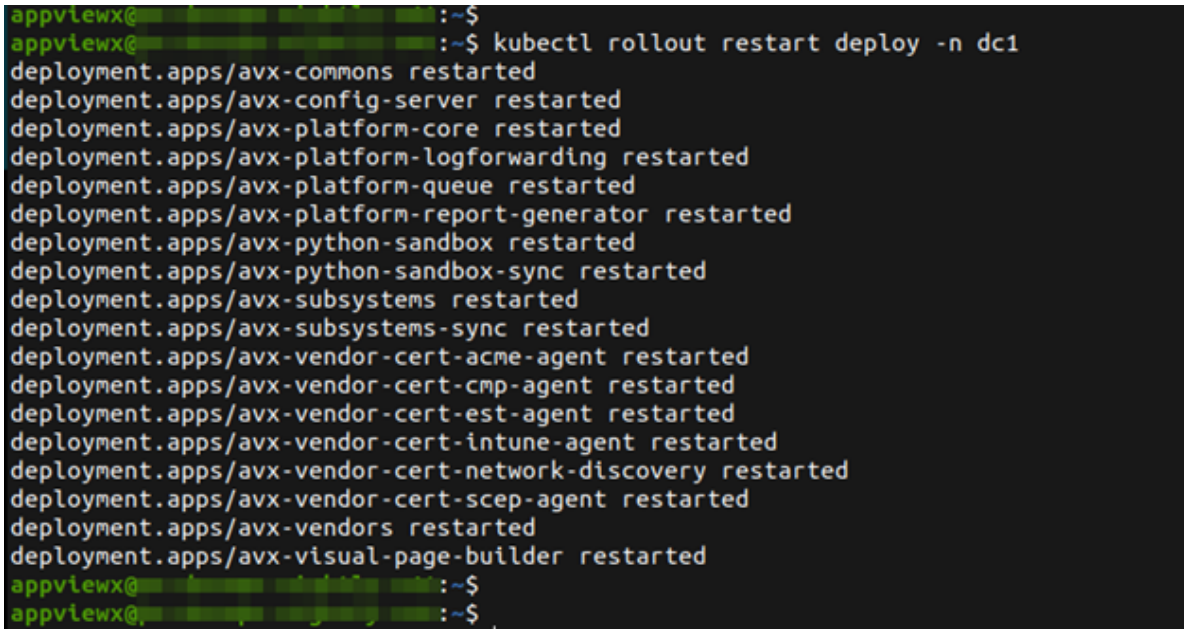


5. Maintain pods distribution on servers post reboot.

   a. After the servers are rebooted, the distribution of pods across the cluster might require realignment. Once all servers in the cluster have been rebooted, run the following commands for all application namespaces:

```
kubectl rollout restart deployment -n <namespace>
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.



b. Verify that the pods are distributed across all available nodes.

```
kubectl get pods -o wide  -n <namespace>
```

c. If there are pods stuck in a terminating state after all activities are complete, they can be forcefully deleted byt the command:

```
kubectl delete pod <pod-name> -n <namespace> --force
```

Adding a Node to the Cluster

This section explains the steps to be followed to add a new node to an existing cluster.

1. Login to the node from where the AppViewX deployment was triggered (installer node).
2. (Optional) Execute this step ONLY when the new node is cloned from an existing node. In this case, you will have to remove the existing Kubernetes configuration files. To delete the existing kubelet config files:

   a. Log in as a root user.

   b. Execute the following command:

   ```
   kubeadm reset -f
   ```

   c. To navigate to the /var/lib/kubelet directory, execute the following command:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
cd /var/lib/kubelet/
```

d. To delete all the files inside the kubelet directory, execute the following command:

```
rm -r *
```

e. To navigate to the /etc/kubernetes directory, execute the following command:

```
cd /etc/kubernetes/
```

f. To delete all the files inside the Kubernetes directory, execute the following command:

```
rm -r *
```

3. Login to the release portal.
4. Download the **new_node_addition.tar.gz** file from the portal.
5. To extract the contents of the file, execute the following command:

```
tar -xvzf new_node_addition.tar.gz
```

The command extracts the contents to a directory named **new_node_addition**.

6. navigate to the **new_node_addition** directory.

The directory contains a package named

- **appviewx_adding_node.tar.gz**
- **node_addition.tar.gz**

.

7. To extract the contents of the **appviewx_adding_node.tar.gz** file, execute the following command:

```
tar -xvzf appviewx_adding_node.tar.gz
```

The command extracts the contents to a directory named **appviewx_adding_node**.

8. Navigate to the **appviewx_adding_node** directory.
9. Copy the files the respective directories
   - To copy all the files and folders to the <installer_location>/appviewx_kubernetes/scripts directory, execute the following command:

```
chart

joining_steps.sh

appviewx_add_node.sh
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
derive_configmap.py

cp -r * <installer_location>/appviewx_kubernetes/scripts/
```

- Manually copy/move the file **node_addition.tar.gz** into the <installer_location>/appviewx_kubernetes folder.

Adding a Master Node

1. For the nodes that will be added using the node addition script, ensure the following prerequisites are met.

   a. Execute/run the prerequisite tool and ensure it meets the tool execution criteria.

   b. Check if the IP forwarding is enabled and firewalld is disabled. If they are not then execute steps c and d.

   c. Enable ip_forward, using the steps below:

      i. Execute the command

      ```
      sudo vi  /etc/sysctl.conf
      ```

      ii. In the conf file add the value `net.ipv4.ip_forward=1`

      iii. Execute the command

      ```
      sudo sysctl -p
      ```

   d. To disable the firewalld, use the command below.

   ```
   sudo systemctl stop firewalld
   ```

2. To add the master node, execute the respective command:

   - Run the command:

   ```
   sudo ./appviewx_add_node.sh
   ```

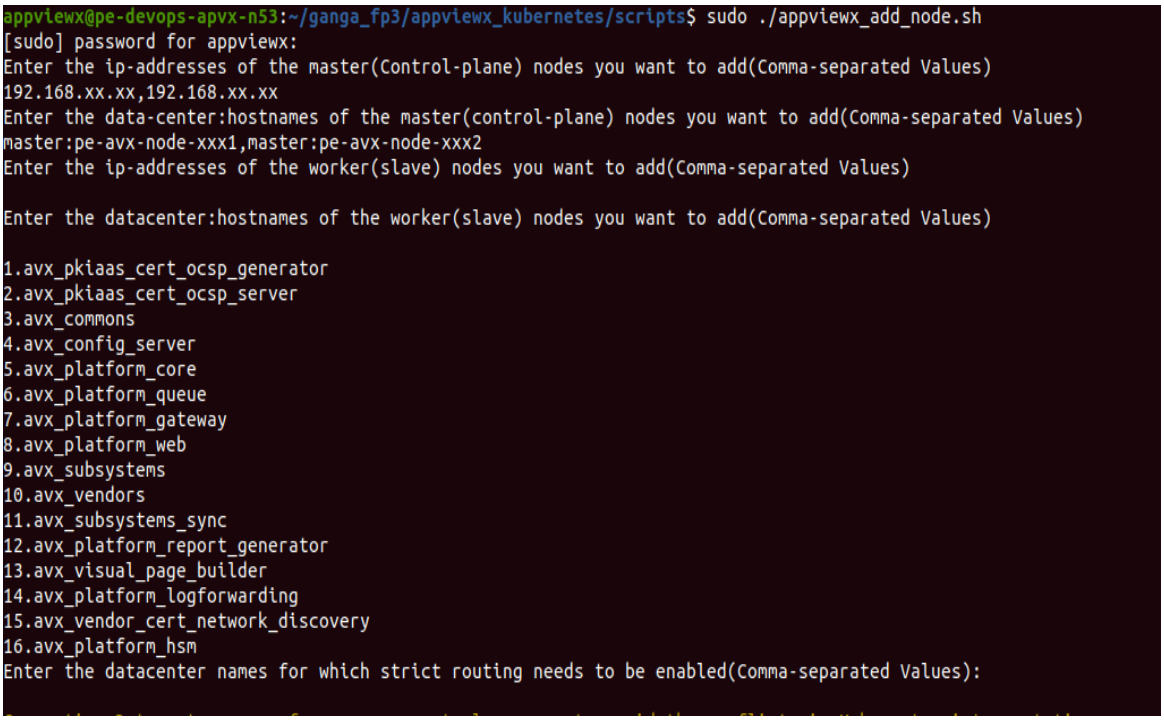
   - The command will prompt you to enter the following details:

     a. IP addresses of new master nodes.

     > **Note:** The total of master nodes must be an odd number (e.g., 3, 5, etc.). Add the master nodes accordingly. To enter multiple IP addresses, separate them with commas (e.g., IP1, IP2).

     b. datacentername:hostname of the new master nodes.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



Note: If you are prompted to enter the worker node details, hit the enter key and proceed.

3. To verify whether the pods are up and running in the new node, execute the following command:

```
kubectl get pods -n <dcname> -o wide
```

Working with Alerts

Alerts are used to notify users when a predefined target or a condition is met. For example, if the memory usage for a cluster exceeds 90%, you can set an email notification to be sent to the users. This type of notification helps in mitigating the dangers of application downtime that might occur when parameters or go unnoticed.

The following alerts are available:

- Application Alerts
- System Alerts

Enabling an Email Alert

AppViewX enables the administrator to send out an email to designated email addresses if the **appviewx.conf** file is modified.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

To enable an email alert when the **appviewx.conf** file is modified:

1. Open the terminal.
2. Navigate to the <avx_installed_path>/conf directory.
3. To open the **appviewx.conf** file, execute the following command:

```
vi appviewx.conf
```

4. Update the following SMTP fields in the **appviewx.conf** file.
   - SMTP_SERVER = <email server>:<port>
   - SMTP_SENDER_USER = <sender email address>
   - SMTP_RECEIVER_USER = <sender email address>
5. To get an email alert if the file is tampered, execute the following command:

```
./appviewx --conf_change_alert cron
```

6. To set the command in crontab, complete the following steps:

```
crontab -e

<cron freq> cd /home/appviewx/appviewx/scripts && ./appviewx --conf_change_alert

cron 2>>/home/appviewx/appviewx/logs/cron_logs 1>/dev/null
```

Working with Backup and Restore

The application level backups are no longer supported in AppViewX. You can back up the mongodb and vault and restore the same in the event of any failure. To facilitate this process, there are scripts available for mongodb and vault backup and restore. You can download them from the release portal.

- Downloading the Scripts
- Backup MongoDB and Vault
- Restore MongoDB and Vault
- Elastic Backup and Restore

Downloading the Scripts

The scripts are used to trigger the backup and restore operations. The backup files will be created under the directory mentioned in the scripts.

Download the following scripts from the release portal:

- **mongo_backup.sh**
- **vault_backup.sh**

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- **vault_restore.sh**
- **mongo_restore.sh**

Copy all the files to the <appviewx_installer_location_path>/appviewx_kubernetes/scripts directory.

Backup MongoDB and Vault

MongoDB Backup

To take a backup of the MongoDB follow the steps below:

1. Navigate to the scripts folder using the command below.

```
cd <appviewx-kubernetes-path>/scripts
```

2. Execute the mongo backup script using the command below.

```
/bin/bash mongo_backup.sh
```

The backup file is created and stored in the location /home/appviewx/<release_label>/appviewx_kubernetes/mongo_backup/mongo_backup_Wed_May_10_02_40_20_EDT_2023.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
[appviewx@pe-iu-rhel-node08 scripts]$ ./mongo_backup.sh
/home/appviewx/Ganga-Fp3/appviewx_kubernetes/scripts
**************************** Fetching running db instance ****************************

mongodb-0
********************************** Fetching db list **********************************

DB list retrieved.
**********************************
admin appSession appviewx appviewxCA config connectedPlatform imageDetails local templateDB workFlowDB workFlowDBEngine
**********************************
Mongo Backup Folder: /appviewx/dependencies/logs//mongo_backup_Wed_May_10_02_27_59_EDT_2023
***************************** Preparing for taking backup *****************************

**********************************
Taking backup of DB: appSession
2023-05-10T06:27:59.538+0000    writing appSession.shiroSession to /appviewx/dependencies/logs/mongo_backup_Wed_May_10_02_27_59_EDT_2023/appSession/shiroSession.bson
2023-05-10T06:27:59.541+0000    done dumping appSession.shiroSession (2 documents)
**********************************
**********************************
Taking backup of DB: appviewx
2023-05-10T06:27:59.977+0000    writing appviewx.logging to /appviewx/dependencies/logs/mongo_backup_Wed_May_10_02_27_59_EDT_2023/appviewx/logging.bson
2023-05-10T06:27:59.977+0000    writing appviewx.certificateContent to /appviewx/dependencies/logs/mongo_backup_Wed_May_10_02_27_59_EDT_2023/appviewx/certificateConte
nt.bson
2023-05-10T06:27:59.978+0000    writing appviewx.certificateContent_backup to /appviewx/dependencies/logs/mongo_backup_Wed_May_10_02_27_59_EDT_2023/appviewx/certifica
teContent_backup.bson
2023-05-10T06:27:59.995+0000    writing appviewx.commandRepository to /appviewx/dependencies/logs/mongo_backup_Wed_May_10_02_27_59_EDT_2023/appviewx/commandRepository
.bson
2023-05-10T06:28:00.012+0000    done dumping appviewx.logging (4365 documents)
2023-05-10T06:28:00.013+0000    writing appviewx.visualworkflow_task_component to /appviewx/dependencies/logs/mongo_backup_Wed_May_10_02_27_59_EDT_2023/appviewx/visua
lworkflow_task_component.bson
2023-05-10T06:28:00.022+0000    done dumping appviewx.certificateContent (1559 documents)
2023-05-10T06:28:00.024+0000    writing appviewx.visualworkflow_template_detail to /appviewx/dependencies/logs/mongo_backup_Wed_May_10_02_27_59_EDT_2023/appviewx/visu
alworkflow_template_detail.bson
2023-05-10T06:28:00.024+0000    done dumping appviewx.certificateContent_backup (1506 documents)
```

```
mongo_backup_Wed_May_10_02_27_59_EDT_2023/connectedPlatform/dashboardOpenCount.bson
mongo_backup_Wed_May_10_02_27_59_EDT_2023/connectedPlatform/provisioningListenerData.bson
mongo_backup_Wed_May_10_02_27_59_EDT_2023/connectedPlatform/accountListenerData.bson
mongo_backup_Wed_May_10_02_27_59_EDT_2023/connectedPlatform/certInventoryListener.bson
mongo_backup_Wed_May_10_02_27_59_EDT_2023/connectedPlatform/adcListenerData.bson
mongo_backup_Wed_May_10_02_27_59_EDT_2023/imageDetails/
mongo_backup_Wed_May_10_02_27_59_EDT_2023/imageDetails/fs.files.metadata.json
mongo_backup_Wed_May_10_02_27_59_EDT_2023/imageDetails/fs.chunks.metadata.json
mongo_backup_Wed_May_10_02_27_59_EDT_2023/imageDetails/fs.chunks.bson
mongo_backup_Wed_May_10_02_27_59_EDT_2023/imageDetails/fs.files.bson
mongo_backup_Wed_May_10_02_27_59_EDT_2023/templateDB/
mongo_backup_Wed_May_10_02_27_59_EDT_2023/templateDB/fs.chunks.metadata.json
mongo_backup_Wed_May_10_02_27_59_EDT_2023/templateDB/fs.files.metadata.json
mongo_backup_Wed_May_10_02_27_59_EDT_2023/templateDB/fs.files.bson
mongo_backup_Wed_May_10_02_27_59_EDT_2023/templateDB/fs.chunks.bson
mongo_backup_Wed_May_10_02_27_59_EDT_2023/workFlowDB/
mongo_backup_Wed_May_10_02_27_59_EDT_2023/workFlowDB/workFlowTemplate.metadata.json
mongo_backup_Wed_May_10_02_27_59_EDT_2023/workFlowDB/workFlowTemplate.bson
mongo_backup_Wed_May_10_02_27_59_EDT_2023.tar.gz                                                    100%   54MB 317.2MB/s   00:00
Copied backup in installer node successfully. Location : /home/appviewx/Ganga-Fp3/appviewx_kubernetes/mongo_backup/mongo_backup_Wed_May_10_02_27_59_EDT_2023.tar.gz

[appviewx@pe-iu-rhel-node08 scripts]$ cd ../mongo_backup/
[appviewx@pe-iu-rhel-node08 mongo_backup]$ ls -lrt
total 55664
-rw-r--r--. 1 appviewx appviewx 56998032 May 10 02:28 mongo_backup_Wed_May_10_02_27_59_EDT_2023.tar.gz
[appviewx@pe-iu-rhel-node08 mongo_backup]$ pwd
/home/appviewx/Ganga-Fp3/appviewx_kubernetes/mongo_backup
[appviewx@pe-iu-rhel-node08 mongo_backup]$
```

Vault Backup

To take a backup of the Vault follow the steps below:

1. Navigate to the scripts folder using the command below.

   ```
   cd <appviewx-kubernetes-path>/scripts
   ```

2. Execute the mongo backup script using the command below.

   ```
   /bin/bash vault_backup.sh
   ```

   The backup file is created and stored in the location /home/appviewx/<release_label>/appviewx_kubernetes/vault_backup/vault_backup_Wed_May_10_02_40_20_EDT_2023.

```
[appviewx@pe-iu-node36 scripts]$ ./vault_backup.sh
/home/appviewx/Hudson/appviewx_kubernetes/scripts
Vault Backup File: /home/appviewx/Hudson/appviewx_kubernetes/vault_backup/vault_backup_Fri_Jul_7_10_15_34_IST_2023
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Restore MongoDB and Vault

MongoDB Restore

To restore the MongoDB follow the steps below:

1. Navigate to the scripts folder using the command below.

```
cd <appviewx-kubernetes-path>/scripts
```

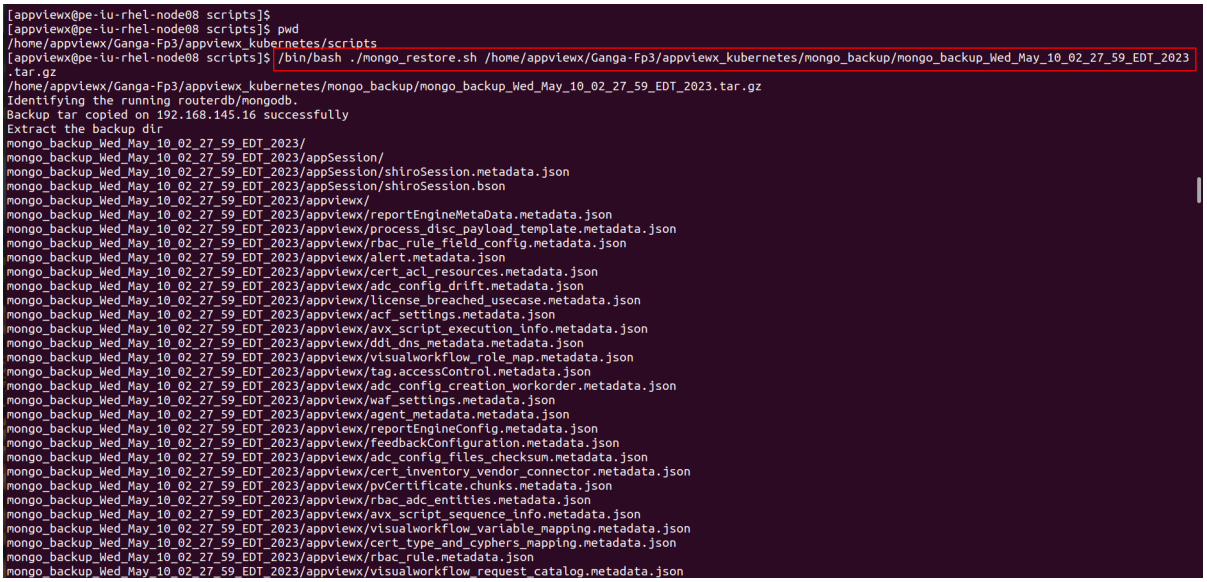
2. Execute the script to restore the backup file using the command below.

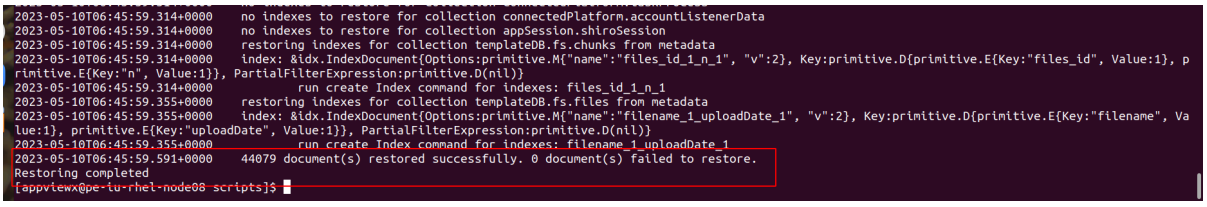
```
/bin/bash mongo_restore.sh <appviewx-kubernetes-path>/mongo_backup/<mongo-backup-file>
```

Example

```
/bin/bash

 mongo_restore.sh /home/appviewx/<release_label>/appviewx_kubernetes/mongo_backup/mongo_backup_Wed_May_10_02_27_59_EDT_2023.tar.gz
```





Vault Restore

To restore the Vault follow the steps below:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

1. Navigate to the scripts folder using the command below.

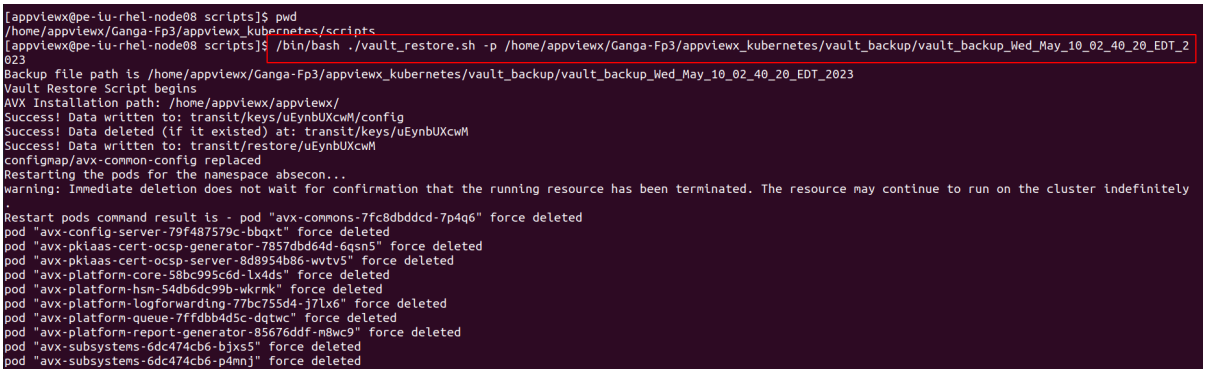
```
cd <appviewx-kubernetes-path>/scripts
```

2. Execute the script to restore the backup file using the command below.

```
/bin/bash vault_restore.sh -p <appviewx-kubernetes-path>/vault_backup/<vault-backup-file>
```

Example

```
/bin/bash vault_restore.sh -p /home/appviewx/<release_label>/appviewx_kubernetes/vault_backup/vault_backup_Wed_May_10_02_40_20_EDT_2023
```





Elastic Backup and Restore

Elastic Backup

The script **elastic_backup.py** is for elastic backup. To manually perform the elastic backup,

1. Navigate to the scripts directory.
2. Run the script using the application provided python binary, example

```
/home/appviewx/appviewx/appviewx_dependency/appviewx_addons/Python_Linux/bin/python
```

3. Before taking the backup we have to set up the Elasticsearch repo for storing the snapshots.

```
/home/appviewx/appviewx/appviewx_dependency/appviewx_addons/Python_Linux/bin/python elastic_backup.py --setup elasticsearch_insight
```

4. To take the backup run the command

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
/home/appviewx/appviewx/appviewx_dependency/appviewx_addons/Python_Linux/bin/python elastic_backup.py --backup elasticsearch_insight
```

5. Perform the following steps after taking the backup
   a. Navigate to the clusters node where the elasticsearch-insight is deployed
   b. Navigate to the installer path and create the tar of the elastic-insight-backup directory
   c. Copy the backup tar into the installer node.

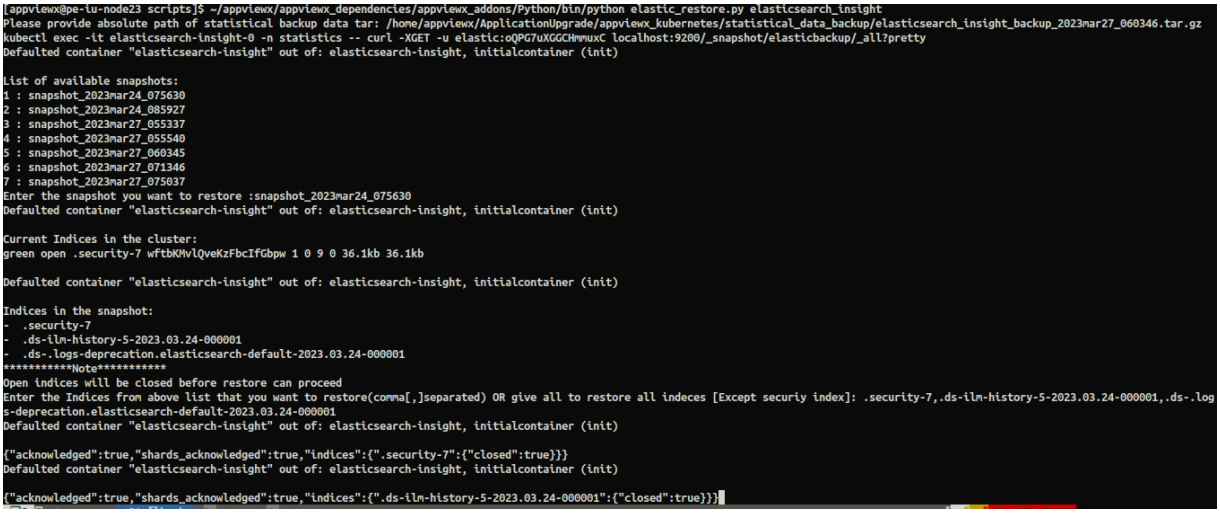
Elastic Restore

The script **elastic_restore.py** is used for restore. To manually perform the elastic restore,

1. Navigate to the scripts directory.
2. Run the **elastic_restore.py** script to restore the backup

```
/home/appviewx/appviewx/appviewx_dependency/appviewx_addons/Python_Linux/bin/python elastic_restore.py elasticsearch_insight
```

3. Script will ask for the backup tar which was created manually. Provide the absolute path of the backup tar.



4. Script will list all the available snapshots that has date and time in the naming. Select the backup which you want to restore.
5. Provide the details of the indices you want to restore (follow the screenshot above).

Working with Logs

In any application, log files are used to record all events. It provides information about the customer usage patterns, the names of modules that are used frequently. In addition, they also help users analyse the issues depending on the events.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

In any application, there are mainly two types of logs that are collected. One of them is application logs that are required to monitor the performance. Another type of log that is maintained is infrastructure logs. These logs are used to monitor the status of the hardware infrastructure like memory usage, disk usage, and CPU usage.

In AppViewX, the log files are collected and maintained for plugins. To manage logs, AppViewX uses Kibana.

- Managing Logs using Kibana
- Managing Logs using AppViewX Nodes
- Automatic Log Collection
- Log Analyser Tool
- Istio Ingress Gateway Access Logs Configuration

Managing Logs using Kibana

Kibana is an open user interface that enables you to graphically represent the log files and monitor system performance.

Before using Kibana, ensure that you have the following:

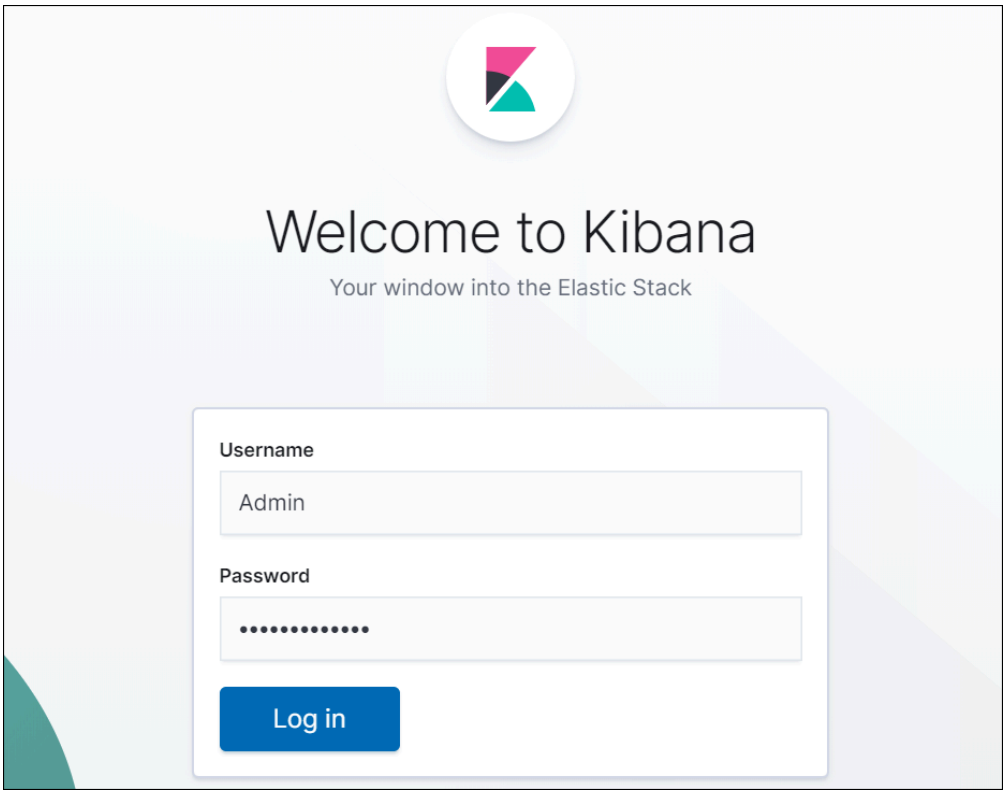
- **Kibana Web URL** - mentioned in the <INSTALLATION_PATH>/.appviewx_configuration file
- **Kibana Username**- mentioned in the <INSTALLATION_PATH>/.appviewx_configuration file
- **Kibana Password** - mentioned in the <INSTALLATION_PATH>/.appviewx_configuration file

- Accessing Kibana
- Creating an Index Pattern
- Viewing Logs
- Generating a Report

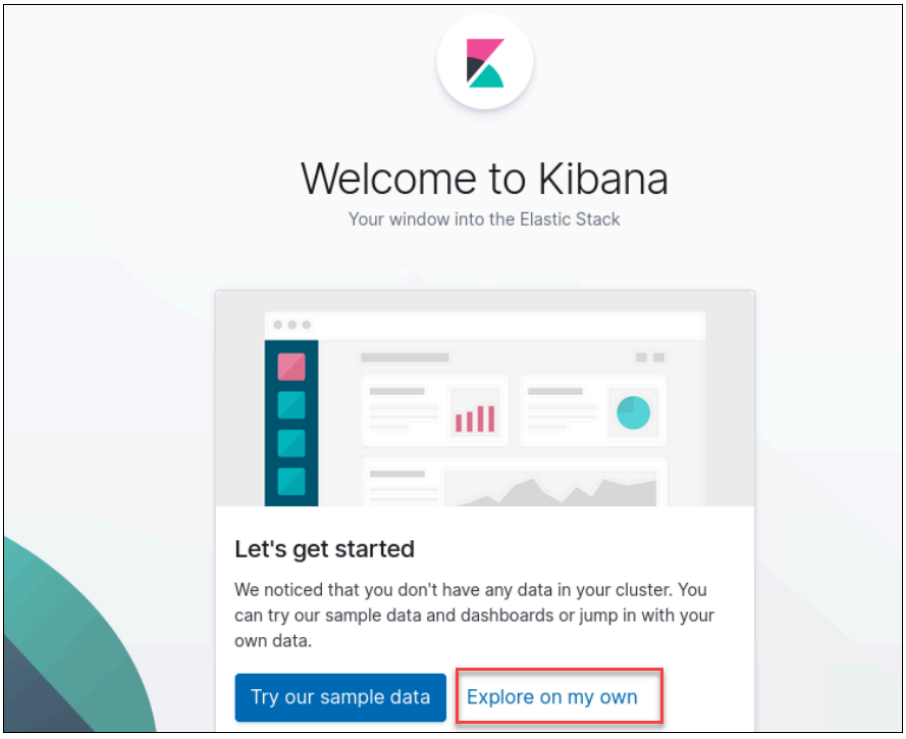
Accessing Kibana

1. Open the Kibana Web URL.
2. Enter the credentials.
3. Click **Log in**.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



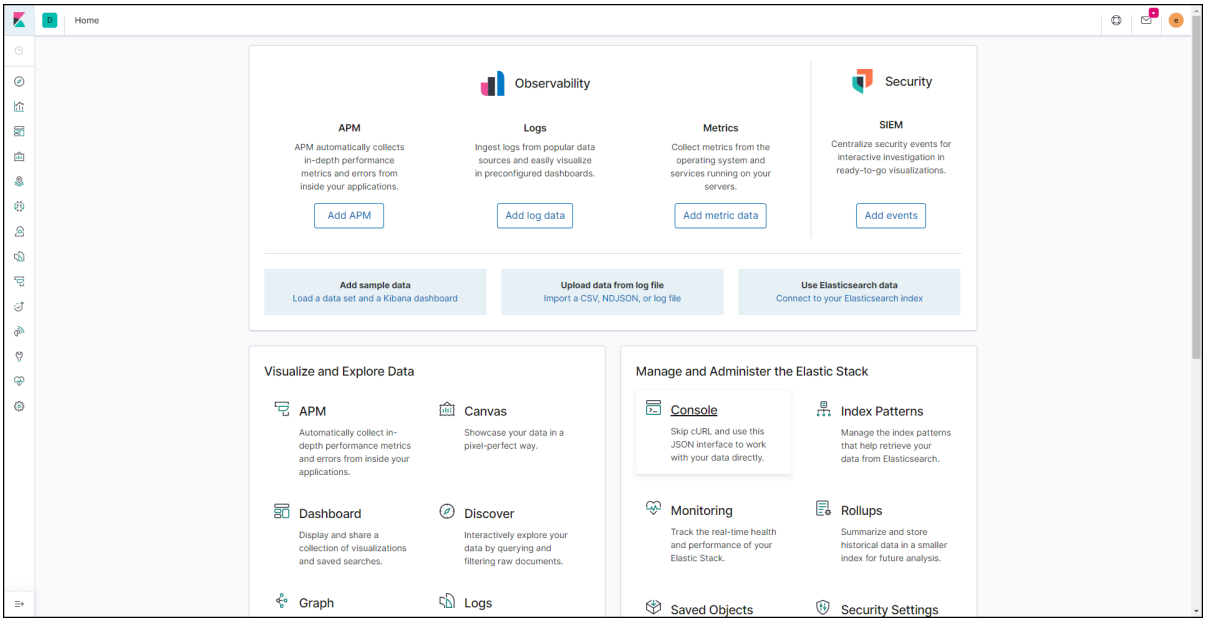
The **Let's get started** page is displayed.



4. Click **Explore on my own**.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

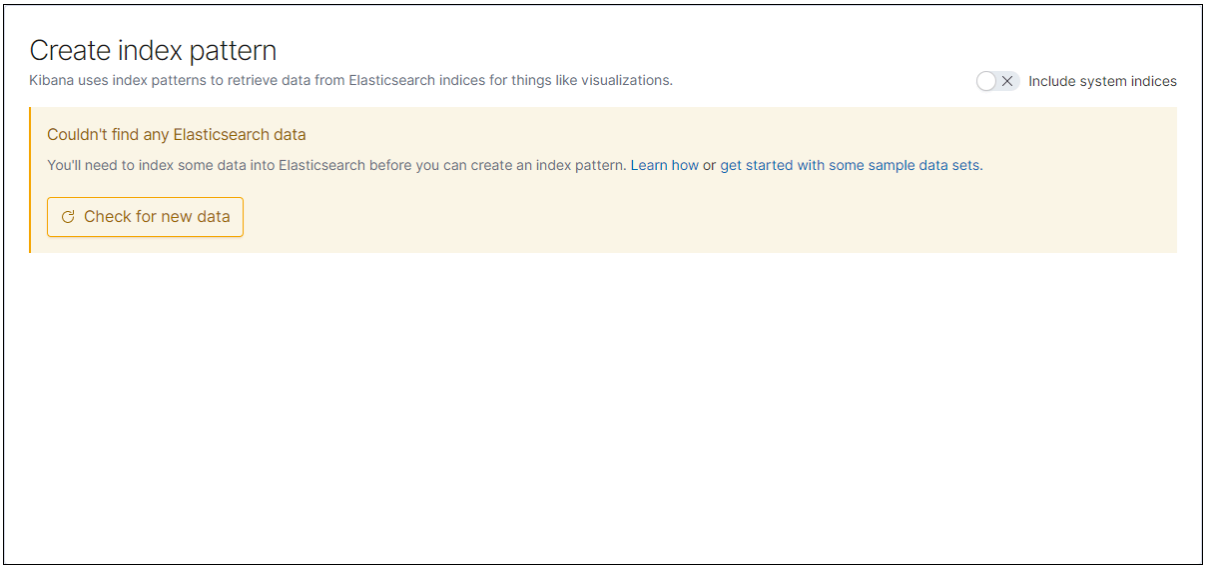
The **Home page** is displayed.



Creating an Index Pattern

1. Login to Kibana.
2. Under **Visualize and Explore Data**, click **Visualize**.

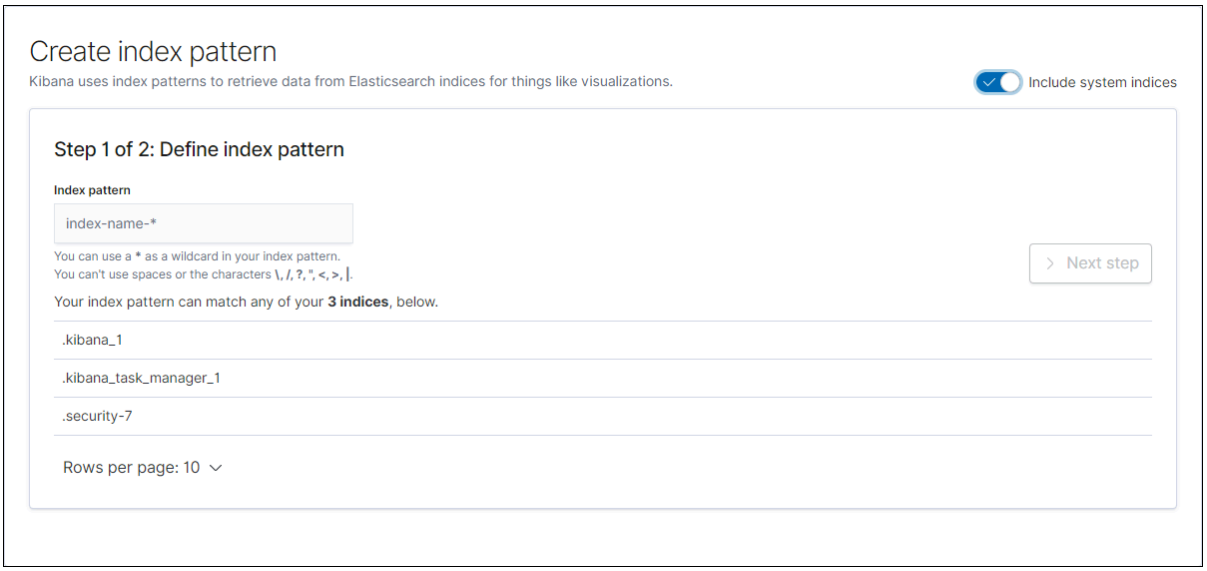
   The **Create index pattern** page is displayed.



3. Enable the **Include system indices** option.

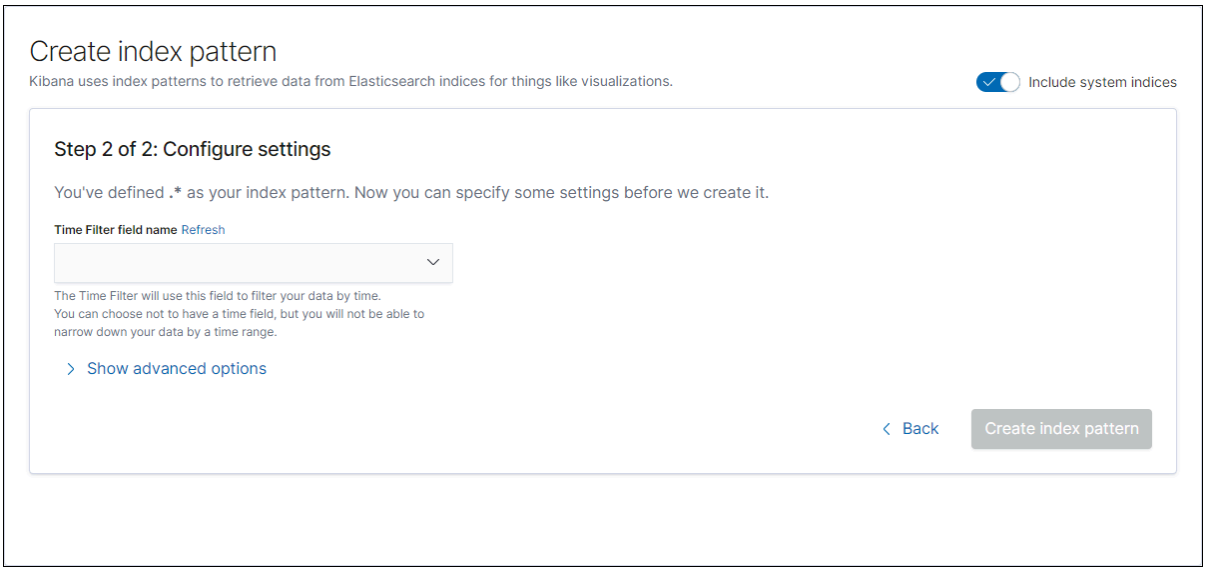
   The **Define index pattern** page is displayed.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



4. Under Index pattern, enter .*.
5. Click **Next step**.

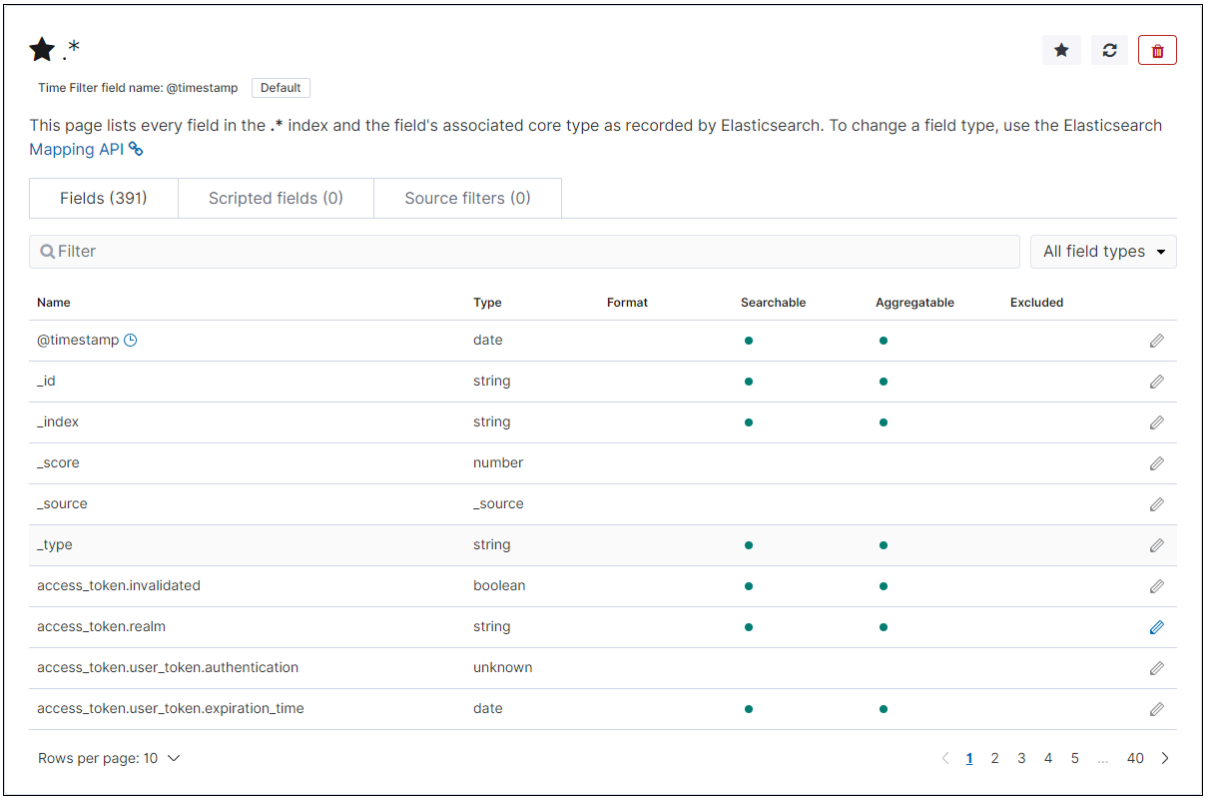
   The **Configure settings** page is displayed.



6. From the **Time Filter field name** list, select **@timestamp**.
7. Click **Create Index Pattern**.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

The system creates an index pattern.

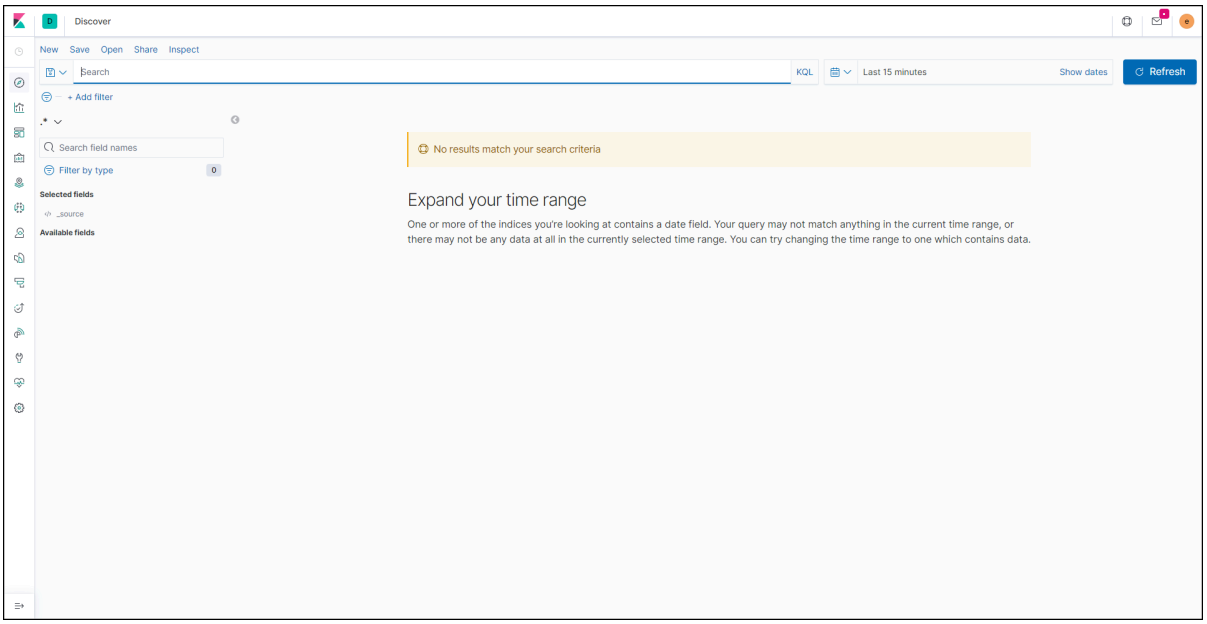


Viewing Logs

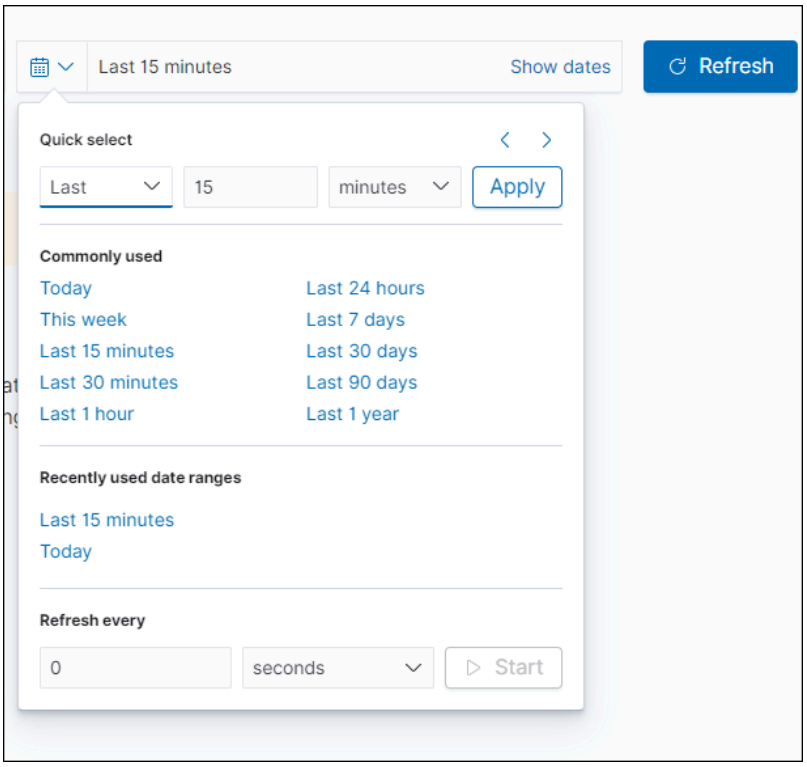
1. Login to Kibana.
2. Under **Visualize and Explore Data**, click **Discover**.

   The **Discover** page is displayed.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



3. In the time frame section, select the time frame within which the logs need to be captured.

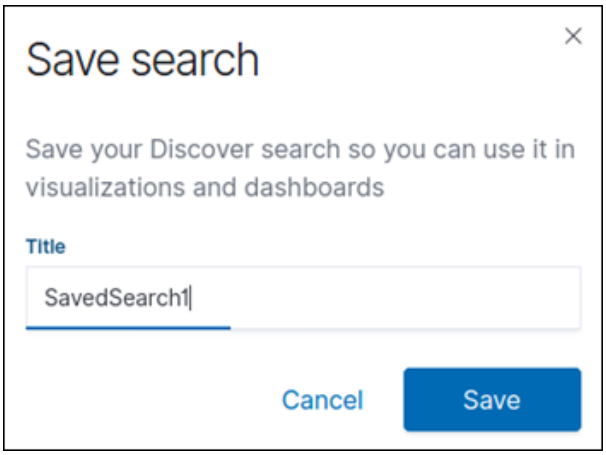


4. To view the updated logs, click **Refresh**.
5. To save the search:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

a. Click **Save**.

b. Enter a valid name to save the search.

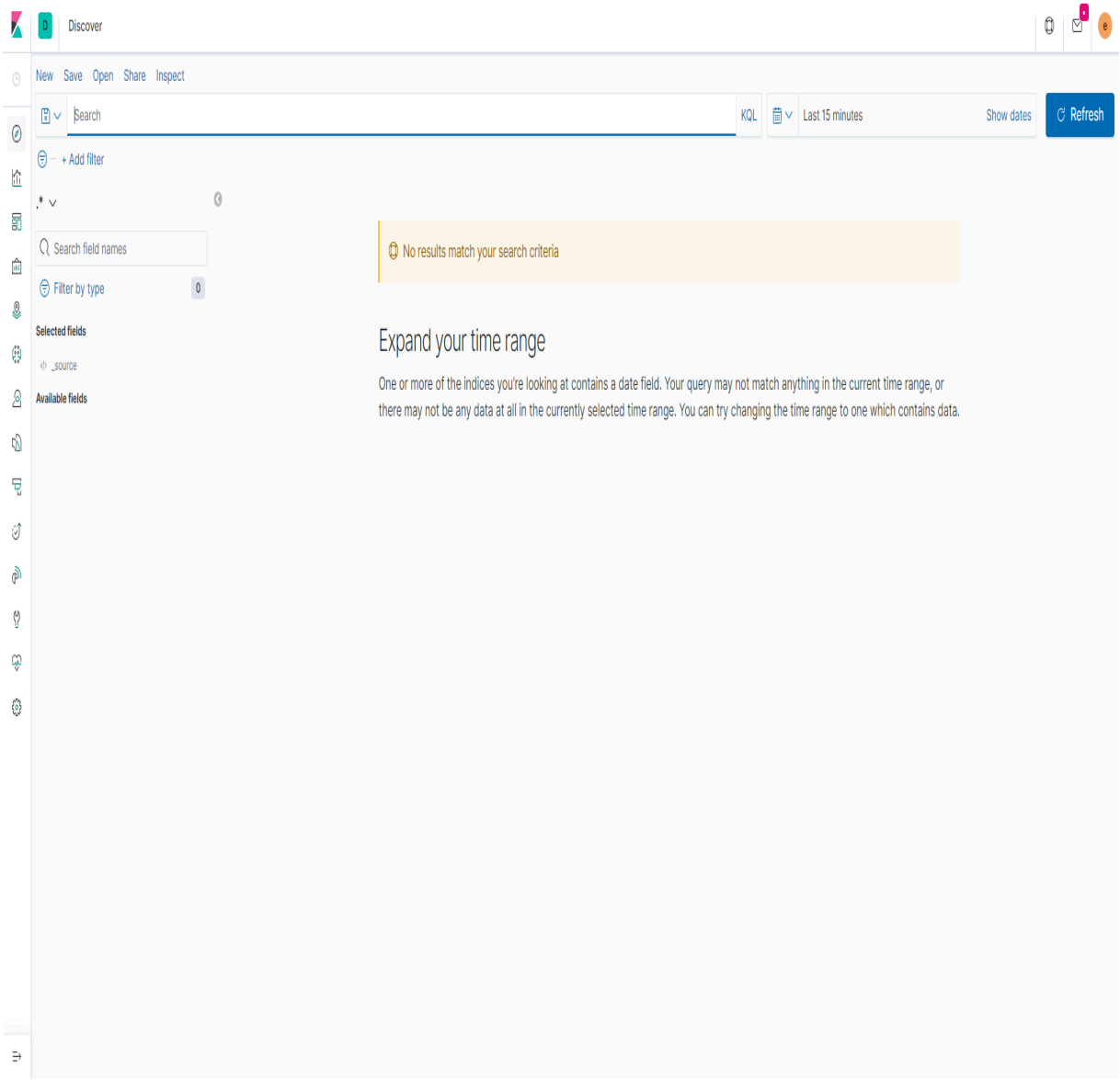


Generating a Report

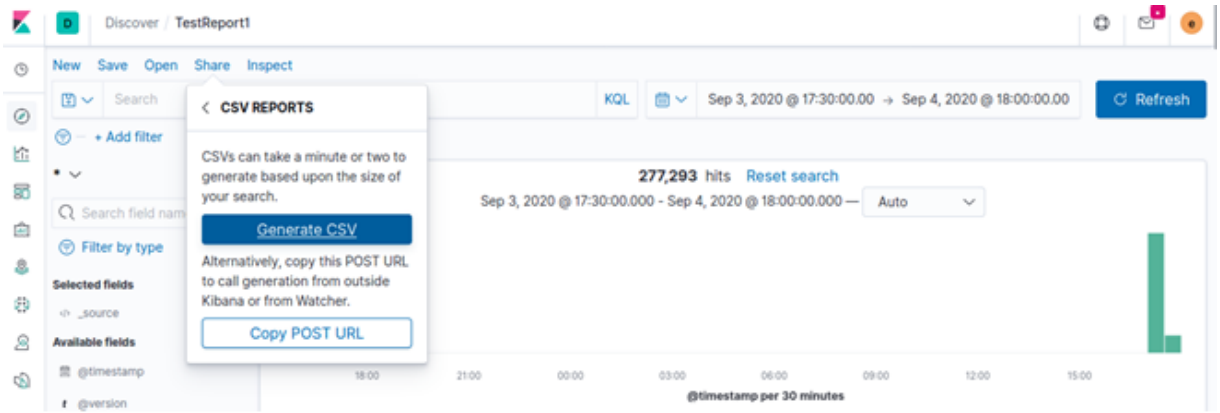
Kibana enables you to generate a report in CSV format. In order to generate the report, you must copy the <.ndjson> ext files from the <InstallerLocation>/appviewx_kubernetes/yaml/appviewx_monitoring/kibana/deploy location and import into the import section (for example, <gateway.ndjson> and <platform.ndjson>).

1. Login to Kibana.
2. Under **Visualize and Explore Data**, click **Discover**.
   The **Discover** page is displayed.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

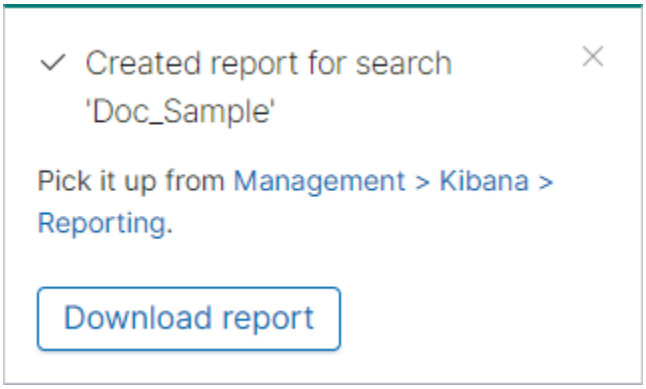


3. Select **Share > CSV reports > Generate CSV**.



Copyright © 2025 AppViewX Inc. All Rights Reserved.

The system generates the report and prompts to download the same.



4. To download the report, click **Download report**.
   The system downloads the report to the default download location.

Managing Logs using AppViewX Nodes

You can also view and manage the log files even if you do not have Kibana installed. In this case, you can use the AppViewX nodes to view and manage the log files. You can also view logs using the command line interface before you install the ELK.

Note: Logs are maintained as per the retention policy. Any log exceeding 30 MB will be rotated and archived as part of the data retention policy.

To view the logs:

1. Log in to the respective node.
2. Navigate to the appviewx/dependencies/logs directory. You can view the CLI logs for pods in the same node.

To view the logs from the AppViewX nodes:

1. Using the command line interface, log in to the AppViewX node.
2. To fetch the node name in which the pod is running, execute the following command:

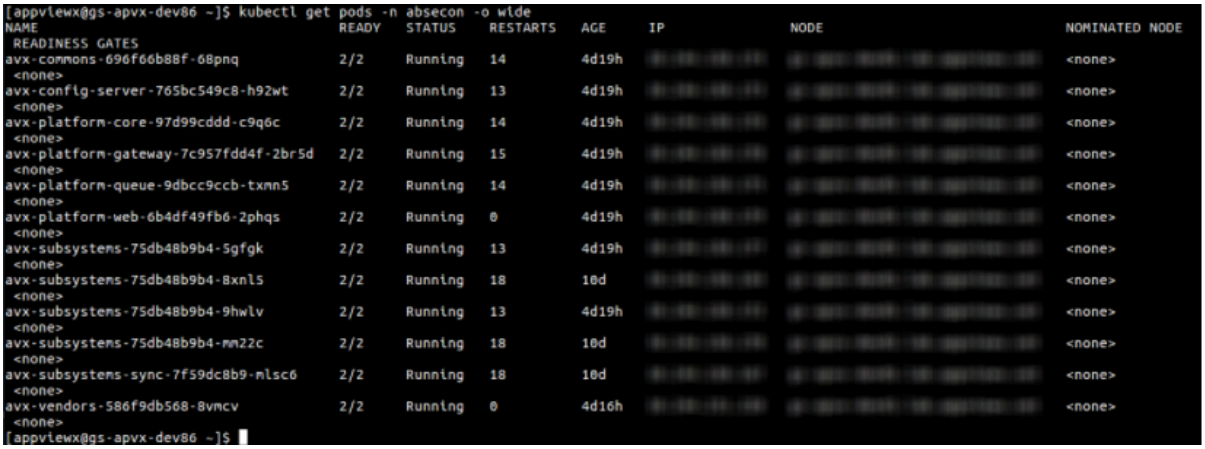
```
kubectl get pods- n <dc> -o wide
```

3. Log in to the respective node using SSH.
4. Navigate to `<INSTALLATION_PATH>/logs` for all log files.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

For example, If you want to view the logs for the subsystem plugin in the datacenter DC1, execute the following command to get the node name of the pod:

```
kubectl get pods -n DC1 -o wide
```



```
[appviewx@gs-apvx-dev86 ~]$ kubectl get pods -n absecon -o wide
NAME                                  READY   STATUS    RESTARTS   AGE     IP              NODE                                  NOMINATED NODE
 READINESS GATES
avx-commons-696f66b88f-68pnq          2/2     Running   14         4d19h   [redacted]      [redacted]                            <none>
 <none>
avx-config-server-765bc549c8-h92wt    2/2     Running   13         4d19h   [redacted]      [redacted]                            <none>
 <none>
avx-platform-core-97d99cddd-c9q6c     2/2     Running   14         4d19h   [redacted]      [redacted]                            <none>
 <none>
avx-platform-gateway-7c957fdd4f-2br5d 2/2     Running   15         4d19h   [redacted]      [redacted]                            <none>
 <none>
avx-platform-queue-9dbcc9ccb-txmn5    2/2     Running   14         4d19h   [redacted]      [redacted]                            <none>
 <none>
avx-platform-web-6b4df49fb6-2phqs     2/2     Running   0          4d19h   [redacted]      [redacted]                            <none>
 <none>
avx-subsystems-75db48b9b4-5gfgk       2/2     Running   13         4d19h   [redacted]      [redacted]                            <none>
 <none>
avx-subsystems-75db48b9b4-8xnl5       2/2     Running   18         10d     [redacted]      [redacted]                            <none>
 <none>
avx-subsystems-75db48b9b4-9hwlv       2/2     Running   13         4d19h   [redacted]      [redacted]                            <none>
 <none>
avx-subsystems-75db48b9b4-mm22c       2/2     Running   18         10d     [redacted]      [redacted]                            <none>
 <none>
avx-subsystems-sync-7f59dc8b9-nlsc6   2/2     Running   18         10d     [redacted]      [redacted]                            <none>
 <none>
avx-vendors-586f9db568-8vmcv          2/2     Running   0          4d16h   [redacted]      [redacted]                            <none>
 <none>
[appviewx@gs-apvx-dev86 ~]$
```

Note: For troubleshooting issues, please refer to the Troubleshooting section.

Automatic Log Collection

The **collect_logs.sh** script (under the *scripts_util* directory) has been updated to collect all logs from all the nodes in a cluster setup (single node and multi node). After all the logs are collected, the log dump directory is cleaned up and the tar ball (tar file) is created with all the generated logs and available for download.

Prerequisite

1. Ensure that the AppViewX application is updated with latest 2022.1.0 FP3 patch or upgraded to 2023.1.0 (FP1-FP3), 2024.3.0.0
2. Input (sudo) passwords for **message** and **mongodb** logs.

To obtain the all the logs

1. Go to the terminal prompt and navigate to the **scripts** folder by executing

```
cd /home/<folder_location>/appviewx_kubernetes/scripts
```

2. To collect all logs, execute the following command,

```
./appviewx.sh - -collect-logs all-logs
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
bash-4.2$ ./appviewx.sh --collect-logs all-logs
Input Validation Completed Successfully !
Please enter appviewx password of master:pe-devops-appvx-node10.lab.appviewx.net :
Please enter appviewx password of absecon:pe-devops-appvx-node7.lab.appviewx.net :
Please enter appviewx password of absecon:pe-devops-appvx-node8.lab.appviewx.net :
Please enter appviewx password of absecon:pe-devops-appvx-node9.lab.appviewx.net :
```

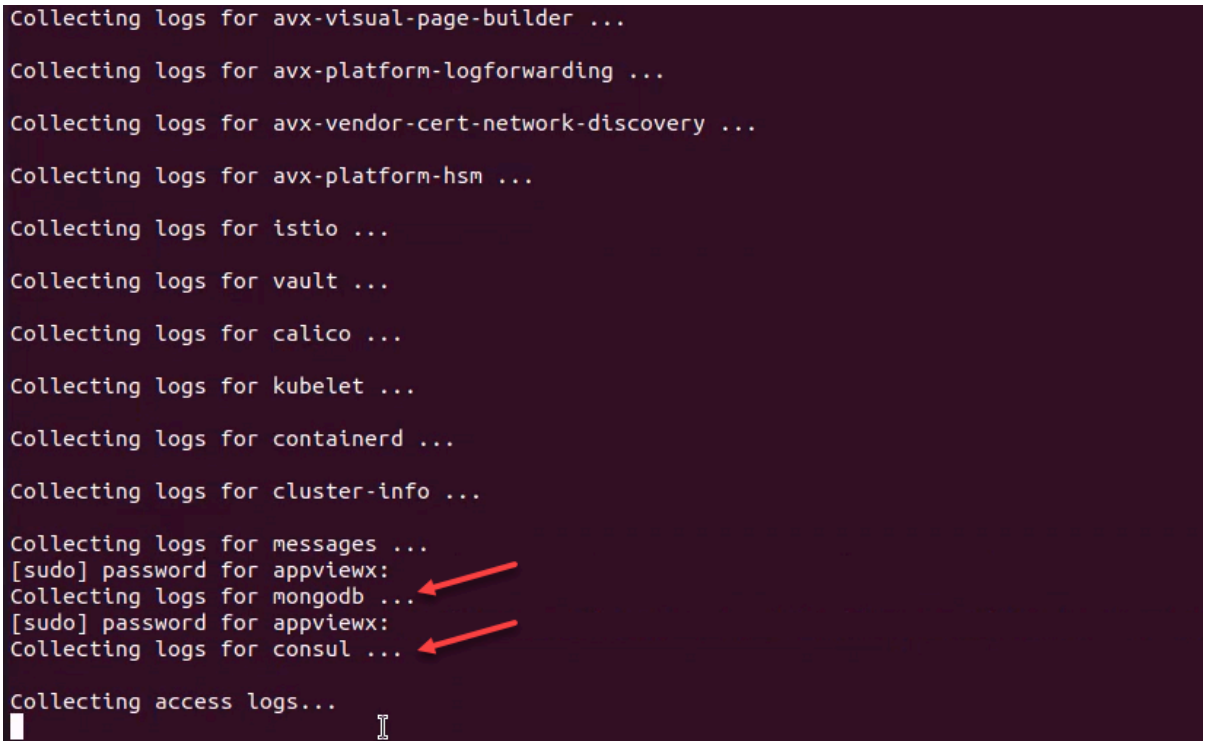
The message "*Input validation completed successfully*" is displayed. Since the command for all-logs include logs from mongodb and message, you will be prompted to enter the passwords.

3. Enter the respective sudo passwords and hit the Enter key.

The below messages are displayed once the log collection starts. If there are no logs for any system then 'No logs present for…." Is displayed.

```
Starting files copy...
No logs present for appviewx-dependencies ...
Collecting logs for avx-pkiaas-cert-ocsp-generator ...
Collecting logs for avx-pkiaas-cert-ocsp-server ...
Collecting logs for avx-commons ...
Collecting logs for avx-crontab ...
Collecting logs for avx-config-server ...
Collecting logs for avx-platform-core ...
Collecting logs for avx-platform-queue ...
Collecting logs for avx-platform-gateway ...
Collecting logs for avx-platform-web ...
Collecting logs for avx-subsystems ...
Collecting logs for avx-vendors ...
Collecting logs for avx-subsystems-sync ...
Collecting logs for avx-platform-report-generator ...
Collecting logs for avx-visual-page-builder ...
Collecting logs for avx-platform-logforwarding ...
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

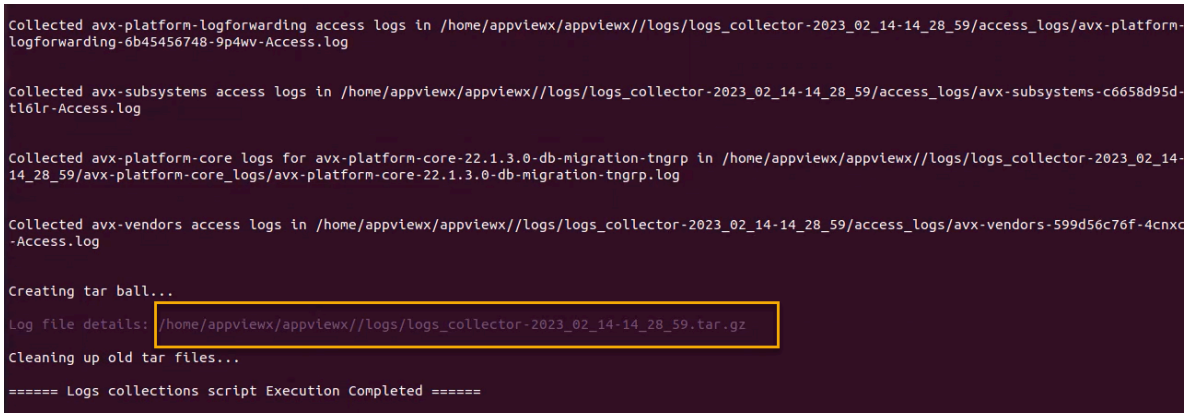


Note: The passwords during the log collection process will be taken automatically as they were entered at the beginning of the log collection process.

4. The log collection process continues with the following steps
   a. Logs files are created for each application (plugins, DBs) and are saved in a folder location as indicated.
   b. The final process is creation of the tar ball. The tar ball contains all the logs, its location is displayed in the end.
   c. Old tar files (up to two days old) are deleted from the folder.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

d. Process ends with the message "*Log collection script execution completed.*"



Individual logs can also be collected by executing for each system as mentioned below:

```
./appviewx.sh - -collect-logs all-plugins
```

```
./appviewx.sh - -collect-logs avx-platform-gateway
```

```
./appviewx.sh - -collect-logs mongo
```

```
./appviewx.sh - -collect-logs istio
```

```
./appviewx.sh - -collect-logs openbao
```

```
./appviewx.sh - -collect-logs calcio
```

```
./appviewx.sh - -collect-logs consul
```

```
./appviewx.sh - -collect-logs kubelet
```

```
./appviewx.sh - -collect-logs containerd
```

```
./appviewx.sh - -collect-logs cluster-info
```

```
./appviewx.sh - -collect-logs messages
```

```
./appviewx.sh - -collect-logs access
```

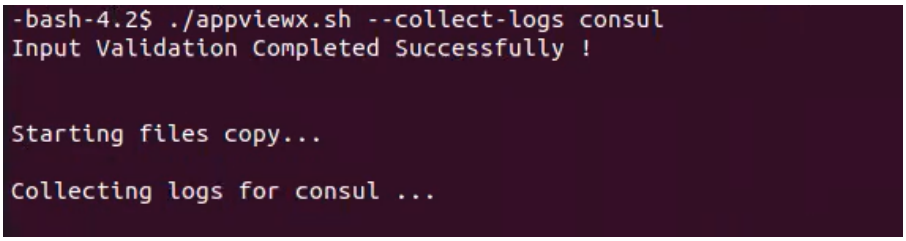
```
/appviewx.sh - -collect-logs <deployment-name> [supported:

 avx-subsystems,avx-vendors,avx-platform-core,avx-platform-queue,kubelet,containerd,messages]
```

Note: <deployment-name> will recognize only one argument from the supported list - [supported: avx-subsystems,avx-vendors,avx-platform-core,avx-platform-queue,kubelet,containerd,messages]

Copyright © 2025 AppViewX Inc. All Rights Reserved.

An example of execution for collection of logs for consul is shown below. This execution will process without the password input.



Note: In case you encounter any problems while using this tool, kindly capture a screenshot of the CLI output and reach out to the support team for assistance.

Log Analyser Tool

This topic contains the installation steps to for the log analyzer tool (logmon). A new component has been added in the existing **scripts** files and a parameter LOGMON_HOST added to the **appview.conf** file to support this feature. To intall the log analyser tool

1. Login to the release portal and download the following filesfile. executing
   a. **scripts.tar.gz**
   b. **appviewx_kubernetes_logmon_20xx.x.x_FPx.tar.gz**
2. Copy both the above files to the installer node.
3. Sync updated scripts with existing installer scripts using the commands below.

```
tar -xf scripts.tar.gz
```

```
cp -r scripts/* <INSTALLER_PATH>/appviex_kuberetes/scripts
```

4. Navigate to installer scripts folder.
5. Sync the **appviewx.conf** file using the command below.

```
./appviewx.sh --conf-merge
```

6. Install the logmon components using the commands below.

```
chmod +x logmon_install.sh
```

```
./logmon_install.sh
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Istio Ingress Gateway Access Logs Configuration

To access the istio ingress gateway access logs configure the ef-ingress-gateway-access-log file as described below.

1. Log in to the installer node and go to the **envoy_filter** directory.

```
cd <appviewx_kubernetes_path>/scripts/envoy_filter
```

2. Create the **ef-ingress-gateway-access-log.yaml** file and paste the code below.

```
apiVersion: networking.istio.io/v1alpha3
kind: EnvoyFilter
metadata:
 name: ingressgateway-access-logs
 namespace: istio-system
spec:
 workloadSelector:
  labels:
   istio: ingressgateway
 configPatches:
 - applyTo: NETWORK_FILTER
  match:
   context: GATEWAY
   listener:
    filterChain:
     filter:
      name: "envoy.filters.network.http_connection_manager"
  patch:
   operation: MERGE
   value:
    typed_config:
     "@type": type.googleapis.com/envoy.extensions.filters.network.http_connection_manager.v3.HttpConnectionManager
     access_log:
     - name: envoy.access_loggers.file
      typed_config:
       "@type": type.googleapis.com/envoy.extensions.access_loggers.file.v3.FileAccessLog
       path: "/var/log/istio-ingressgateway-access.log"
       log_format:
        json_format:
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
        "start_time": "%START_TIME%"
        "method": "%REQ(:METHOD)%"
        "path": "%REQ(X-ENVOY-ORIGINAL-PATH?:PATH)%"
        "protocol": "%PROTOCOL%"
        "response_code": "%RESPONSE_CODE%"
        "response_flags": "%RESPONSE_FLAGS%"
        "bytes_received": "%BYTES_RECEIVED%"
        "bytes_sent": "%BYTES_SENT%"
        "duration": "%DURATION%"
        "upstream_cluster": "%UPSTREAM_CLUSTER%"
        "user_agent": "%REQ(USER-AGENT)%"
        "downstream_remote_address": "%DOWNSTREAM_REMOTE_ADDRESS%"
        "downstream_local_address": "%DOWNSTREAM_LOCAL_ADDRESS%"
        "upstream_service_time": "%RESP(X-ENVOY-UPSTREAM-SERVICE-TIME)%"
        "x_forwarded_for": "%REQ(X-FORWARDED-FOR)%"
```

3. Execute the command below to apply the envoy filter.

```
kubectl apply -f ef-ingress-gateway-access-log.yaml
```

```
appviewx@pe-devops-apvx-n75:~/Thames-FP1/appviewx_kubernetes/scripts/envoy_filter$ kubectl apply -f ef-ingress-gateway-access-log.yaml
Warning: EnvoyFilter exposes internal implementation details that may change at any time. Prefer other APIs if possible, and exercise extreme caution,
 especially around upgrades.
envoyfilter.networking.istio.io/ingressgateway-access-logs created
appviewx@pe-devops-apvx-n75:~/Thames-FP1/appviewx_kubernetes/scripts/envoy_filter$
```

4. Patch the istio ingress gateway deployment as shown below.

> **Note:** Please update the **<INSTALLATION_PATH>** placeholder before executing patch command below.

```
kubectl patch deployment istio-ingressgateway -n istio-system \
 -p '{
   "spec": {
     "template": {
       "spec": {
         "volumes": [
           {
             "name": "istio-gateway-access-logs",
             "hostPath": {
               "path": "<INSTALLATION_PATH>/logs/istio-logs",
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
            "type": "DirectoryOrCreate"
          }
        }
      ],
      "containers": [
        {
          "name": "istio-proxy",
          "volumeMounts": [
            {
              "mountPath": "/var/log",
              "name": "istio-gateway-access-logs"
            }
          ]
        }
      ]
    }
  }
}'
```

5. Update ownership of the istio-logs directory and configure the correct installation user

```
cd <INSTALLATION_PATH>/logs
```

```
sudo chown -R appviewx:appviewx istio-logs/
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
appviewx@pe-devops-apvx-n75:~/appviewx/logs/istio-logs$ pwd
/home/appviewx/appviewx/logs/istio-logs
appviewx@pe-devops-apvx-n75:~/appviewx/logs/istio-logs$ kubectl patch deployment istio-ingressgateway -n istio-system \
  -p '{
    "spec": {
      "template": {
        "spec": {
          "volumes": [
            {
              "name": "istio-gateway-access-logs",
              "hostPath": {
                "path": "/home/appviewx/appviewx/logs/istio-logs",
                "type": "DirectoryOrCreate"
              }
            }
          ],
          "containers": [
            {
              "name": "istio-proxy",
              "volumeMounts": [
                {
                  "mountPath": "/var/log",
                  "name": "istio-gateway-access-logs"
                }
              ]
            }
          ]
        }
      }
    }
  }'
deployment.apps/istio-ingressgateway patched
appviewx@pe-devops-apvx-n75:~/appviewx/logs/istio-logs$
```

6. Replace the **logrotate_configmap.yaml** under <appviewx_kubernetes_path>/yaml.

```
cd <appviewx_kubernetes_path>/appviewx_logrotate/logrotate_setup/chart/templates
```

```
logrotate_configmap.yaml:


{{- $root := . -}}
{{- $dot := . }}
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: logrotate-config
  namespace: {{ $dot.Values.appviewx_logrotate.namespace }}
data:
  mongodb.conf: |
    /appviewx/dependencies/logs/mongodb-*.log {
        rotate 5
        size 200M
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
        compress
        missingok
        notifempty
        copytruncate
        sharedscripts
        postrotate
          /bin/bash /appviewx/dependencies/properties/mongodb_logrotate.sh $(find /appviewx/dependencies/logs/
-iname "mongo*db*.log" -exec stat --format='%Y %n' "{}" + | sort -nr | head -n1 | awk '{gsub(/\.log$/, "", $2); print $2}' | xargs
basename)
        endscript
    }
 heapfile.conf: |
   /appviewx/dependencies/logs/*.hprof {
        rotate 5
        missingok
        notifempty
    }
 gc.conf: |
   /appviewx/dependencies/logs/*.gc {
        rotate 5
        missingok
        notifempty
        copytruncate
        compress
    }
 istiologs.conf: |
   /appviewx/dependencies/logs/istio-logs/*.log {
        rotate 5
        missingok
        notifempty
        copytruncate
        compress
    }
---
apiVersion: v1
kind: ConfigMap
metadata:
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
  name: mongo-logrotate
  namespace: {{ $dot.Values.appviewx_logrotate.namespace }}
data:
  mongodb_logrotate.sh: |-
{{ $root.Files.Get $dot.Values.appviewx_logrotate.script_mongo_local_path | indent 4 }}
```

7. Uninstall and reinstall the **appviewx-logrotate-sa** cronjob by the following commands:

```
cd <appviewx_kubernetes_path>/appviewx_logrotate/
```

```
helm uninstall appviewx-logrotate-sa
```

```
./run.sh
```

Working with Plugins

- Adding a New Plugin
- Removing a Plugin
- Restarting a Plugin
- Scaling a Plugin
- Changing the Memory for a Plugin

Adding a New Plugin

During the AppViewX installation, the user may not enable all the plugins that are required. Therefore, the user can enable those plugins after the AppViewX installation.

To enable a plugin after installation:

1. Navigate to the /home/appviewx/appviewx_kubernetes/scripts directory.
2. Open the **appviewx.conf** file.
3. Modify the **ENABLED_PLUGINS** as new plugins that need to be installed.

> **Warning:** It is not recommended to delete the appviewx_dependencies in the ENABLED_PLUGINS value. For example, ENABLED_PLUGINS= avx_dependencies,avx_vendors.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
ENABLED_PLUGINS=appviewx_dependencies,avx_platform_amc,avx_platform_gateway

SSH_OTHER_USER=appviewx

avx_platform_amc=dc1,dc2
avx_config_server=dc1,dc2
```

4. Enter the data center value in which the plugin needs to be installed.

   For example, **avx_vendors=dc1**.

```
-bash-4.2$ kubectl get pods -A | grep amc
dc1                  avx-platform-amc-68b9fbc7f-fj7wr                  2/2     Running     1          2d2h
dc2                  avx-platform-amc-68b9fbc7f-kv8k8                  2/2     Running     2          2d2h
-bash-4.2$
```

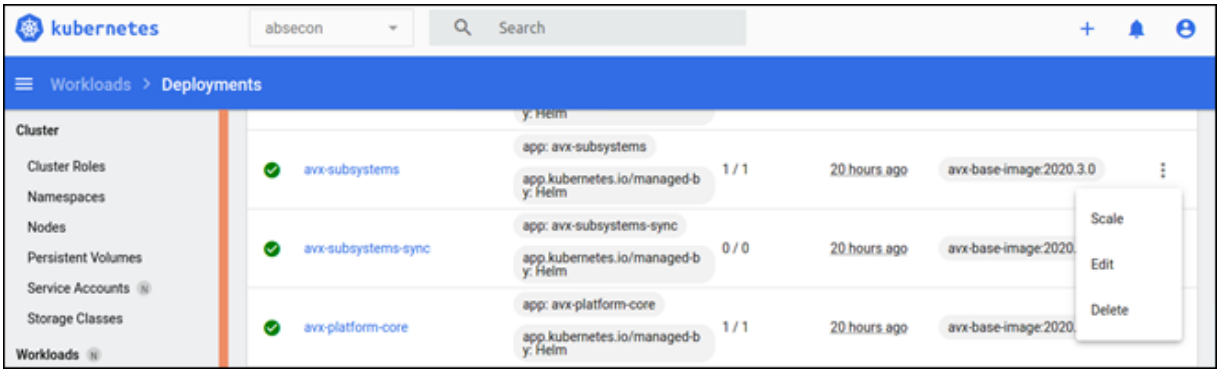
5. Save and exit the **appviewx.conf** file.
6. Navigate to the **scripts** directory.
7. In the scripts directory, execute the following command:

```
script plugins_install.sh
```

Removing a Plugin

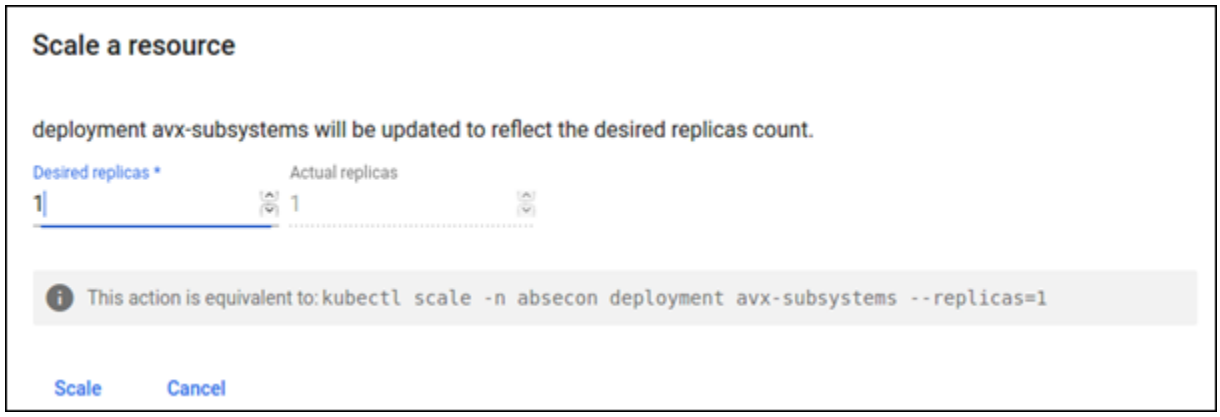
To remove a plugin for maintenance purposes:

1. Log in into the kubernetes management console.
2. From the top list, select the required namespace or datacenter.
3. From the left pane, click **Deployments**.
4. Search for the specific deployment/plugin that needs to be stopped.
5. Against the name of the pod, click the three dots and select **Scale**.



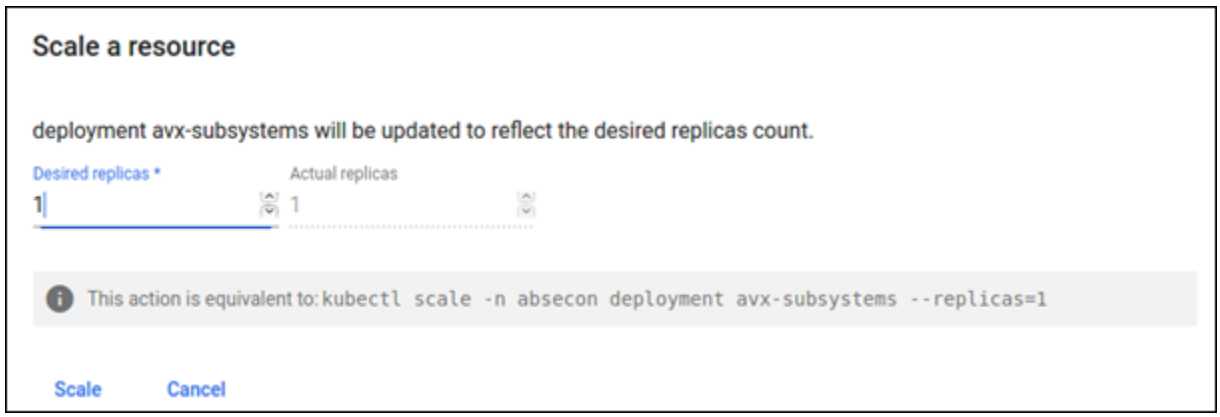
The **Scale a Resource** page is displayed.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



6. Set the value for **Desired replicas** to 0.

   This will delete all the pods and does not spin any new pod for that plugin.

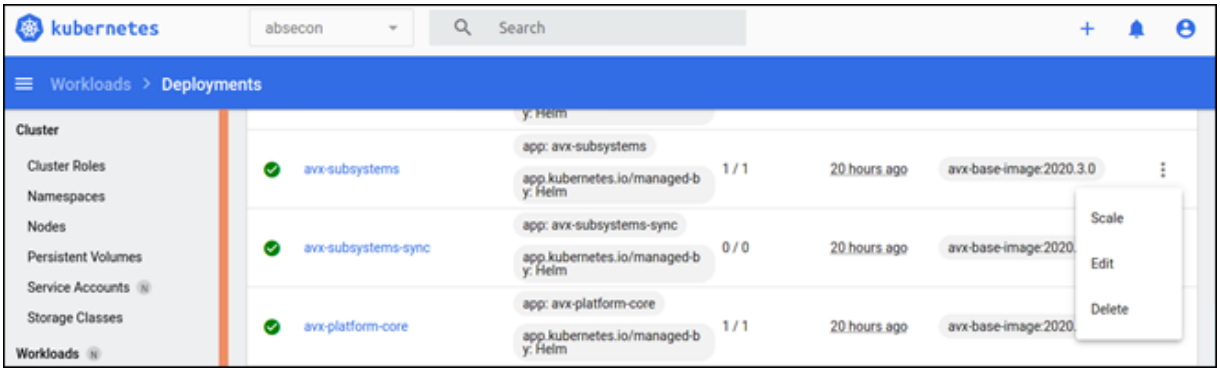


Restarting a Plugin

1. Log in into the kubernetes management console.
2. From the top list, select the required namespace or datacenter.
3. From the left pane, click **Deployments**.
4. Search for the specific deployment/plugin that needs to be restarted.
5. Against the name of the pod, click the three dots and select **Delete**.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

This will stop the current pod and create a new pod.

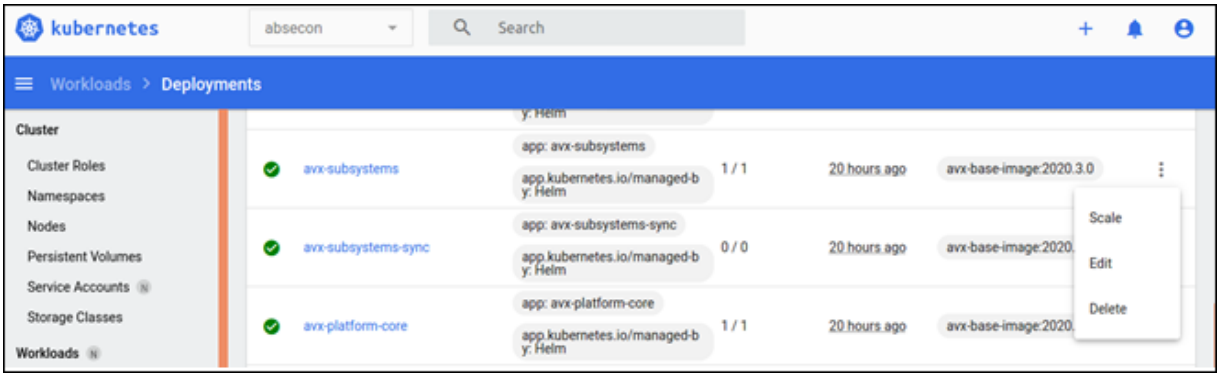


Scaling a Plugin

Scale refers to an increase or decrease in the number of plugins manually. You have an option to scale it from the Kubernetes management console.

To increase/decrease the number of plugins of a specific type:

1. Log in into the kubernetes management console.
2. From the top list, select the required namespace or datacenter.
3. From the left pane, click **Deployments**.
4. Search for the specific deployment/plugin that needs to be scaled.
5. Against the name of the pod, click the three dots and select **Scale**.



The **Scale a Resource** page is displayed.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



6. Update the value of the **Desired replicas** parameter to increase or decrease the number of pods for a plugin.
7. Click **Scale**.

Changing the Memory for a Plugin

Every plugin inside the node runs on a dedicated memory. It can be adjusted to the maximum and minimum memory that a pod can use.

To increase or decrease the plugins memory:

1. Log in to the Kubernetes dashboard of AppViewX.
2. From the left pane, under **Workloads**, click **Deployments**.
3. Search for the respective deployment to modify it.
4. Click **Edit**.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

5. Modify the xmx and xms values to the required values as shown below.



Working with the Management Console

The management console allows you to monitor, maintain, and manage the application as well as the performance. The console provides a graphical interface to view and monitor the application instance.

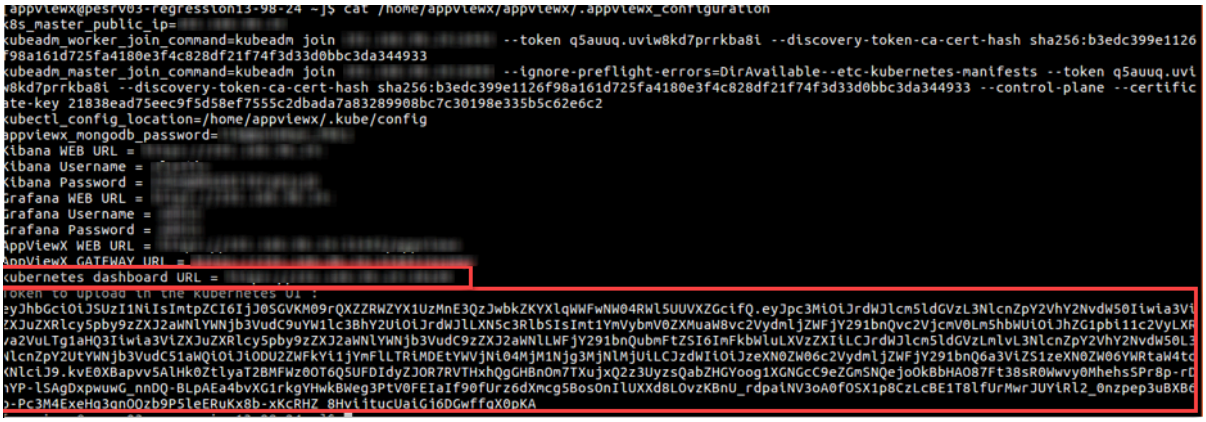
- Accessing the Management Console
- Viewing the POD Status
- Accessing the POD Console
- Accessing the Database Command Line
- Exporting a Database Collection

Accessing the Management Console

1. Navigate to the installation directory.
2. Open the **.appviewx_configuration** file.

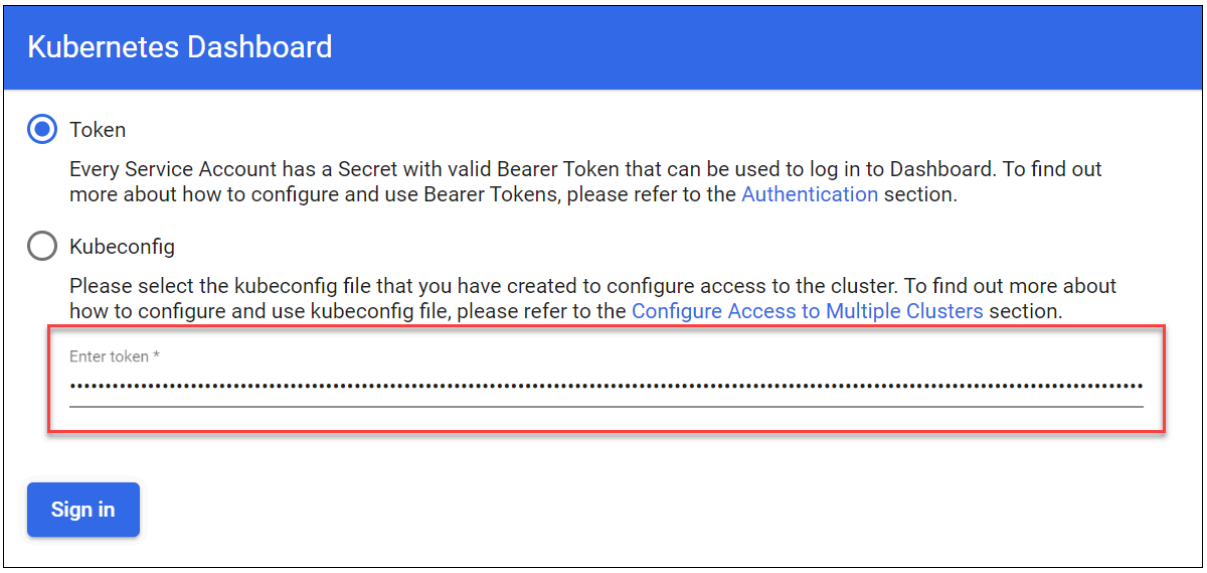
   For example: cat/home/appviewx/appviewx/.appviewx_configuration
3. Search for Kubernetes dashboard URL and copy the URL to a web browser.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



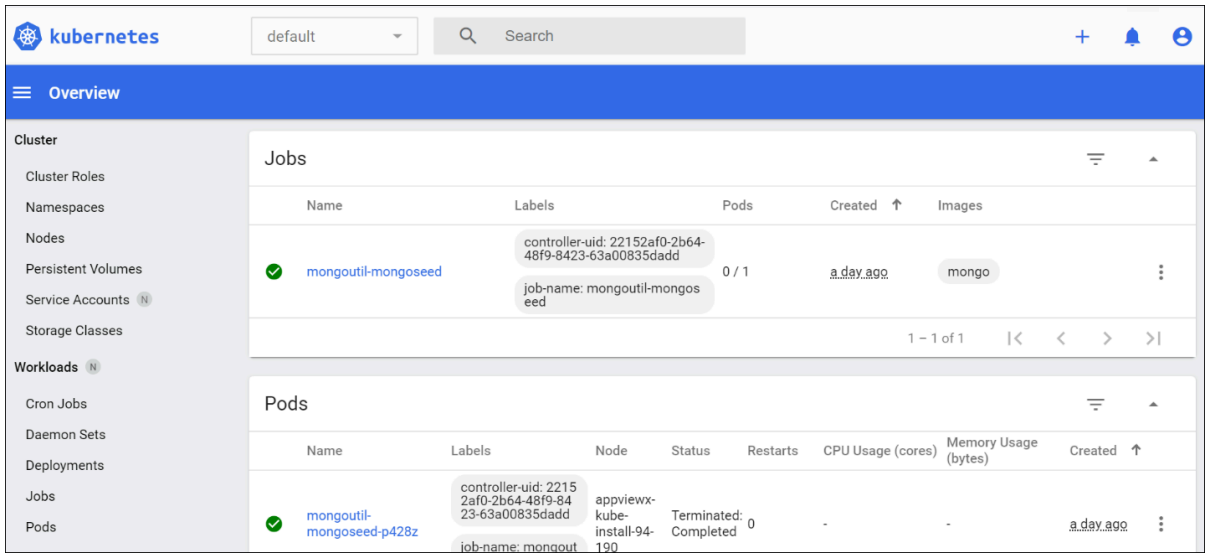
4. On the **Kubernetes Dashboard** login screen, enter the token value
5. Click **Sign In**.

> **Note:** You can find the token value from the **.appviewx_configuration** file.



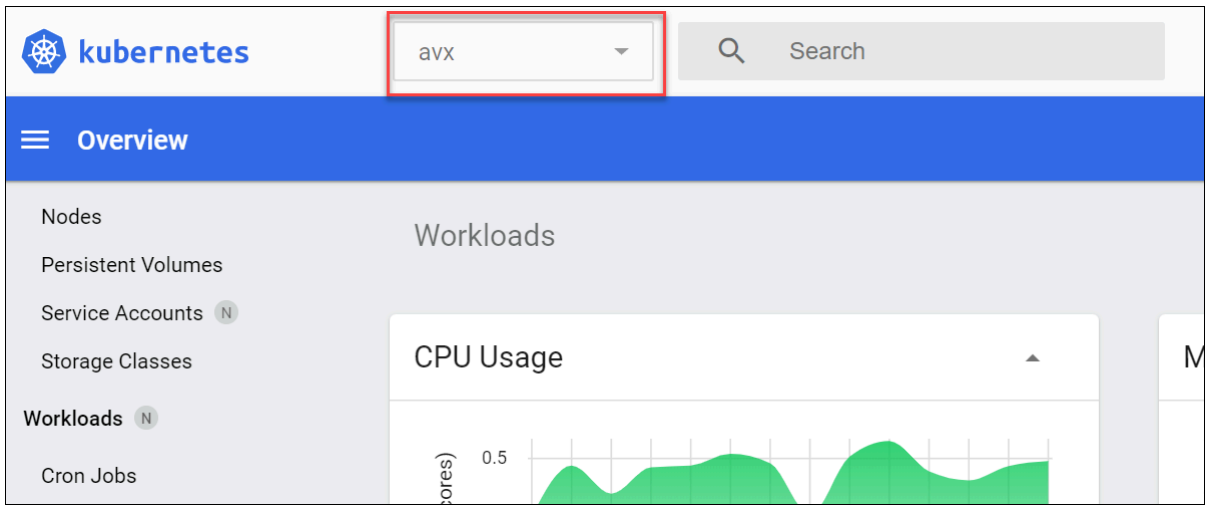
Copyright © 2025 AppViewX Inc. All Rights Reserved.

After you log in, you can access the Kubernetes management console and manage AppViewX components.



Viewing the POD Status

1. Open the Kubernetes management console.
2. Select a namespace from the top list.
3. Select **Pods** on the left menu.

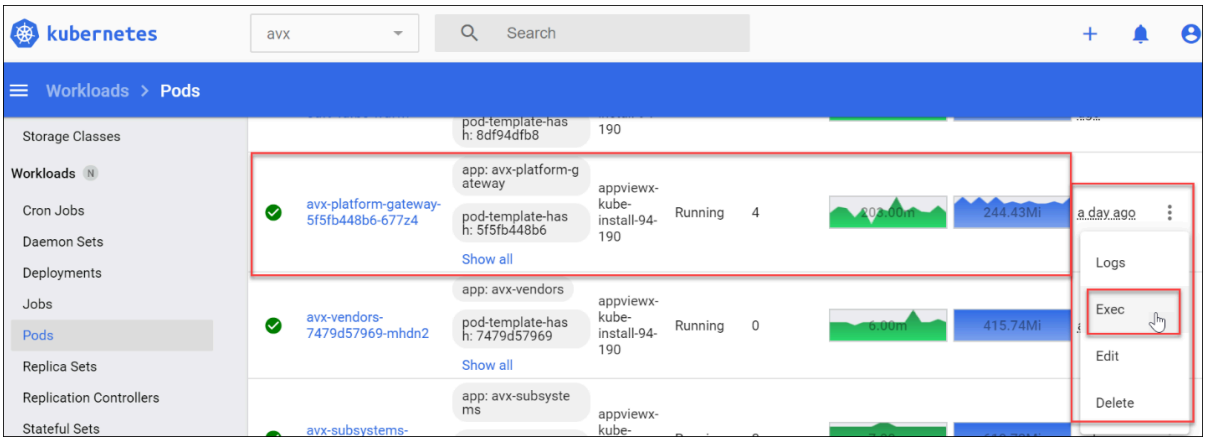


Copyright © 2025 AppViewX Inc. All Rights Reserved.

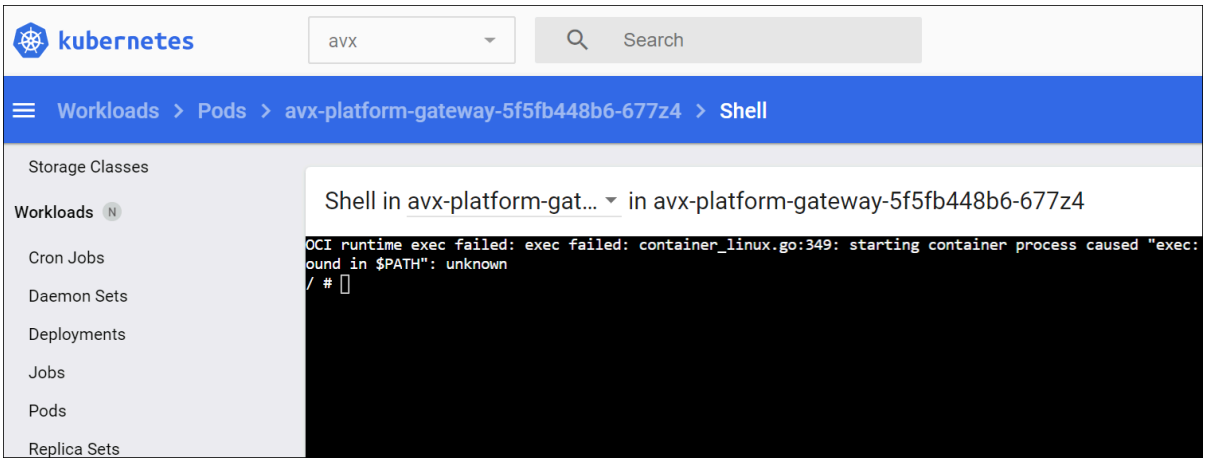
Accessing the POD Console

1. Open the Kubernetes management console.
2. Select the required namespace.
3. Under **Workloads**, click **Pods**.

   The **Pods** page is displayed.
4. Click on the three dots next to the pod and select **Exec**.



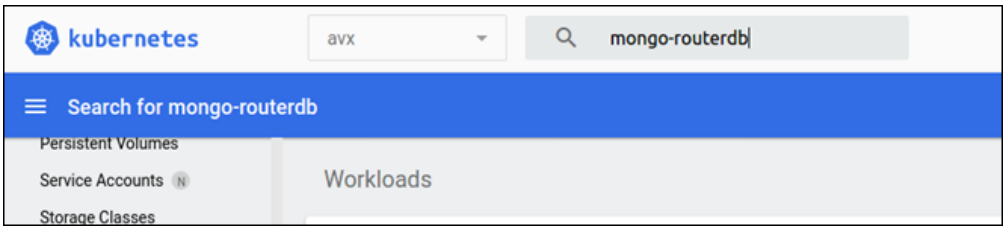
The Pod command line shell is displayed.



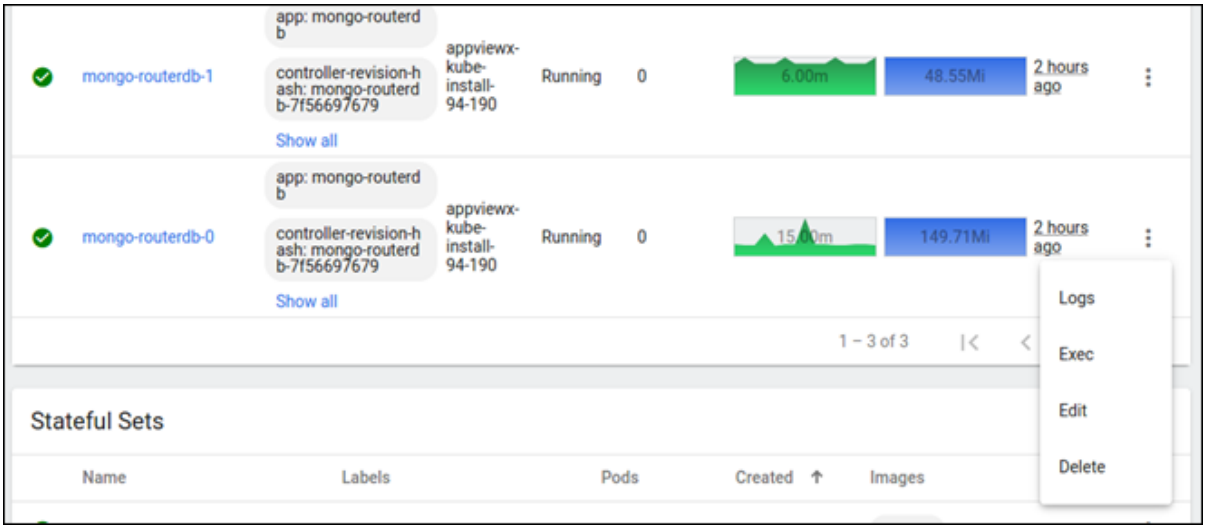
Accessing the Database Command Line

1. Open the Kubernetes management console.
2. Select **avx** in the namespace.
3. Search for **mongo-routerdb**.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



4. Click on the three dots next to mongo-routerdb-0 pod and select **Exec**.



5. To launch the mongo db prompt, execute the following command: `<mongo>`
6. Execute the following command:

```
<use admin>
```

7. Execute the following command:

```
db.auth("admin","<mongodbpasssword>")
```

> **Note:** The password can be taken from the value of the `appviewx_mongodb_password` variable from the `<INSTALLATION_PATH>/.appviewx_configuration` file.



Copyright © 2025 AppViewX Inc. All Rights Reserved.

Exporting a Database Collection

Collections serve as generic repositories that hold any data in key-value pair format. It acts as interfaces to enter and modify data into the AppViewX Mongo database. Data from collections is consumed as a part of the provisioning request process or by any other scripts that are triggered by AppViewX. The structure of the collections is based on the Mongo database.

To export a mongo database collection:

1. Login to Kubernetes dashboard UI with the token.
2. From the top section, select the **avx** namespace.
3. Click pods and search for **mongo-routerdb-0**.
4. Click on the three dots icon and select **Exec**.
5. To navigate to the logs directory, execute the following command:

   ```
   cd /appviewx/dependencies/logs
   ```

6. Check if export_collection directory is available. Otherwise, to create the directory, execute the following command:

   ```
   mkdir export_collection
   ```

7. To navigate to the export_collection directory, execute the following command:

   ```
   cd /appviewx/dependencies/logs/export_collection
   ```

8. To export the database collection, execute the following command: Change the fields highlighted in bold with the desired values according to your setup.

   ```
   mongoexport --username admin --password <password> --db=appviewx --collection=<collectionName> --out=<fileName>.json --authenticationDatabase admin
   ```

   This command will export the collection and the file will be available at the following location: /appviewx/dependencies/logs/export_collection

   Note: The exported file is also available at the following location on the host where the mongodb pod is running: INSTALLATION_PATH/appviewx/logs/export_collection

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Applying Custom Pod Configurations

Objective

The objective of this chapter is to implement custom changes related to pod specifications and prevent any issues with overridden changes after upgrades.

Feature Specifications and Commands

The features and the respective commands are as follows:

1. Update pod affinity

```
./appviewx.sh --apply-affinities
```

2. To update node labels.

```
./appviewx.sh --apply-labels
```

3. Update memory allocations for particular application pod.

```
./appviewx.sh --apply-allocate-memory
```

4. Change the pipeline worker count for logstash conf

```
./appviewx.sh --apply-logstashCustoms
```

5. Change the replica count

```
./appviewx.sh --apply-replicas
```

6. Change the HPA value of deployment

```
./appviewx.sh --apply-hpa
```

Steps to Apply Custom Changes

1. Navigate to <installer>/scripts location.
2. Copy the **custom_changes.yaml.template** to **custom_changes.yaml**
3. Find the formatting below for all custom configurations.
   a. Pod affinities

```
affinities:
  <Plugin Name>:
    nodeAffinity:
      enable: false
      type: "required"
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
      key: "dummy"
      values:
        - "dummy"
    podAffinity:
      enable: false
      key: "app"
      values:
        - "dummy"
      namespaces:
        - "avx"
      topologyKey: "kubernetes.io/hostname"
    podAntiAffinity:
      enable: false
      key: "app"
      values:
        - "dummy"
      topologyKey: "kubernetes.io/hostname"
```

Note: All three types of affinities should be mentioned during the configuration. It is necessary to set the enable field to *true* or *false* according to use case.

b. Node labels

```
node_labels:
  <node name>:
    ingress: "true"
```

c. Pod memory allocation

```
memoryAllocations:
    avx_subsystems:
      xms: "1g"
      xmx: "2g"
      memoryRequest: "100Mi"
      cpuRequest:  "100"
```

Note: xms and **xmx** fields are mandatory in case of custom memory allocations. After applying changes on the plugin you must re-deploy the same plugin in your setup.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

d. Pipeline worker count for logstash conf

```
logstash_customs:
    pipelineWorkerCount: "5"
```

e. Replica count

```
replication:
    <pod_name>:
      count: "2"
```

f. HPA value of deployment

```
horizontalPodAutoScalling:
    avx_subsystems_sync:
      maxReplicas: "3"
      minReplicas: "1"
      cpu: "300"
      memory: "300"
```

Note: memory for HPA is optional parameter, you may skip it during configuration.

At the time modifying the custom_changes.yaml file for custom configurations, the rules of yaml code should be intact.

1. Before you start writing the custom configurations, ensure the three dashes “---” are present on the top of very first line of the custom configuration.
2. Field and its scope must be accurate.

Post the changes in **custom_changes.yaml**, installing the plugins will apply all custom changes.

Remote Transfer of AppViewX Backups

This section outlines the steps to set up SFTP backups, modify the SFTP server settings post-installation, and decrypt and restore backups.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Setting Up SFTP Backups

Configuring SFTP in the appviewx.conf

1. Enable the SFTP transfer by setting the SFTP_TRANSFER option to TRUE in the **appviewx.conf** file
2. Specify the SFTP server details as follows:
   a. For the REMOTE_BACKUP_SERVER parameter, enter the IP address of the SFTP backup server.
   b. For the REMOTE_BACKUP_SERVER_SSH_PORT parameter, specify the SSH port used by the remote server.
   c. For the REMOTE_BACKUP_ABSOLUTE_PATH parameter, provide the path where backups should be stored on the remote server.
3. Choose the authentication method for the REMOTE_BACKUP_SERVER_AUTHENTICATION_METHOD parameter, elect the authentication method. Options include '*rsa'*, '*password'*, or '*passwordless*'.

Authentication Methods

Based on the authentication method selected, you will prompted to enter details as follows:

1. **RSA-Based Authentication:**
   - If REMOTE_BACKUP_SERVER_AUTHENTICATION_METHOD is set to 'rsa', you will be prompted to enter only the username during installation.
   - REMOTE_BACKUP_SERVER_AUTHENTICATION: Specify the file path of the private key used to connect to the remote server.
2. **Password-Based Authentication:** If REMOTE_BACKUP_SERVER_AUTHENTICATION_METHOD is set to 'password', you will be prompted to enter the username and password for the remote server during the installation process.
3. **Passwordless Authentication:**
   - If REMOTE_BACKUP_SERVER_AUTHENTICATION_METHOD is set to 'passwordless', only the username will be prompted during installation.
   - Ensure the private key is stored as id_rsa under ~/.ssh/ on each server listed in MONGODB_HOST in the appviewx.conf file and chmod 400 id_rsa.

Modifying SFTP Server Settings Post-Installation

1. Use the AppViewX utility to modify SFTP configurations.
2. Update the appviewx.conf file by making the necessary changes in it.
3. To run the SFTP setup command

Copyright © 2025 AppViewX Inc. All Rights Reserved.

a. Navigate to the scripts directory:

```
cd <INSTALLER_DIRECTORY>/scripts
```

b. Trigger the SFTP setup:

```
./appviewx.sh --setup_sftp
```

The utility will prompt you for inputs based on the chosen authentication method, similar to the initial installation process.

Decrypting and Restoring Backups

Decrypting the Encrypted Mongo Backup File

1. Navigate to the location where the Mongo backups are stored on the server.
2. Run the following command to extract the backup and key files:

```
tar -xzf <mongo_backup>.tar.gz
```

3. Copy the private key generated during the installation or **setup_sftp** utility from the **.appviewx_configuration** directory to a file named **backup_key.pem**.
4. Use the following command to decrypt the AES key:

```
openssl pkeyutl -decrypt -inkey backup_key.pem -in <aes_encrypted_key>.bin -out decrypted_aes_key.bin
```

5. Decrypt the Mongo backup file using the decrypted AES key:

```
openssl enc -d -aes-256-cbc -pbkdf2 -in <mongo_backup>.bin -out <mongo_backup>.tar.gz -pass file:decrypted_aes_key.bin
```



Note: The file name specified with -out should follow the pattern: the file name of the **.bin** file up to UTC_2024 (the year), followed by .tar.gz.

Restoring the Mongo Backup

1. Navigate to the installer scripts directory.

```
cd INSTALLER_PATH/scripts
```

2. Execute the Mongo restore script with the path to the decrypted backup file:

```
./mongo_restore.sh <DECRYPTED_BACKUP_FILE_PATH>
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Decrypting the Encrypted Vault Backup File

1. Navigate to the location where the Vault backups are stored on the server.
2. Run the following command to extract the backup and key files:

```
tar -xzf <vault_backup>.tar.gz
```

3. Copy the private key generated during the installation or **setup_sftp** utility from the **.appviewx_configuration** directory to a file named **backup_key.pem**.
4. Use the following command to decrypt the AES key:

```
openssl pkeyutl -decrypt -inkey backup_key.pem -in <aes_encrypted>.bin -out decrypted_aes_key.bin
```

5. Decrypt the Vault backup file using the decrypted AES key:

```
openssl enc -d -aes-256-cbc -pbkdf2 -in <vault_backup>.bin -out vault_backup.txt -pass file:decrypted_aes_key.bin
```

Restoring the Vault Backup

1. Navigate to the installer scripts directory.

```
cd INSTALLER_PATH/scripts
```

2. Execute the Vault restore script with the path to the decrypted backup file:

```
./vault_restore.sh <DECRYPTED_BACKUP_FILE_PATH>
```

External Certificate for Kubernetes

The section describes the steps to update the external certificate authority for the Kubernetes kubeadm. It contains the certificate specifications for the different certificates to be generated since .p12 is the only file format that is supported.

- Certificate Specifications
- Entering All Certificates in the appviewx.conf File
- Rollback Steps For Failure in Certificate Updates

Certificate Specifications

Certificates must be generated individually for each of the common names listed in the table below. All master nodes (IP address and hostname) listed in the table must be added in the SAN of the certificates for a multi-node environment.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Table - Common name and IP address

Common Name	Type	O (in Subject)	SAN (refer notes below)	Parent CA	Cert and Location
kube-etcd	server	-	<master_hostnames>, <master_Host_IPs>, <kube_api_address>, localhost, 127.0.0.1,<service_ip>	etcd-ca	etcd/ server.crt,etcd/ server.key
kube-etcd-peer	server	-	<master_hostnames>, <master_Host_IPs>, <kube_api_address>, localhost, 127.0.0.1, <service_ip>	etcd-ca	etcd/ peer.crt,etcd/ peer.key
kube-etcd-healthcheck-client	client	-	-	etcd-ca	etcd/ healthcheck-client.crt,etcd/ healthcheck-client.key
kube-apiserver-etcd-client	client	system:masters	-	etcd-ca	pki/apiserver-etcd-client.key, pki/apiserver-etcd-client.crt
kube-apiserver	server	-	<master_hostnames>, <master_Host_IPs>, <kube_api_address>, localhost, 127.0.0.1,<service_ip>, kubernetes, kubernetes.default, kubernetes.default.svc, kubernetes.default.svc.cluster, kubernetes.default.svc.cluster.local	kubernetes-ca	pki/ apiserver.key, pki/ apiserver.crt

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Table - Common name and IP address (continued)

Common Name	Type	O (in Subject)	SAN (refer notes below)	Parent CA	Cert and Location
kube-apiserver-kubelet-client	client	system:masters	-	kubernetes-ca	pki/apiserver-kubelet-client.key, pki/apiserver-kubelet-client.crt
front-proxy-client	client	-	-	kubernetes-front-proxy-ca	client.key,pki/front-proxy-client.crt
kubernetes-admin	client	system:masters	-	kubernetes-ca	admin.crt, admin.key
system:kube-controller-manager	client	-	-	kubernetes-ca	controller-manager.crt, controller-manager.key
system:kube-scheduler	client	-	-	kubernetes-ca	scheduler.crt, scheduler.ke
system:node:<hostname>	client	system:nodes	-	kubernetes-ca	kubelet.crt, kubelet.key

SAN values are as follows:

<master_hostnames> Hostname of the master.

<master_Host_IPs> IPs of the master.

<service_ip> The service IP can be obtained by executing the `kubectl get svc` from any node.

```
[appviewx@pe-iu-node18 scripts]$ kubectl get svc
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   23h
[appviewx@pe-iu-node18 scripts]$
```

<kube_api_address> Load balancer for kube apiserver (if configured).

Copyright © 2025 AppViewX Inc. All Rights Reserved.

system:node:<hostname>The *kubelet.crt* should be generated for all the servers (master and worker) separately. For example, if the setup consists of three nodes, the certificates must be generated for all three nodes. The value of the **<hostname>** should be entered from the output of the `hostname` command.

Note:

- Enter only the `hostname` command output in SAN; do not enter the `hostname -f` command output.
- A general rule for the SAN of the certificate is to add the IP address in the *IP Address* field and the hostnames in the *DNS* field.
- An example of a wrong entry is shown below:



Entering All Certificates in the appviewx.conf File

1. Navigate to the <appviewx_installed_location>/appviewx_kubernetes/scripts and open *appviewx.conf* file.
2. To enable the external CA for kubeadm, set the value KUBE_EXTERNAL_CERT=TRUE

```
#Manage kubernetes with the external certficates
#Replace /home/appviewx/external_p12_multinode with the absolute path of the certificate file
#Certificate should be in .p12 format
#Follow the the guide for the certificate speficifications
#Execute ./appviewx.sh --password-encrypt command to encrypt the CERT_PASSWORD password

KUBE_EXTERNAL_CERT=TRUE
```

3. Enter the encrypted certificate password in the `CERT_PASSWORD` key. To encrypt the password,

   a. Navigate to <appviewx_installer_location>/appviewx_kubernetes/scripts

   b. Execute the command

```
./appviewx.sh --password-encrypt
```

```
#Manage kubernetes with the external certficates
#Replace /home/appviewx/external_p12_multinode with the absolute path of the certificate file
#Certificate should be in .p12 format
#Follow the the guide for the certificate speficifications
#Execute ./appviewx.sh --password-encrypt command to encrypt the CERT_PASSWORD password

KUBE_EXTERNAL_CERT=TRUE
CERT_PASSWORD=vault:v1:XJufPkIER2fdnAYt6bZEEZHhq66r7VPGhNw7AhZ/UQPGaBNNs5WJNg==
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

4. Enter the absolute path of the certificate which is generated for the common name **kube-etcd** in `KUBE_ETCD_PATH`
5. Enter the absolute path of the certificate which is generated for the common name **kube-etcd-peer** in `KUBE_ETCD_PEER_PATH`
6. Enter the absolute path of the certificate which is generated for the common name **kube-etcd-healthcheck-client** in `KUBE_ETCD_HEALTHCHECK_CLIENT_PATH`
7. Enter the absolute path of the certificate which is generated for the common name **kube-apiserver-etcd-client** in `KUBE_APISERVER_ETCD_CLIENT_PATH`
8. Enter the absolute path of the certificate which is generated for the common name **kube-apiserver** in `KUBE_APISERVER_PATH`
9. Enter the absolute path of the certificate which is generated for the common name **kube-apiserver-kubelet-client** in `KUBE_APISERVER_KUBELET_CLIENT_PATH`
10. Enter the absolute path of the certificate which is generated for the common name **front-proxy-client** in `FRONT_PROXY_CLIENT_PATH`
11. Enter the absolute path of the certificate which is generated for the common name **kubernetes-admin** in `KUBERNETES_ADMIN_PATH`
12. Enter the absolute path of the certificate which is generated for the common name **system:kube-controller-manager** in `KUBE_CONTROLLER_MANAGER_PATH`
13. Enter the absolute path of the certificate which is generated for the common name **system:kube-scheduler** in `KUBE_SCHEDULER_PATH`

```
KUBE_ETCD_PATH=/home/appviewx/external_p12_multinode/kube-etcd_17_BA_FA_51_75_3A_CE_0D_E5_86_9B_20_A5_5A_4D_14_00_35_89_DD.p12
KUBE_ETCD_PEER_PATH=/home/appviewx/external_p12_multinode/kube-etcd-peer_51_A3_CE_5F_51_35_9A_72_3C_15_1B_54_BE_83_5C_25_ED_94_CB_C4.p12
KUBE_ETCD_HEALTHCHECK_CLIENT_PATH=/home/appviewx/external_p12_multinode/kube-etcd-healthcheck-client_31_54_F6_E1_3E_68_AB_B1_65_EC_02_99_E2_FB_A9_A7_5D_0C_D5_D3.p12
KUBE_APISERVER_ETCD_CLIENT_PATH=/home/appviewx/external_p12_multinode/kube-apiserver-etcd-client_27_FC_1E_94_84_0A_A8_90_D8_5D_99_5F_98_BB_B9_10_BF_E8_B5_4A.p12
KUBE_APISERVER_PATH=/home/appviewx/external_p12_multinode/kube-apiserver_19_33_6A_BE_B7_5E_F0_90_E6_2A_A8_F8_5D_C3_A0_2C_2A_78_BD_D1.p12
KUBE_APISERVER_KUBELET_CLIENT_PATH=/home/appviewx/external_p12_multinode/kube-apiserver-kubelet-client_7D_5F_B2_78_2C_51_03_D1_39_17_BF_FD_26_6E_A2_1A_60_93_1C_BF.p12
FRONT_PROXY_CLIENT_PATH=/home/appviewx/external_p12_multinode/front-proxy-client_61_97_2B_D9_E8_13_2B_24_3F_7E_85_B3_1A_F9_3A_AF_10_4C_5F_45.p12
KUBERNETES_ADMIN_PATH=/home/appviewx/external_p12_multinode/kubernetes-admin_2D_A0_1B_5E_A0_CF_27_2E_6B_9C_34_02_D9_E0_CA_60_95_BD_92_E0.p12
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
KUBE_CONTROLLER_MANAGER_PATH=/home/appviewx/external_p12_multinode/system_kube-controller-manager_31_32_15_2E_5F_4A_9C_B9_0E_2A_11_9B_CE_15_AA_59_5D_B7_FC_D1.p12
KUBE_SCHEDULER_PATH=/home/appviewx/external_p12_multinode/system_kube-scheduler_6A_FF_10_E1_F1_C9_9F_3C_0F_9D_82_88_18_38_EB_01_FB_3D_02_70.p12
```

14. Enter the **Kubelet certificates** in a colon ':' separated format, such as *<hostname>:<kubelet_certificate.p12>*. There should not be any spaces and also no colon (:) in the certificate file name.

Note:

a. If the kubelet certificate is generated for the host **pe-iu-node20.lab.appviewx.net**, the entry should be in the format **KUBELET_CERT_PATH=<hostname>:<absolute certificate file path>**. The entry for the host would be `KUBELET_CERT_PATH=pe-iu-node20.lab.appviewx.net:/home/appviewx/external_p12_multinode/system_node_pe-iu-node20.lab.appviewx.net.p12`

b. Enter all certificates that match the hosts in a comma-separated format, as given in the example below:

```
KUBELET_CERT_PATH=pe-iu-node20.lab.appviewx.net:/home/appviewx/external_p12_multinode/system_node_pe-iu-node20.lab.appviewx.net.p12,pe-iu-node16.lab.appviewx.net:/home/appviewx/external_p12_multinode/system_node_pe-iu-node16.lab.appviewx.net.p12,pe-iu-node17.lab.appviewx.net:/home/appviewx/external_p12_multinode/system_node_pe-iu-node17.lab.appviewx.net.p12,pe-iu-node18.lab.appviewx.net:/home/appviewx/external_p12_multinode/system_node_pe-iu-node18.lab.appviewx.net.p12,pe-iu-node19.lab.appviewx.net:/home/appviewx/external_p12_multinode/system_node_pe-iu-node19.lab.appviewx.net.p12,pe-iu-node20.lab.appviewx.net:/home/appviewx/external_p12_multinode/system_node_pe-iu-node20.lab.appviewx.net.p12,pe-iu-node21.lab.appviewx.net:/home/appviewx/external_p12_multinode/system_node_pe-iu-node21.lab.appviewx.net.p12
```

Warning: Entering wrong certificates in the paths mentioned above will compromise the functioning of the application.

15. After adding all the certificate entries in the *appviewx.conf*
    a. Navigate to the **<appviewx_installer_location>/appviewx_kubernetes/scripts**
    b. Execute the command `./appviewx.sh --enable-kube-external-ca`

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
[appviewx@pe-iu-node18 scripts]$ ./appviewx.sh --enable-kube-external-ca
MAC verified OK
MAC verified OK
MAC verified OK
MAC verified OK
MAC verified OK
MAC verified OK
MAC verified OK
MAC verified OK
MAC verified OK
MAC verified OK
MAC verified OK
MAC verified OK
MAC verified OK
```

16. The command prompt for the passwords of all the nodes once the validations are completed. Enter the passwords, and hit the keyboard **Enter** key to proceed further.

```
node/pe-iu-node20.lab.appviewx.net drained
NAME                            STATUS                     ROLES                  AGE   VERSION
pe-iu-node16.lab.appviewx.net   Ready,SchedulingDisabled   <none>                 25h   v1.20.7
pe-iu-node17.lab.appviewx.net   Ready,SchedulingDisabled   <none>                 25h   v1.20.7
pe-iu-node18.lab.appviewx.net   Ready,SchedulingDisabled   <none>                 25h   v1.20.7
pe-iu-node19.lab.appviewx.net   Ready,SchedulingDisabled   control-plane,master   25h   v1.20.7
pe-iu-node20.lab.appviewx.net   Ready,SchedulingDisabled   control-plane,master   25h   v1.20.7
pe-iu-node21.lab.appviewx.net   Ready,SchedulingDisabled   control-plane,master   25h   v1.20.7
/home/appviewx/FP6/appviewx_kubernetes/scripts/script_util
Please enter appviewx password of master:pe-iu-node20.lab.appviewx.net :
Please enter appviewx password of master:pe-iu-node21.lab.appviewx.net :
Please enter appviewx password of master:pe-iu-node19.lab.appviewx.net :
Please enter appviewx password of absecon:pe-iu-node16.lab.appviewx.net :
Please enter appviewx password of antartica:pe-iu-node17.lab.appviewx.net :
Please enter appviewx password of antartica:pe-iu-node18.lab.appviewx.net :
null_resource.ssh_connectivity[5]: Creating...
null_resource.ssh_connectivity[3]: Creating...
null_resource.ssh_connectivity[2]: Creating...
null_resource.ssh_connectivity[4]: Creating...
null_resource.ssh_connectivity[1]: Creating...
null_resource.ssh_connectivity[0]: Creating...
null_resource.ssh_connectivity[3]: Provisioning with 'remote-exec'...
null_resource.ssh_connectivity[2]: Provisioning with 'remote-exec'...
```

The following message is displayed on the successful completion of the execution:

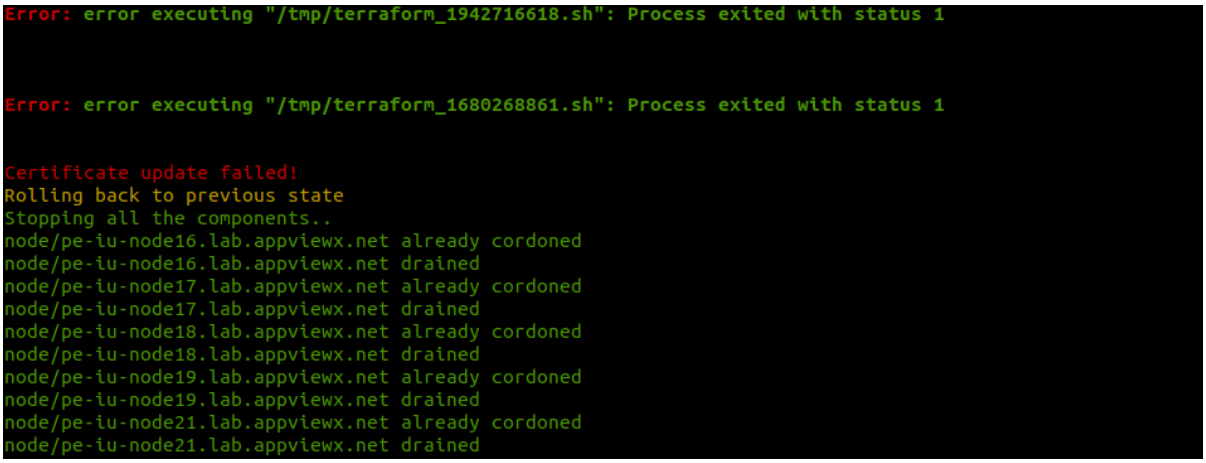
```
Starting all the components..
node/pe-iu-node20.lab.appviewx.net uncordoned
node/pe-iu-node19.lab.appviewx.net uncordoned
node/pe-iu-node21.lab.appviewx.net uncordoned
node/pe-iu-node16.lab.appviewx.net uncordoned
node/pe-iu-node17.lab.appviewx.net uncordoned
node/pe-iu-node18.lab.appviewx.net uncordoned
NAME                            STATUS   ROLES                  AGE   VERSION
pe-iu-node16.lab.appviewx.net   Ready    <none>                 25h   v1.20.7
pe-iu-node17.lab.appviewx.net   Ready    <none>                 25h   v1.20.7
pe-iu-node18.lab.appviewx.net   Ready    <none>                 25h   v1.20.7
pe-iu-node19.lab.appviewx.net   Ready    control-plane,master   25h   v1.20.7
pe-iu-node20.lab.appviewx.net   Ready    control-plane,master   25h   v1.20.7
pe-iu-node21.lab.appviewx.net   Ready    control-plane,master   25h   v1.20.7
/home/appviewx/FP6/appviewx_kubernetes/scripts/script_util

Components will take few mins to start..Please wait..
Old certificates and conf file backups can be found under /etc/kubernetes/external_ca_bkp_05-03-2022_22_51_42 and /var/lib/kubelet/pki_05-03-2022_22_51_42
Logs can be found under /home/appviewx/FP6/appviewx_kubernetes/scripts/script_util/../../logs/kubeadm-external-ca_05-03-2022_22_51_42.log
[appviewx@pe-iu-node18 scripts]$
[appviewx@pe-iu-node18 scripts]$
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Rollback Steps For Failure in Certificate Updates

This section describes the commands that can be executed to restore the certificates and config files to their previous state, in the event of a certificate update failure.



Note: The pods can either be in the **Init:CrashLoopBackOff** state or the **Pending** state.

1. **Init:CrashLoopBackOff:** If the pod is in this state, delete the pods by executing the command

```
kubectl delete pod <podname> -n <namespace> –force
```

2. **Pending:** If the pod is in this state, execute the commands in the order mentioned below:

   a.
```
kubectl scale –replicas=0 deploy/<component name> -n <namespace>
```

   b.
```
kubectl get pods --all-namespaces | awk '{if ($4=="Terminating") print "kubectl delete pod " $2 " -n " $1 " --force --grace-period=0 ";}' | sh > /dev/null 2>&1
```

   c.
```
kubectl scale –replicas=3 deploy/<component name> -n <namespace>
```

   Replicas can be changed based on the initial setup.

Uninstalling AppViewX

Users can uninstall AppViewX when they want to migrate into another environment. They can also uninstall AppViewX when it is no longer required.

To uninstall an application package safely:

1. Open the terminal window.
2. To navigate to the **appviewx_kubernetes** directory, execute the following command:

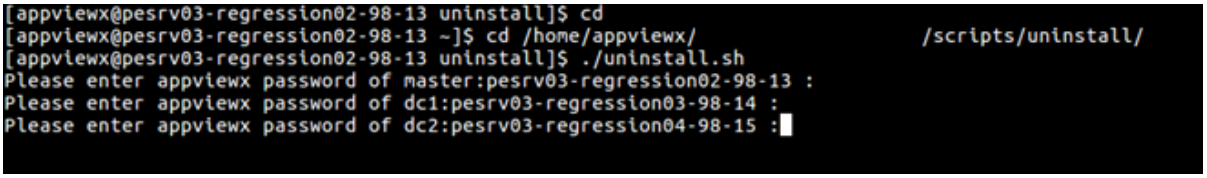
```
cd /home/appviewx/appviewx_kuberbetes/scripts/uninstall
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

3. To start the uninstallation process, execute the following command:

```
/uninstall.sh
```

4. Enter the node's credentials when prompted.



5. Reboot all the nodes after completion of the AppViewX uninstallation.

Troubleshooting

This section lists the issues encountered with AppViewX.

Whenever the AppViewX installation fails, you will get an error stating that some script execution failed.

- **Pre requisites not met**

  Please check for all the items below.

  - port not opened
  - insufficient disk/CPU
  - time not in sync
  - packages not found
  - hostname incorrect in configuration
  - etc
- **Error while initializing the kube master/worker**

  In certain cases, when uninstallation does not clean up the data properly, we may observer errors while initializing kube master and worker. In such cases, perform an uninstall, reboot all the nodes and then go ahead with the install. Additionally, there are cases where the installation fails due to port connectivity issues. If a failure occurs in this stage, check if ports 6443, 10250, 2379 and 2380 are opened properly.
- **Error while initializing the mongodb chart**

  This specific error occurs after a timeout of 5 minutes to initialize the mongodb charts. This error occurs when the pods are not able to communicate between themselves. Use the following commands to verify that:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
kubectl describe statefulset -n avx mongodb
```

For any connectivity issues, the output of this command will display the specific error stating connection timed out.

- **IP in IP tunnelling is not enabled**

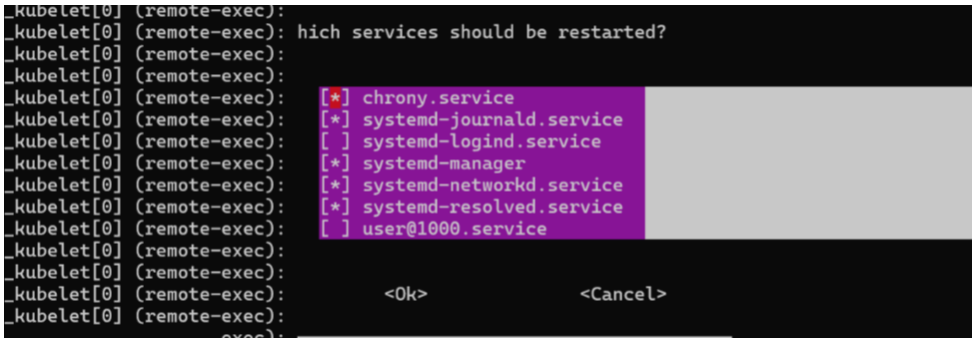
  If the IP in IP traffic is disabled, which means that the IPv4 protocol is not permitted, we will encounter the same problem. The prerequisite check script does not identify this, so we need to verify it separately to confirm.

- **Error while installing the AppViewX plugins**

  If an error occurs during the installation of AppViewX plugins, it is likely due to an error in the configuration file. You may observe and error such as `Upload failed: scp`, in such cases re-trigger plugins_install.sh to install the plugins. Likewise, ensure to review the configuration file carefully and proceed with the execution of plugins_install.sh to install only the plugins.

- **Issue with uninstall script hanging on Ubuntu 22.04 due to needrestart command**

  Root cause: It is observed that in some Ubuntu setups, the uninstall process gets stuck due to a prompt that waits indefinitely, causing the script to hang. This issue is new in Ubuntu 22.04 and is related to the needrestart command, which is now part of the **apt-get** upgrade process. By default, needrestart is set to "*interactive*" mode, leading to interruptions in scripts.



Remediation: To change this behavior, edit the /etc/needrestart/needrestart.conf file by modifying the line from

```
#$nrconf{restart} = 'i';
```

to,

```
$nrconf{restart} = 'a';
```

After making this change, subsequent runs should not encounter this issue.

- **Error initializing storage of type raft: failed to create fsm: failed to open bolt file: timeout**

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Due to an existing bug in Vault/OpenBao, data replication can sometimes exceed the default timeout for opening the Raft database. Currently, there is no configurable option to adjust this timeout. As a workaround, the **vault.db** file can be manually copied to the nodes where Vault crashes due to this issue. Please follow the steps below for this process:

Note: Before performing these steps, please take the backup of **vault.db** in the nodes where the vault has crashed. Location: $INSTALLATION_PATH/vault-data/.

1. Login to the node where the Openbao pod is running.
2. Navigate to $INSTALLATION_PATH/vault-data dir.
3. Use scp to copy the vault.db to the node where vault is crashing.

```
scp $INSTALLATION_PATH/vault-data/vault.db <username>@<node_ip>:$INSTALLATION_PATH/vault-data/
```

Source location: $INSTALLATION_PATH/vault-data/vault.db

Destination location: $INSTALLATION_PATH/vault-data/

4. Restart the crashing Openbao pods.

Steps to Change Mongodb Password

This section walks you through the steps to be taken to change the MongoDB Password.

- Untar Scripts Tarball

Untar Scripts Tarball

- Command for changing MongoDB Password
- User command

Command for changing MongoDB Password

For single-node setup

Use command

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
echo "db.getSiblingDB(\"admin\").changeUserPassword(\"admin\", \"<newpass>\")" |
        kubectl exec -it mongodb-0 -n avx -- mongo --authenticationDatabase admin -u admin –p
        <currentPass>
```

For multi-node setup/For FP5

Use command

```
echo "db.getSiblingDB(\"admin\").changeUserPassword(\"admin\", \"<newpass>\")" |
kubectl exec -it routerdb-0 -n avx -- mongo --authenticationDatabase admin -u admin –p <currentPass>
```

Note: It is necessary to cross-check whether your password is changed or not. Copy **other_user_internal.pem** file into scripts directory from **appviewx_kubernetes/scripts** directory.

User command

For single-node setup

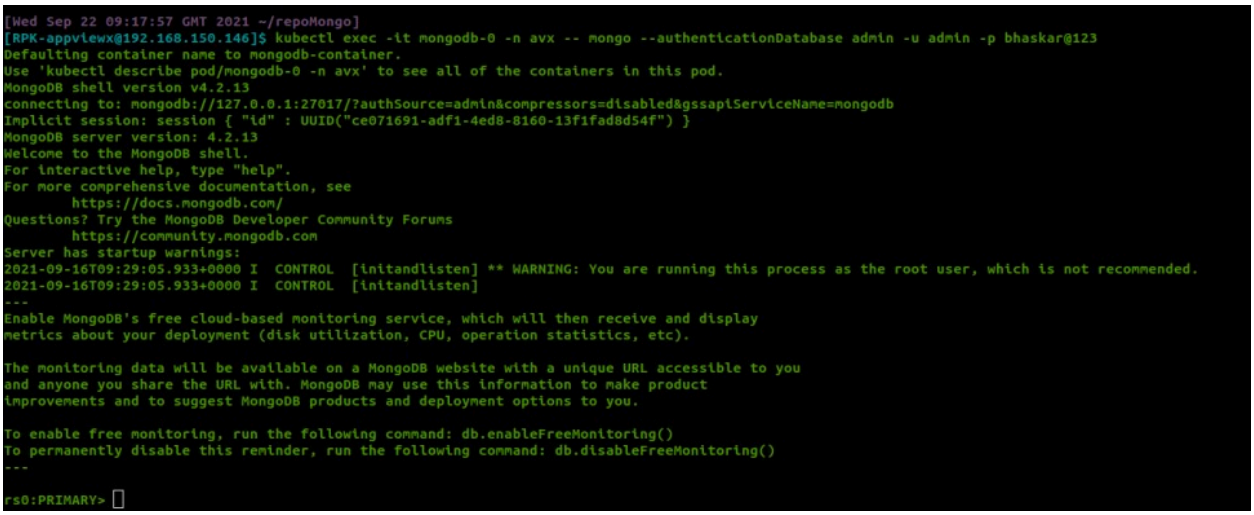
```
kubectl exec -it mongodb-0 -n avx -- mongo --authenticationDatabase admin -u admin -p
<newpass>
```

For multi-node setup/For FP5

```
kubectl exec -it routerdb-0 -n avx -- mongo --authenticationDatabase admin -u admin –p
<newpass>
```

If you are successfully logged in, it will be displayed as shown in the below image:

Copyright © 2025 AppViewX Inc. All Rights Reserved.



Trigger Script

`./scripts/appviewx.sh --password`

After successful execution of script, delete all pods.

Note: Make sure the vault is up and running.

Disable Kex Algorithm Guide

This guide is designed to help disable kex Algorithm `diffie-hellman-group1-sha1` in systems with 2022.1.0 OVAs.

- Steps to Disable Kex Algorithm

Steps to Disable Kex Algorithm

1. Run command: `nmap --script ssh2-enum-algos -p 22 <ip address>`

   Note: Replace `<ip address>` in the command with the actual IP.

   The deprecated algorithm `diffie-hellman-group1-sha1` will be active. Refer the following image for the same.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
-bash-4.2$ nmap --script ssh2-enum-algos -p 22 192.168.145.200

Starting Nmap 6.40 ( http://nmap.org ) at 2021-11-23 06:08 GMT
Nmap scan report for gs-apvx-dev120.appviewx.net (192.168.145.200)
Host is up (0.000084s latency).
PORT   STATE SERVICE
22/tcp open  ssh
| ssh2-enum-algos:
|   kex_algorithms (12)
|       curve25519-sha256
|       curve25519-sha256@libssh.org
|       ecdh-sha2-nistp256
|       ecdh-sha2-nistp384
|       ecdh-sha2-nistp521
|       diffie-hellman-group-exchange-sha256
|       diffie-hellman-group16-sha512
|       diffie-hellman-group18-sha512
|       diffie-hellman-group-exchange-sha1
|       diffie-hellman-group14-sha256
|       diffie-hellman-group14-sha1
|       diffie-hellman-group1-sha1
|   server_host_key_algorithms (5)
```

2. Run command: `sudo vi /etc/ssh/sshd_config`
   a. `sshd_config` file will open.

> **Note:** Make sure you have a KexAlgorithms list. This list should not include `diffie-hellman-group1-sha1` entry.

```
# Ciphers and keying
#RekeyLimit default none
KexAlgorithms  curve25519-sha256,curve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-sha256,diffie-hellman-group16-sha512,diffie-hellman-group18-sha512,diffie-hellman-group-exchange-sha1,diffie-hellman-group14-sha256,diffie-hellman-group14-sha1
```

> **Note:** Below mentioned is the reference text used in the above image:`KexAlgorithms curve25519-sha256, curve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-sha256,diffie-hellman-group16-sha512,diffie-hellman-group18-sha512,diffie-hellman-group-exchange-sha1,diffie-hellman-group14-sha256,diffie-hellman-group14-sha1`

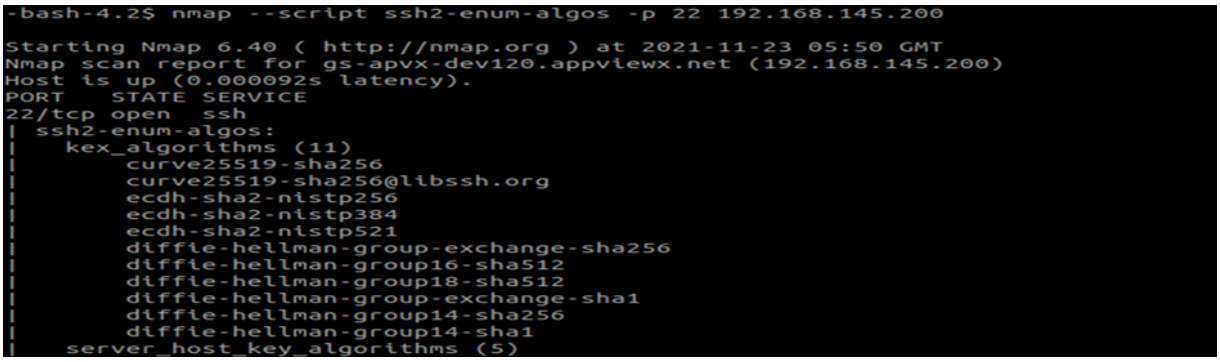
   b. Save the changes and exit from the file.
3. Run command: `udo systemctl restart sshd`
4. Execute command: `nmap --script ssh2-enum-algos -p 22 <ip address>`

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Note:

a. Replace <ip address> with the actual IP.

b. Confirm that diffie-hellman-group1-sha1 is disabled. Refer the following image for the same.



Migrating CentOS to Ubuntu/RHEL

Attention: This document is only for customers with CentOS on AppViewX v2020.3.0 FP10 and FP11 and upgrading to v2024.3.0.0. You may ignore this if you are currently using AppViewX v2022.1.0 (FP1-FP3), v2023.1.0 and v2023.1.0 FP1.

AppViewX would like to notify their customers about an important update regarding the Operating System (OS) support for AppViewX installations. Presently with AppViewX 2020.3.10 and 2022.1.3, we support three Operating Systems: CentOS & RHEL 7, Ubuntu 20.04 LTS, RHEL 8x

However, as some may be aware, CentOS is rapidly approaching its end-of-life (EOL) in June 2024, we need to adjust our AppViewX deployment strategy.

Starting from 2022.1.3 (Ganga FP3), we have transitioned from providing the OVAs with CentOS 7 to Ubuntu. This change ensures that our new customers receive continuous support without any complications. However, please be aware that existing customers can still utilize the CentOS OVA until it reaches its EOL in June 2024 but, we strongly encourage customers from CentOS 7 as soon as they are able to.

For a seamless transition, we advise our current customers who are using CentOS to migrate to either the Ubuntu OVA or bring their own Ubuntu/RHEL OS before CentOS reaches its end-of-life (EOL). Detailed migration steps can be found in the sections below.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- Prerequisites
- AppViewX 2020.3.0 FP10 (CentOS) to v2024.3.0.0 (Ubuntu/RHEL)
- AppViewX 2020.3.0 FP10 (CentOS) to v2020.3.0 FP10 (Ubuntu/RHEL)
- Post Upgrade Steps

Prerequisites

- New Nodes must have any of the following OS:
  - RHEL 8.5-8.8, 9.2, or 9.3
  - Ubuntu 20.04
- To list the nodes in the cluster, execute the command below and save the outputs for further reference

```
kubectl get nodes --show-labels
```

> **Note:** If custom labels are detected add them to the custom_changes.yaml file. Refer to the chapter *Adding Custom Pod Configuration* of the section **Monitoring and Maintaining AppViewX** in the **Install, Upgrade, and Maintenance Guide**.

- To get the status of HPA, execute command below and save the outputs for further reference

```
kubectl get hpa
```

- Keep a backup of the iControl jar files available at location -/home/appviewx/appviewx/appviewx_dependencies/external_libs/iControl-13.1.0.jar for iControl to be done after the migration.
- Check for any other custom changes that may have been done specifically for the customer
- Check the enabled plugins in appviewx.conf file of the old installer (previous setup), in case there are plugins that are not present but are required to be installed in the new setup, please add them, example
  - avx_pkiaas_cert_ocsp_generator
  - avx_pkiaas_cert_ocsp_server, avx_pkiaas_cert_ocsp_server
  - avx_platform_hsm

AppViewX 2020.3.0 FP10 (CentOS) to v2024.3.0.0 (Ubuntu/RHEL)

Overview

1. Take the mongodb and vault backup from the old CentOS environment.
2. Install AppViewX v2024.3.0.0 in the new environment with Ubuntu/RHEL.
3. Restore mongodb in the new environment.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

4. Restore the vault in the new environment.
5. Trigger the plugins install.

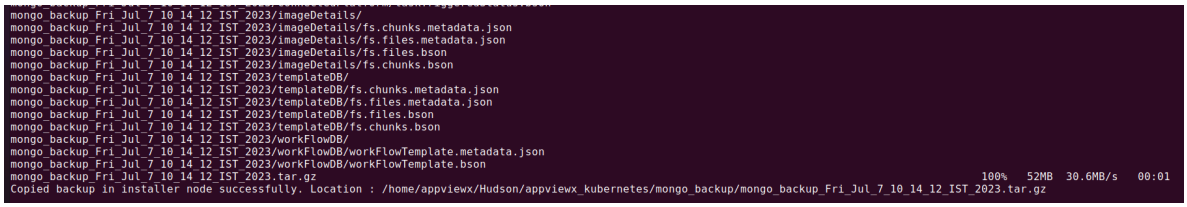
1. In the CentOS environment, take mongodb and vault backup as follows

   a. Navigate to installer node's scripts directory - <appviewx_installer_location>/appviewx_kubernetes/scripts/

   b. Initiate a backup of the AppViewX Database using the following command

   ```
   ./mongo_backup.sh
   ```

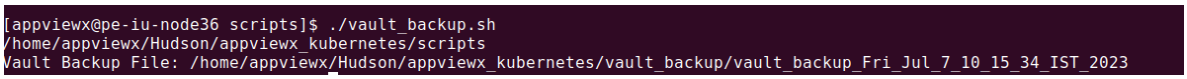
   After the backup is taken successfully, the database's backup file and its location are displayed on the screen. Copy this backup file to a safe location for future reference.

   c. Initiate a backup of the Secrets Vault using the command

   ```
   ./vault_backup.sh
   ```

   After the backup is taken successfully, the vault's backup file and its location are displayed on the screen. Copy this backup file to a safe location for future reference.

2. If you have completed the backup process, proceed to install the new server with the chosen version on the supported operating system.

   a. Choose the preferred OS or OVA from the supported list.
      i. RHEL (8.7, 8.8, 8.10, 9.2, 9.3 or 9.4)
      ii. Ubuntu 20.04 LTS, 22.04
      iii. Ubuntu OVA (with Ubuntu 20.04 LTS)

   b. Once you have decided on the OS or OVA, install the AppViewX server by following the steps below:
      i. Install 2024.3.0.0 with OVA - Click here or
      ii. Install 2024.3.0.0 with installer - Click here

3. To restore mongodb in the new environment,

Copyright © 2025 AppViewX Inc. All Rights Reserved.

a. Copy the backed up file(s) to the new environment's installer node.

b. Navigate to the installer location's scripts directory - <appviewx_installer_location>/appviewx_kubernetes/scripts/

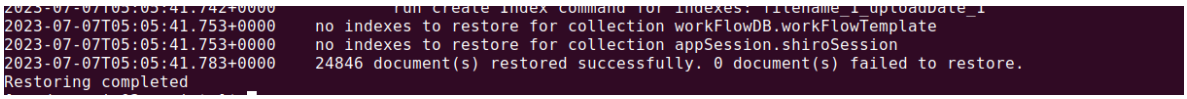
c. Restore the database by triggering database restoration script. Execute the following command:

```
./mongo_restore.sh <location of mongo backup file>
```

Example:

```
./mongo_restore.sh /home/appviewx/mongo_backup/mongo_backup_Fri_Jul_7_10_14_12_IST_2023.tar.gz
```

d. Wait for the successful completion message.

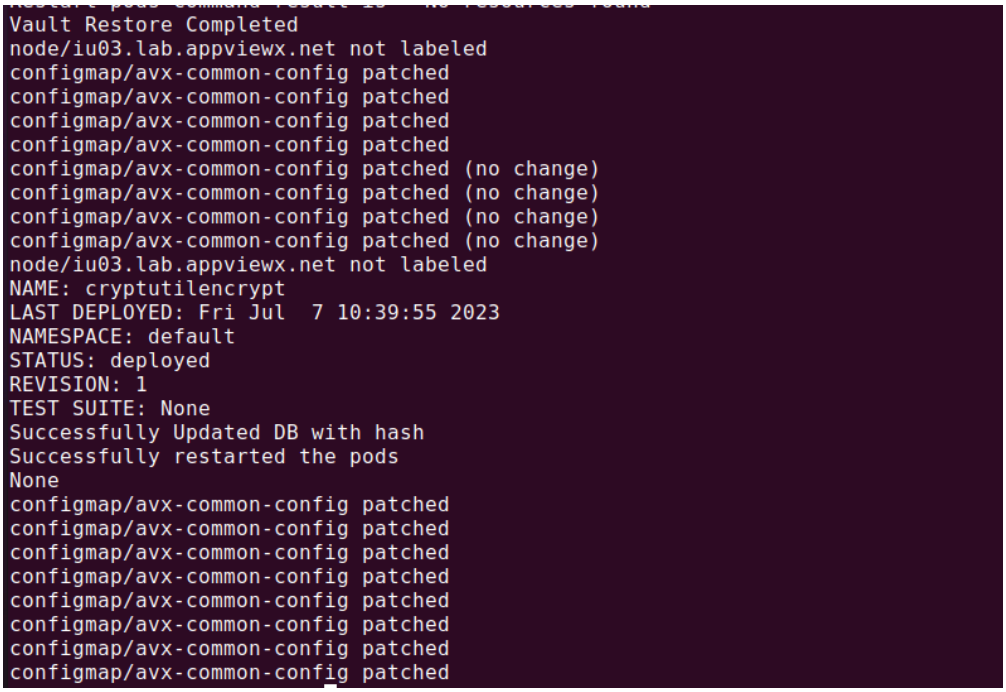


4. Restore the Secrets Vault by triggering the vault restoration script by using the following command:

```
./vault_restore.sh -p <location of vault backup file> --removedek
```

Example:

```
./vault_restore.sh -p /home/appviewx/vault_backup/vault_backup_Wed_Jul_28_05_50_40_UTC_2021 –removedek
```



5. To install plugins,

Copyright © 2025 AppViewX Inc. All Rights Reserved.

a. Navigate to the appviewx_kubernetes/scripts folder.

b. Execute the following command

```
./plugins_install.sh
```

AppViewX 2020.3.0 FP10 (CentOS) to v2020.3.0 FP10 (Ubuntu/RHEL)

1. Take the mongodb and vault backup from the old environment with CentOS.
2. Install AppViewX v2023.3.0 FP10 in the new environment with Ubuntu/RHEL.
3. Restore mongodb in the new environment.
4. Restore the vault in the new environment.

Refer the detailed steps in the section above (only steps 1 to 4).

Post Upgrade Steps

The following actions must be taken to avoid any post-migration errors listed below.

1. **Loss of Mongo replica set priority configurations**

   During installation, mongodb is freshly set up with the latest upgraded versions. The previous replicaset configurations such as replicaset priorities will not be taken ahead and hence have to be re-configured. High latency customers must perform the following step:

   a. Configure the parameters OPTIMISE_ROUTING_FOR_LATENCY and PREFERRED_DEFAULT_DC in the appviewx.conf

   b. Re-trigger the `plugins_install.sh` to change the configurations.

2. **Custom changes**

   If the custom_changes.yaml, custom_vm_args.conf are present and updated in the custom changes, then the custom changes will be persistent. Any of the custom changes that may have been done specifically for the customer as noted in the prerequisites will not be present if the above mentioned files are not updated with this configuration.

3. **External web cert is not upgraded from 2022.1.0 to 2023.1.0**

   To update the external CA web certificate, execute the command below:

```
./appviewx.sh --update-web-cert
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

The following prompts will be displayed:

- Enter the absolute path of external cert file:
- Enter the absolute path of external key file:

Enter both the values to proceed. Once the cert upgrade is completed, restart the gateway and web.

4. Set the ELASTIC_ENABLE as True in the **Statistics Configuration**. There are two ways to do it, choose from either the command prompt (a) or from the management console UI (b).

> **Note:** Elastic insight must be reinstalled. Application upgrade will take care of installing insight if it's already there in the existing setup, but this won't be taken care automatically in the fresh install. Hence, install insight before step b.

a. Execute the command

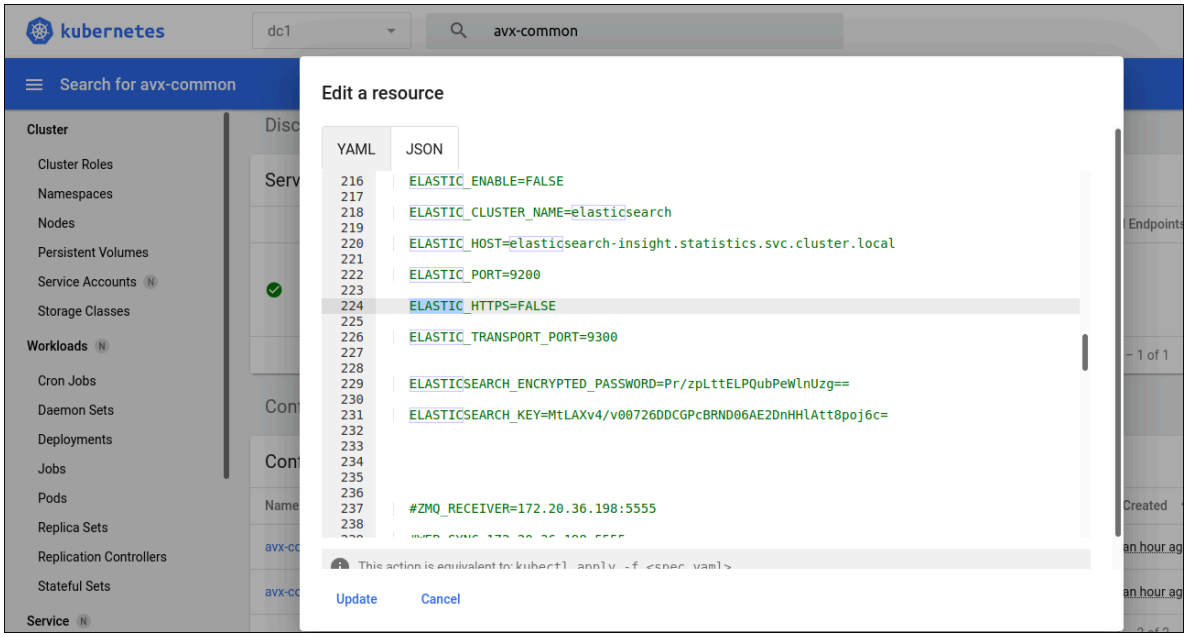
```
kubectl edit configmaps -n <Datacenter1>
```

Search for the keyword as Elastic and set ELASTIC_ENABLE as True and below params should have default values as below

```
ELASTIC_ENABLE=TRUE
ELASTIC_CLUSTER_NAME=elasticsearch
ELASTIC_HOST=elasticsearch-insight.statistics.svc.cluster.local
ELASTIC_PORT=9200
ELASTIC_HTTPS=FALSE
ELASTIC_TRANSPORT_PORT=9300
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

b. Login to management console >> Search for the namespace with the configured DC >> Search for avx-common-config in config maps >> Click on Edit and search for Elastic >> Set as True and give as update as shown below.



- Enabling HSM
- Elastic Restore

Enabling HSM

Note: Refer this section only if you have upgraded from AppViewX v2020.3.0.

Prerequisites

- To configure Fortanix and Utimaco, the **.so** file and **config** file must be present in the current Appviewx version.
  - The **.so** file is essential for communicating with the HSM using the PKCS11 interface.
  - The **config** file facilitates communication between the HSM and Appviewx.

After the successful upgrade, proceed with the steps below to enable HSM.

1. Ensure the HSM pod is operational and running in the required datacenters and that the HSM node is specified in the appviewx.conf file, execute the following command:

```
kubectl get pods -A -o wide |grep hsm
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

2. From the command line interface, navigate to the properties folder path {APPVIEWX_INSTALLATION_PATH}/appviewx_dependencies/properties

For Fortanix

a. Open the HSM file using the following command:

```
vi hsm
```

b. Check and confirm if the HSM file has the following lines. If not, uncomment the following lines:

```
export FORTANIX_PKCS11_CONFIG_PATH= /appviewx/dependencies/hsm/fortanix/pkcs11.conf
```

```
echo "FORTANIX Config Path : $FORTANIX_PKCS11_CONFIG_PATH"
```

c. If the file is edited, restart the **avx-platform-hsm** pod, using the following commands:

```
kubectl get pods -n <namespace>
```

```
kubectl delete pods -n <namespace> <PodName> --force
```

For Utimaco

a. Open the HSM file using the following command:

```
vi hsm
```

b. Check and confirm if the HSM file has the following lines. If not, uncomment the following lines:

```
export CS_PKCS11_R2_CFG=/appviewx/dependencies/hsm/utimaco/cs_pkcs11_R2.cfg
```

```
echo "UTIMACO Config Path : $CS_PKCS11_R2_CFG"
```

c. If the file is edited, restart the **avx-platform-hsm** pod, using the following commands:

```
kubectl get pods -n <namespace>
```

```
kubectl delete pods -n <namespace> <PodName> --force
```

3. Once the HSM pod is back to running state, login to AppViewX and navigate to **Platform** > **Vault & Security** > **HSM**.
4. Access the required HSM.

- Configuring Fortanix
- Configuring Utimaco
- Verifying/Modifying HSM Configuration for Private Key Encryption

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Configuring Fortanix

1. If the added HSM is Fortanix, upload the **.so** file and **config** file as follows:
   To upload the **.so** file,
   a. Click **Browse**.
   b. Navigate to the location of the .so file.
   c. Select the .so file and click **Open**.
   To upload the **.cfg** (config) file,
   a. Click **Browse**.
   b. Navigate to the location of the .cfg file.
   c. Select the .cfg file and click **Open**.
2. Choose the respective datacenter.
3. Update each HSM available in the inventory one by one, ensuring that the HSM is moved to the **Available** status.

Configuring Utimaco

1. If the added HSM is Utimaco, upload the **.so** file and **config** file as follows:
   To upload the **.so** file,
   a. Click **Browse**.
   b. Navigate to the location of the .so file.
   c. Select the .so file and click **Open**.
   To upload the **.cfg** (config) file,
   a. Click **Browse**.
   b. Navigate to the location of the .cfg file.
   c. Select the .cfg file and click **Open**.
2. Choose the respective datacenter.
3. Update each HSM available in the inventory one by one, ensuring that the HSM is moved to the **Available** status.

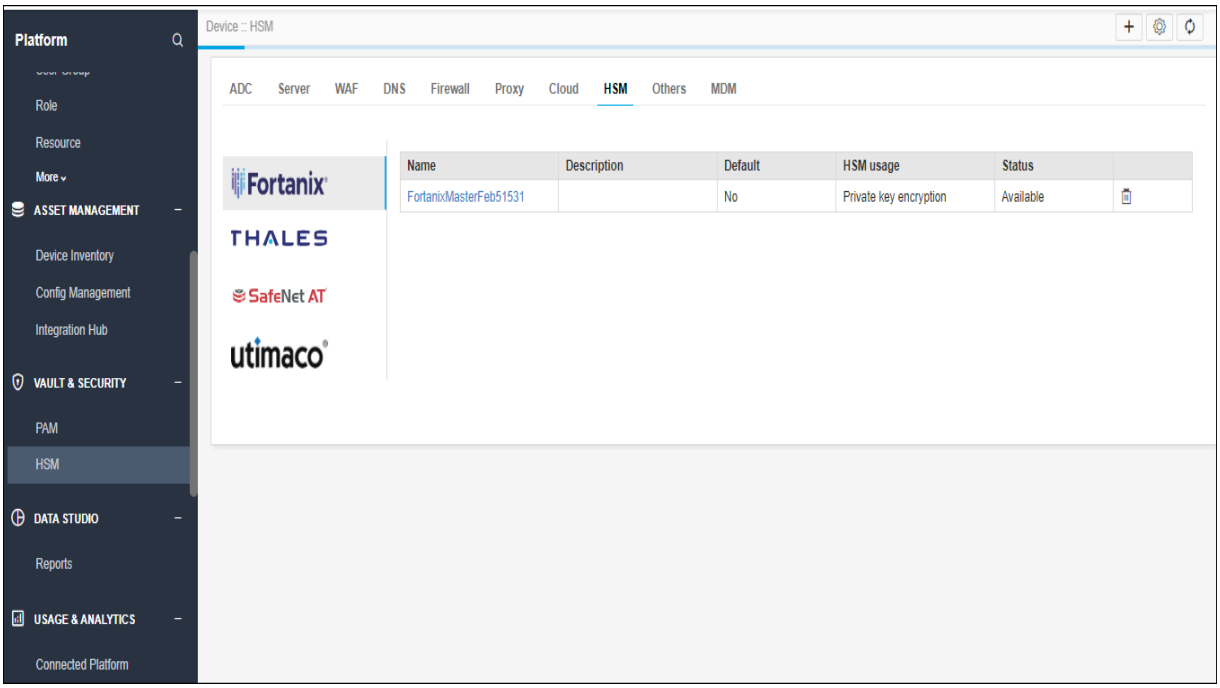
If you have previously configured Thales GPN, Thales DPoD, and Thales TCT in the AppViewX v2020.3.0 and have performed the Application Upgrade, then freshly configure the same in the upgraded version. Refer the links below for the configuration:

- Thales GPN
- Thales DPoD
- Thales TCT

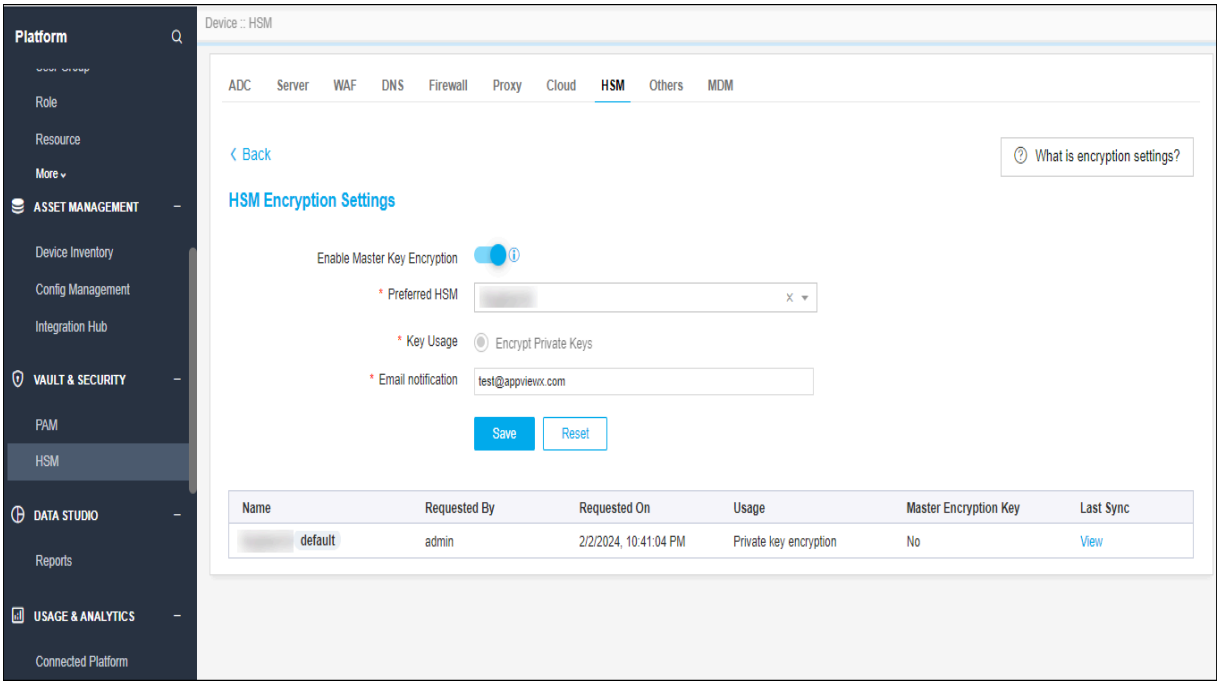
Copyright © 2025 AppViewX Inc. All Rights Reserved.

Verifying/Modifying HSM Configuration for Private Key Encryption

1. On the top-right corner of the HSM inventory page, click the master encryption settings icon.



2. If the already added HSM has implementation type **private key generation** or **Both** and Set as “default” in 2020.3.0 Appviewx version then that HSM will be available in the Master encryption settings page as default HSM.



Copyright © 2025 AppViewX Inc. All Rights Reserved.

3. To receive emails for default HSM health status, configure the email address in the settings page.

> **Note:** For email use case SMTP settings should be available in **Platform** > **System Administration** > **SMTP**.

4. To change the new default HSM, add the new HSM with HSM usage as **Master key encryption** or **both**.
5. Once the HSM is moved to **Available** status it will be present in the **Preferred HSM** dropdown field of the master key encryption settings page.
6. Choose the new HSM and click **Save**.

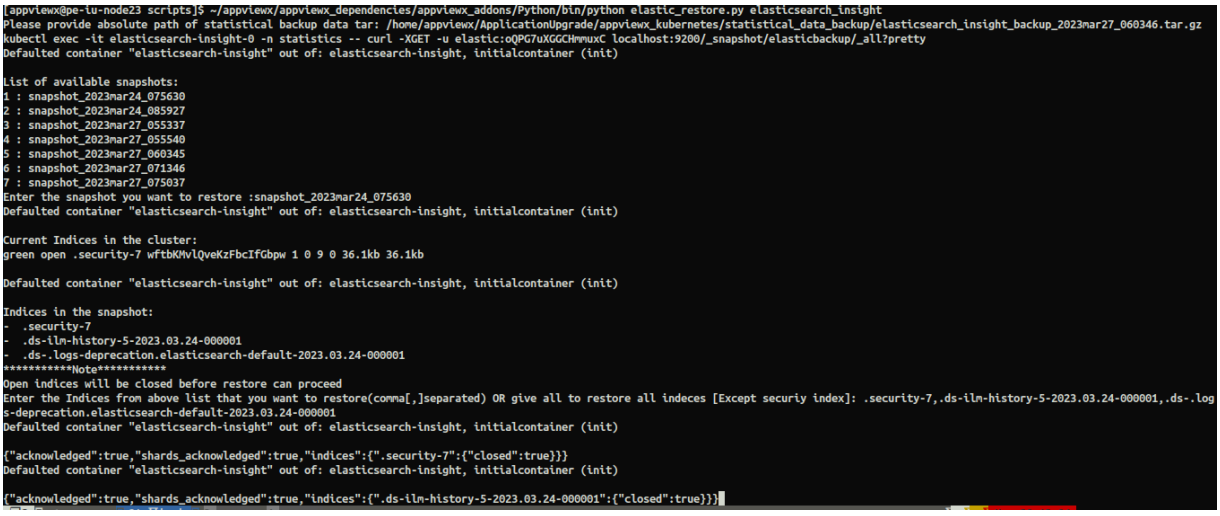
Elastic Restore

The script **elastic_restore.py** is used for restore. To manually perform the elastic restore,

1. Navigate to the scripts directory.
2. Run the elastic_restore.py script to restore the backup.

```
/home/appviewx/appviewx/appviewx_dependency/appviewx_addons/Python_Linux/bin/python elastic_restore.py elasticsearch_insight
```

3. Script will ask for the backup tar which was created manually. Provide the absolute path of the backup tar.



4. Script will list all the available snapshots that has date and time in the naming. Select the backup which you want to restore.
5. Provide the details of the indices you want to restore (follow the screenshot above).

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Upgrading RHEL to v8.10/v9.4

This section provides instructions on how to perform an in-place upgrade from Red Hat Enterprise Linux (RHEL) 8.x to versions 8.10 and 9.4.

- Prerequisites
- Upgrading RHEL to v8.10
- Upgrading RHEL to v9.4
- Post-Upgrade Steps and Troubleshooting

Prerequisites

Before you upgrade the OS version, perform the steps below.

1. **Take snapshot of existing setup** - VM snapshots of the current setup should be taken as a backup.
2. **Stop the application and services that are running** - Execute the commands in the order below in all the nodes, if you have a multi-node setup.

```
systemctl stop appviewx.service
```

```
systemctl stop kubelet.service
```

```
systemctl stop containerd.service
```

3. **Disable the following services** - Execute the commands in the order below in all the nodes, if you have a multi-node setup.

```
systemctl disable appviewx.service
```

```
systemctl disable kubelet.service
```

```
systemctl disable containerd.service
```

Upgrading RHEL to v8.10

This section provides instructions on how to perform an in-place upgrade from RHEL 8.x to 8.10.

1. Set the target OS Version using subscription-manager command below.

```
sudo subscription-manager release --set=8.10
```

2. Clear cache using the command below.

```
sudo dnf clean all
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

3. Check if any updates are available for the system. Run the command below.

```
sudo dnf check-update
```

4. To update package to the latest version that are available and resolvable, run the command below.

```
sudo dnf upgrade -y
```

5. Reboot the system using the command below.

```
sudo reboot
```

6. To check the upgraded version of the OS, run the command

```
cat /etc/os-release
```

7. (*Optional step*) If there are any kernel specific error messages write the kernel messages in Linux operating systems to standard output using the **dmesg** command.

```
dmesg | grep -i 'err|warn|cri'
```

Upgrading RHEL to v9.4

Important:

- If your current OS version is 8.x, upgrade to version 8.10 first, and then follow the steps to upgrade to version 9.4.
- If you've already upgraded to version 8.10, follow the prerequisites and then proceed with the upgrade to version 9.4.

Refer to the Upgrading from RHEL 8 to RHEL 9section on the Red Hat documentation website.

Post-Upgrade Steps and Troubleshooting

Prerequisite for RHEL 9.4

Note: After you have upgraded the OS to RHEL 9.4 and before you start using AppViewX, perform the steps below to update the **sysctl.conf** file.

1. To update the **sysctl.conf** file, execute the command below.

```
sudo vi /etc/sysctl.conf
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

2. Add these two lines (parameters) to the end of the file.

```
fs.inotify.max_user_instances=8192
```

```
fs.inotify.max_user_watches=524288
```

3. Save and exit.
4. Now, run the command below.

```
sudo sysctl -p
```

Starting the Services and Pods

Perform these steps after you upgrade to RHEL 8.10 or 9.4

1. Start the infra and application services using the following commands

   a. To enable the services, run the command below.

```
systemctl enable containerd.service
```

```
systemctl enable kubelet.service
```

```
systemctl enable appviewx.service
```

   b. To start the services, run the command below.

```
systemctl start containerd.service
```

```
systemctl start kubelet.service
```

```
systemctl start appviewx.service
```

2. Restart all the pods by executing the command below.

```
kubectl delete po –all -A –force
```



Note: Wait until all the pods are in the running state.

3. Verify the application status.

   a. To check if all pods are up and running, run the command below.

```
kubectl get pods -A
```

   b. Ensure the UI is accessible and functioning as expected.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Troubleshooting

1. **Handling istio issues (if present).**

   a. If the **istio-ingressgateway** pods are stuck in 0/1 Running state, run the command below.

   ```
   sudo iptables -P INPUT ACCEPT; sudo iptables -P FORWARD ACCEPT; sudo iptables -P OUTPUT ACCEPT; sudo iptables -F; sudo iptables -X; sudo iptables -Z; sudo iptables -t nat -F; sudo iptables -t nat -X; sudo iptables -t mangle -F; sudo iptables -t mangle -X; sudo iptables -t raw -F; sudo iptables -t raw -X;
   ```

   This command flushes the iptables rules to resolve any networking issues related to istio.

   b. Restart all pods (after flushing the iptables).

   ```
   kubectl delete pods --all -A --force
   ```

2. **Handling pod specific issues (if present)** - If vault, mongo, or other pods start crashing with error message as `xtables parameter problem: iptables-restore: unable to initialize table 'nat'`, do the following:
   a. Manually create the **iptable_filter.conf** file under **/etc/modules-load.d/**.
   b. To ensure the changes take effect after creating the above file, reboot the node.

Upgrading Ubuntu 20.04 to Ubuntu 22.04

This guide is intended for all AppViewX customers that are currently using Ubuntu 20.04 operating system. It offers step-by-step instructions for transitioning from Ubuntu 20.04 to Ubuntu 22.04, covering essential pre-upgrade preparations, updating packages, configuring necessary repositories, and resolving compatibility issues with deprecated modules for a seamless upgrade. This process applies specifically to AppViewX-shipped OVAs.

• Initiating the Upgrade Process
• Post Upgrade Actions
• Troubleshooting

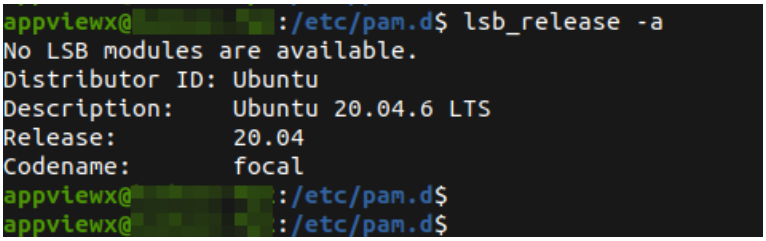
Pre-Upgrade Considerations

Before starting the upgrade, ensure the following are addressed:

- **Backup:** Take a VM snapshot before starting the upgrade to enable rollback in case of issues.
- **Internet Connectivity:** Ensure the node has internet or proxy access to download the OS update from `repos.appviewx.com`.
- **Disk Space:** Verify that at least 3GB of free space is available in the root (`/`) partition.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- **Current Version Check:**Confirm that the system is running Ubuntu 20.04 by executing the following command:
  - lsb_release -a



Prerequisite

1. Stop AppViewX on All Nodes (On-Premise Setups Only).

> **Note:** This step is applicable only to on-premise setups. Cloud Connectors (CC) do not require component stoppage.

a. Log in to the AppViewX installer node and navigate to the appviewx_kubernetes/scripts directory.

b. If you are upgrading the OS **with a downtime**, then use the following command to stop the AppViewX on all the nodes.

i. For **single-node**:

```
./appviewx.sh --stop <hostname>
```

ii. For **multi-node**:

```
./appviewx.sh --stop -all
```

c. If you are upgrading the OS **without a downtime**, then use the following command to stop the AppViewX nodes.

- For multi-node - This applies only to multi-node clusters. For multi-master clusters, ensure that the master load balancer is properly configured to prevent downtime.

```
./appviewx.sh –stop <hostname>
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- Before proceeding to the next node, upgrade the OS and verify that the components are running on the node using:

```
kubectl get pods -A -o wide | grep -i <hostname>
```

> **Note:** Use VMware Console access for this upgrade. If upgrading over SSH, use "**screen**" or "**tmux**" to prevent interruptions in case of disconnection.

2. Check if the **tmux** package is installed on your node. If not, install it using

```
sudo apt install tmux
```

3. Attach to a tmux session by running:

```
tmux
```

Preparing for the Upgrade

To prepare for the upgrade, update all installed packages on Ubuntu 20.04. Execute the following commands:

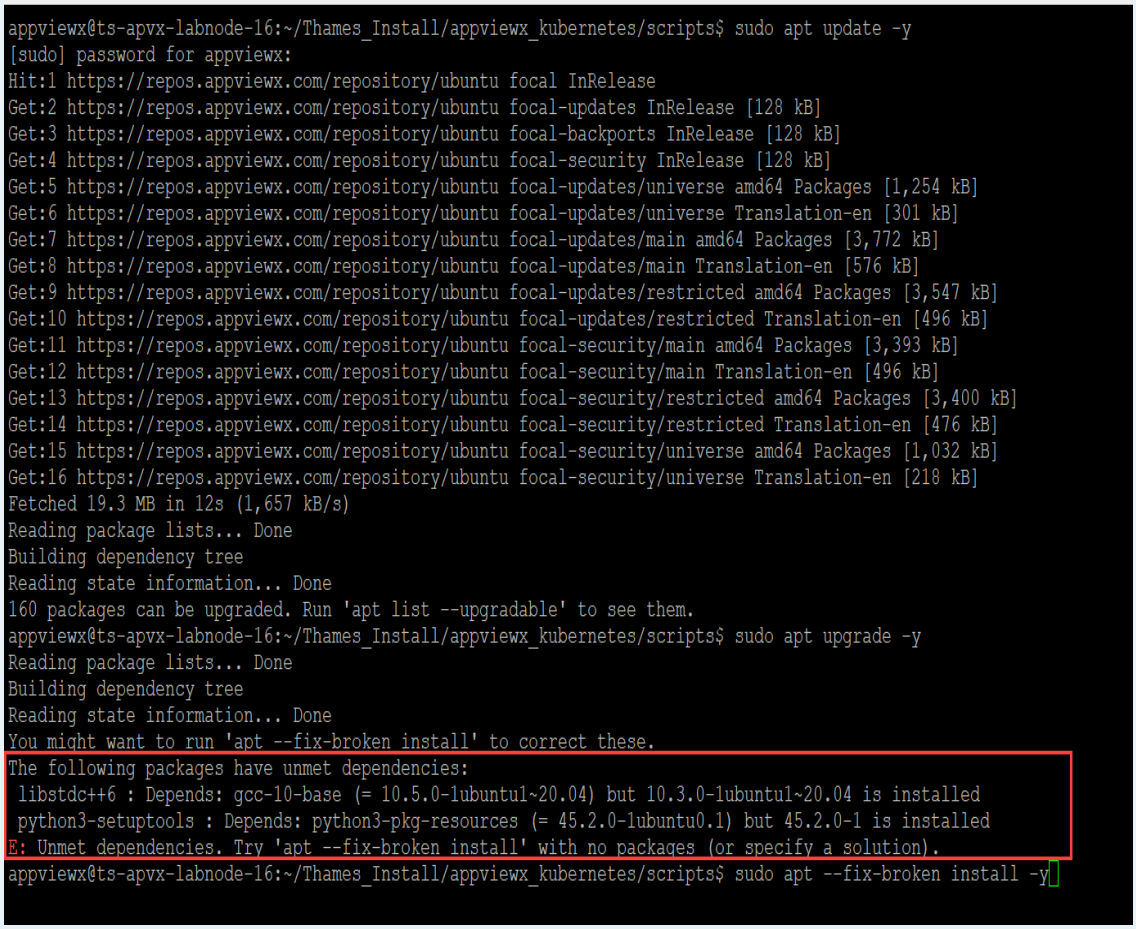
- `sudo apt update -y`

```
root@[redacted]:~#
root@[redacted]:~# sudo apt update -y
Hit:1 https://repos.appviewx.com/repository/ubuntu focal InRelease
Get:2 https://repos.appviewx.com/repository/ubuntu focal-updates InRelease [128 kB]
Get:3 https://repos.appviewx.com/repository/ubuntu focal-backports InRelease [128 kB]
Get:4 https://repos.appviewx.com/repository/ubuntu focal-security InRelease [128 kB]
Get:5 https://repos.appviewx.com/repository/ubuntu focal-updates/universe amd64 Packages [1,237 kB]
Get:6 https://repos.appviewx.com/repository/ubuntu focal-updates/universe Translation-en [297 kB]
Get:7 https://repos.appviewx.com/repository/ubuntu focal-updates/main amd64 Packages [3,657 kB]
Get:8 https://repos.appviewx.com/repository/ubuntu focal-updates/main Translation-en [560 kB]
Get:9 https://repos.appviewx.com/repository/ubuntu focal-updates/restricted amd64 Packages [3,346 kB]
Get:10 https://repos.appviewx.com/repository/ubuntu focal-updates/restricted Translation-en [468 kB]
Get:11 https://repos.appviewx.com/repository/ubuntu focal-security/main amd64 Packages [3,284 kB]
Get:12 https://repos.appviewx.com/repository/ubuntu focal-security/main Translation-en [482 kB]
Get:13 https://repos.appviewx.com/repository/ubuntu focal-security/universe amd64 Packages [1,014 kB]
Get:14 https://repos.appviewx.com/repository/ubuntu focal-security/universe Translation-en [214 kB]
Fetched 14.9 MB in 10s (1,539 kB/s)
Reading package lists... Done
Building dependency tree
```

- `sudo apt upgrade -y`

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Note: If there is an error on executing the above command as shown below



then execute the following command:

```
sudo apt --fix-broken install -y
```

After the `apt --fix-broken install` command executes successfully, execute the `apt upgrade` command once again.

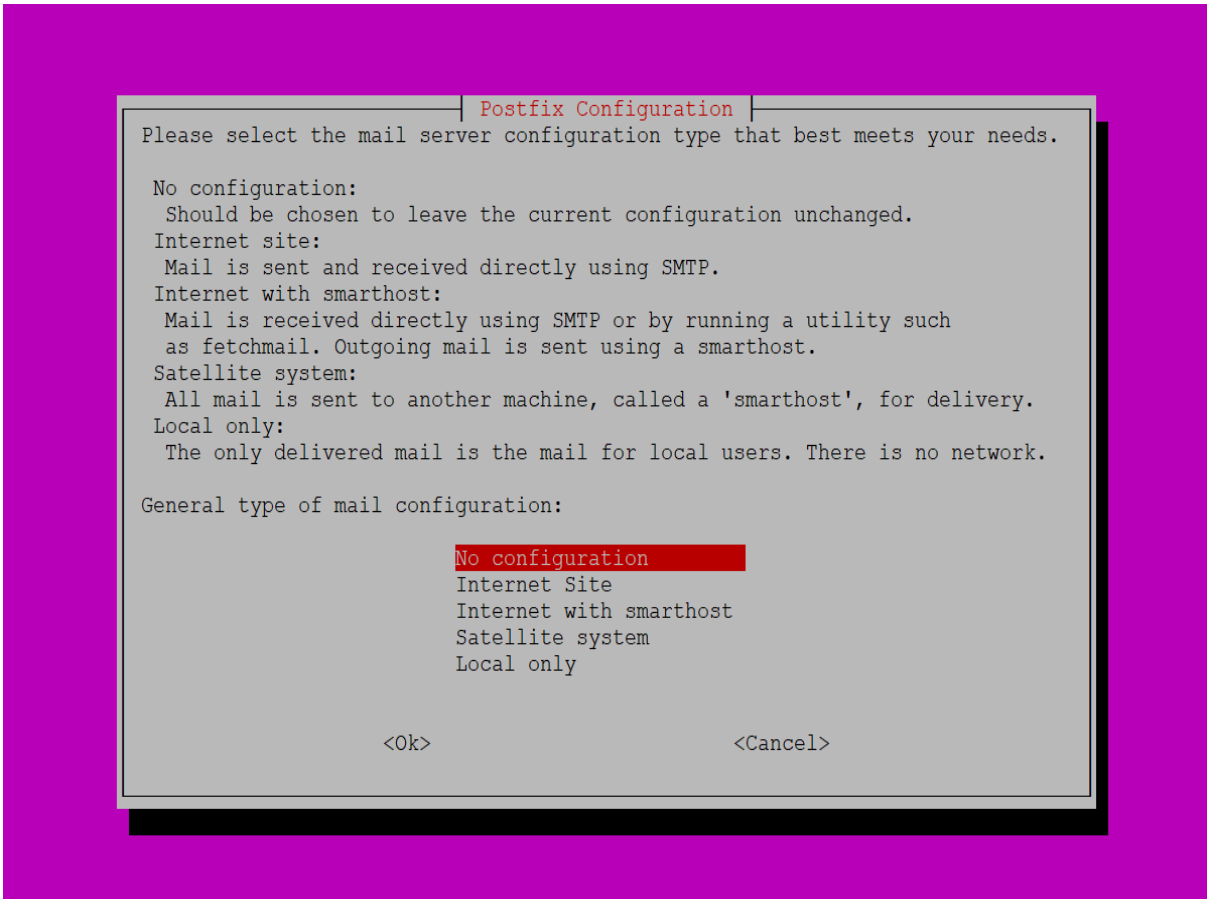
```
sudo apt upgrade -y
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
root@ :~# sudo apt upgrade -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  adwaita-icon-theme at-spi2-core gtk-update-icon-cache hicolor-icon-theme humanity-icon-theme libatk-bridge2.0-0 libatk1.0-0 libatk1.0
  libatspi2.0-0 libavahi-client3 libavahi-common-data libavahi-common3 libcairo-gobject2 libcairo2 libcolord2 libcups2 libdatrie1 libep
  libevent-core-2.1-7 libevent-pthreads-2.1-7 libfwupdplugin1 libgdk-pixbuf2.0-0 libgdk-pixbuf2.0-bin libgdk-pixbuf2.0-common libgraphi
  libgtk-3-0 libgtk-3-bin libgtk-3-common libharfbuzz0b libjbig0 libjpeg-turbo8 libjpeg8 liblcms2-2 libopts25 libpango-1.0-0 libpangoca
  libpangoft2-1.0-0 libpixman-1-0 librest-0.7-0 librsvg2-2 librsvg2-common libsoup-gnome2.4-1 libthai-data libthai0 libtiff5 libwaylanc
  libwayland-cursor0 libwayland-egl1 libwebp6 libxcb-render0 libxcb-shm0 libxcomposite1 libxcursor1 libxdamage1 libxfixes3 libxi6 libxi
  libxkbcommon0 libxmlb1 libxrandr2 libxrender1 libxtst6 linux-headers-5.4.0-156 linux-headers-5.4.0-156-generic linux-headers-5.4.0-17
  linux-headers-5.4.0-173-generic linux-headers-5.4.0-26 linux-headers-5.4.0-26-generic linux-image-5.4.0-156-generic linux-image-5.4.0
  linux-image-5.4.0-26-generic linux-modules-5.4.0-156-generic linux-modules-5.4.0-173-generic linux-modules-5.4.0-26-generic
```

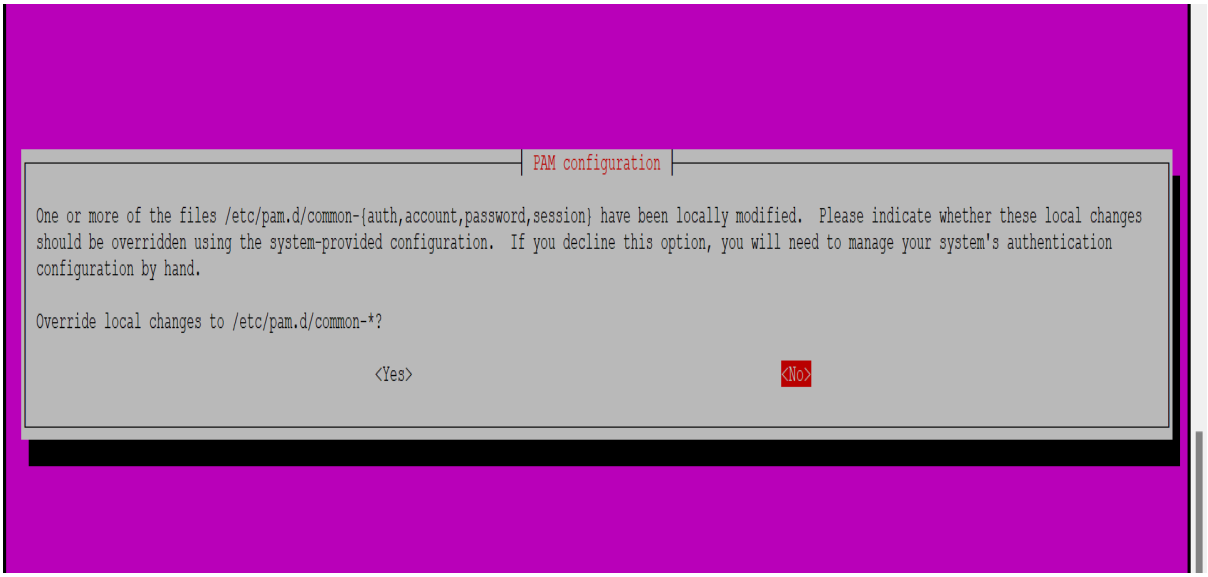
- sudo apt full-upgrade

  - On receiving a prompt for **Postfix Configuration**, select ***No configuration*** and proceed.



Copyright © 2025 AppViewX Inc. All Rights Reserved.

- On receiving a prompt for **PAM Configuration**, select ***No*** and proceed.




```
root@          :~#
root@          :~# sudo apt full-upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  adwaita-icon-theme at-spi2-core gtk-update-icon-cache hicolor-icon-theme humanity-icon-theme libatk-bridge2.0-0 libatk1.0-0 libat
  libatspi2.0-0 libavahi-client3 libavahi-common-data libavahi-common3 libcairo-gobject2 libcairo2 libcolord2 libcups2 libdatrie1 l
  libevent-core-2.1-7 libevent-pthreads-2.1-7 libfwupdplugin1 libgdk-pixbuf2.0-0 libgdk-pixbuf2.0-bin libgdk-pixbuf2.0-common libgr
  libgtk-3-0 libgtk-3-bin libgtk-3-common libharfbuzz0b libjbig0 libjpeg-turbo8 libjpeg8 liblcms2-2 libopts25 libpango-1.0-0 libpar
  libpangoft2-1.0-0 libpixman-1-0 librest-0.7-0 librsvg2-2 librsvg2-common libsoup-gnome2.4-1 libthai-data libthai0 libtiff5 libway
  libwayland-cursor0 libwayland-egl1 libwebp6 libxcb-render0 libxcb-shm0 libxcomposite1 libxcursor1 libxdamage1 libxfixes3 libxi6 l
  libxkbcommon0 libxmlb1 libxrandr2 libxrender1 libxtst6 linux-headers-5.4.0-156 linux-headers-5.4.0-156-generic linux-headers-5.4.
  linux-headers-5.4.0-173-generic linux-headers-5.4.0-192 linux-headers-5.4.0-192-generic linux-headers-5.4.0-26 linux-headers-5.4.
  linux-image-5.4.0-156-generic linux-image-5.4.0-173-generic linux-image-5.4.0-192-generic linux-image-5.4.0-26-generic
```

Configuring the AppViewX Repository

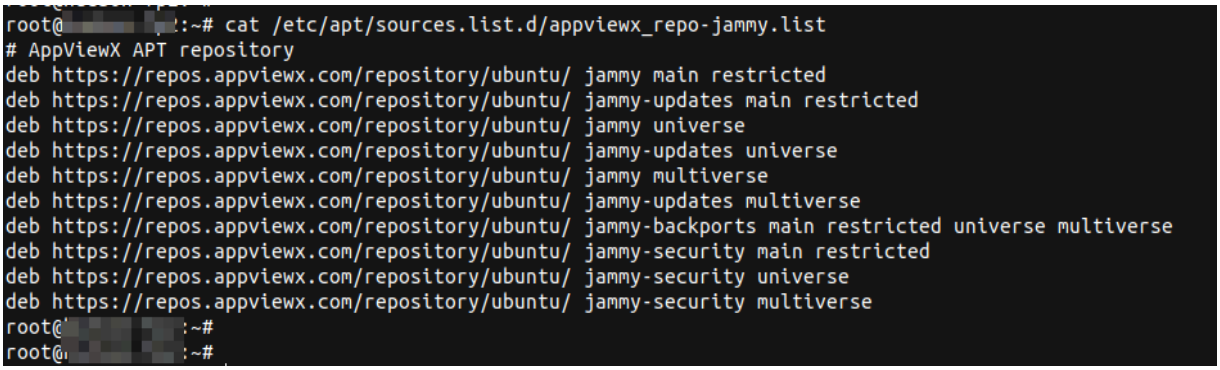
1. Validate the authentication for the repos.appviewx.com repository using the command below.

```
cat /etc/apt/auth.conf
```

2. Create a new APT Repository; add the AppViewX repository for Ubuntu 22.04 (Jammy Jellyfish).

```
sudo vi /etc/apt/sources.list.d/appviewx_repo-jammy.list
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.



Add the following lines to the file:

```
# AppViewX APT repository
deb https://repos.appviewx.com/repository/ubuntu/ jammy main restricted
deb https://repos.appviewx.com/repository/ubuntu/ jammy-updates main restricted
deb https://repos.appviewx.com/repository/ubuntu/ jammy universe
deb https://repos.appviewx.com/repository/ubuntu/ jammy-updates universe
deb https://repos.appviewx.com/repository/ubuntu/ jammy multiverse
deb https://repos.appviewx.com/repository/ubuntu/ jammy-updates multiverse
deb https://repos.appviewx.com/repository/ubuntu/ jammy-backports main restricted universe multiverse
deb https://repos.appviewx.com/repository/ubuntu/ jammy-security main restricted
deb https://repos.appviewx.com/repository/ubuntu/ jammy-security universe
deb https://repos.appviewx.com/repository/ubuntu/ jammy-security multiverse
```

Handling Deprecated Modules

To avoid issues with deprecated pam_tally and pam_tally2 modules, replace them with pam_faillock using the following commands:

```
cd /etc/pam.d
```

```
sudo sed -i 's/pam_tally2\?/pam_faillock/g' /etc/pam.d/*
```

Initiating the Upgrade Process

Execute the following commands:

```
sudo apt update -y
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
appviewx@          :/etc/pam.d$ sudo apt update -y
[sudo] password for appviewx:
Get:1 https://repos.appviewx.com/repository/ubuntu jammy InRelease [270 kB]
Get:2 https://repos.appviewx.com/repository/ubuntu jammy-updates InRelease [128 kB]
Get:3 https://repos.appviewx.com/repository/ubuntu jammy-backports InRelease [127 kB]
Get:4 https://repos.appviewx.com/repository/ubuntu jammy-security InRelease [129 kB]
Hit:5 https://repos.appviewx.com/repository/ubuntu focal InRelease
Hit:6 https://repos.appviewx.com/repository/ubuntu focal-updates InRelease
Hit:7 https://repos.appviewx.com/repository/ubuntu focal-backports InRelease
Hit:8 https://repos.appviewx.com/repository/ubuntu focal-security InRelease
Get:9 https://repos.appviewx.com/repository/ubuntu jammy/main amd64 Packages [1,395 kB]
Get:10 https://repos.appviewx.com/repository/ubuntu jammy/main Translation-en [510 kB]
Get:11 https://repos.appviewx.com/repository/ubuntu jammy/main amd64 c-n-f Metadata [30.3 kB]
Get:12 https://repos.appviewx.com/repository/ubuntu jammy/restricted amd64 Packages [129 kB]
Get:13 https://repos.appviewx.com/repository/ubuntu jammy/restricted Translation-en [18.6 kB]
Get:14 https://repos.appviewx.com/repository/ubuntu jammy/restricted amd64 c-n-f Metadata [488 B]
Get:15 https://repos.appviewx.com/repository/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:16 https://repos.appviewx.com/repository/ubuntu jammy/universe Translation-en [5,652 kB]
Get:17 https://repos.appviewx.com/repository/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:18 https://repos.appviewx.com/repository/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:19 https://repos.appviewx.com/repository/ubuntu jammy/multiverse Translation-en [112 kB]
Get:20 https://repos.appviewx.com/repository/ubuntu jammy/multiverse amd64 c-n-f Metadata [8,372 B]
Get:21 https://repos.appviewx.com/repository/ubuntu jammy-updates/main amd64 Packages [2,127 kB]
Get:22 https://repos.appviewx.com/repository/ubuntu jammy-updates/main Translation-en [365 kB]
Get:23 https://repos.appviewx.com/repository/ubuntu jammy-updates/main amd64 c-n-f Metadata [17.9 kB
Get:24 https://repos.appviewx.com/repository/ubuntu jammy-updates/restricted amd64 Packages [2,605 k
Get:25 https://repos.appviewx.com/repository/ubuntu jammy-updates/restricted Translation-en [450 kB]
```

```
Get:51 https://repos.appviewx.com/repository/ubuntu jammy-security/multiverse Translation-en [7,588 B]
Get:52 https://repos.appviewx.com/repository/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [224 B]
Fetched 36.6 MB in 23s (1,626 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
660 packages can be upgraded. Run 'apt list --upgradable' to see them.
appviewx@          :/etc/pam.d$
```

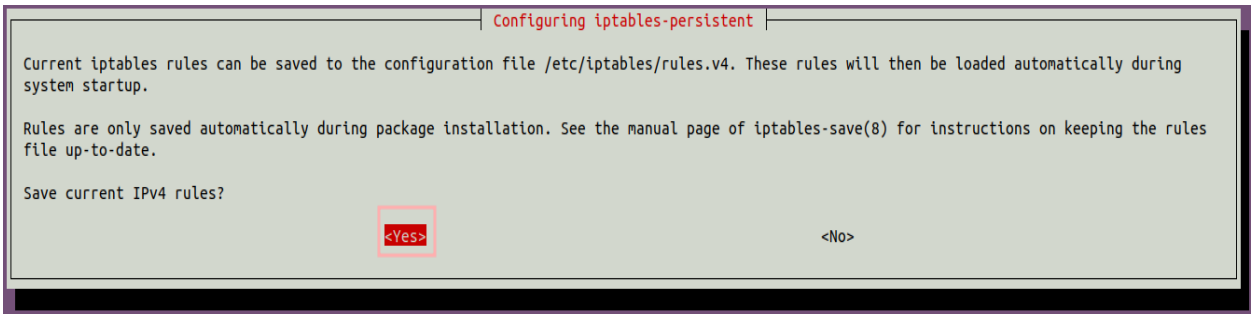
sudo apt upgrade -y

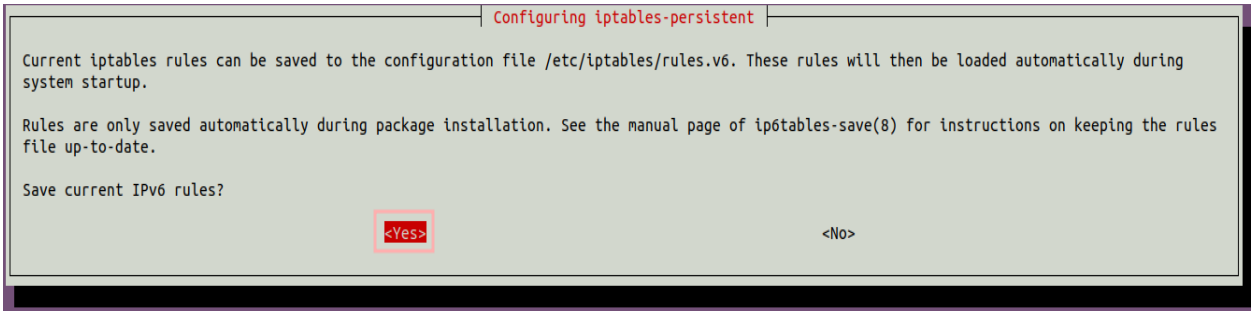
```
appviewx@          :/etc/pam.d$ sudo apt upgrade -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  adwaita-icon-theme at at-spi2-core bsdmainutils cryptsetup-run dconf-gsettings-backend dconf-service glib-networking glib-
  glib-networking-services gsettings-desktop-schemas gtk-update-icon-cache hicolor-icon-theme humanity-icon-theme libasn1-8-
  libatk-bridge2.0-0 libatk1.0-0 libatk1.0-data libatspi2.0-0 libavahi-client3 libavahi-common-data libavahi-common3 libcai
  libcbor0.6 libcolord2 libcups2 libdatrie1 libdconf1 libdeflate0 libdns-export1109 libepoxy0 libevent-2.1-7 libevent-pthrea
  libfwupdplugin1 libgdk-pixbuf-2.0-0 libgdk-pixbuf-xlib-2.0-0 libgdk-pixbuf2.0-0 libgdk-pixbuf2.0-bin libgdk-pixbuf2.0-comm
  libgssapi3-heimdal libgtk-3-0 libgtk-3-bin libgtk-3-common libharfbuzz0b libhcrypto4-heimdal libheimbase1-heimdal libheim
```

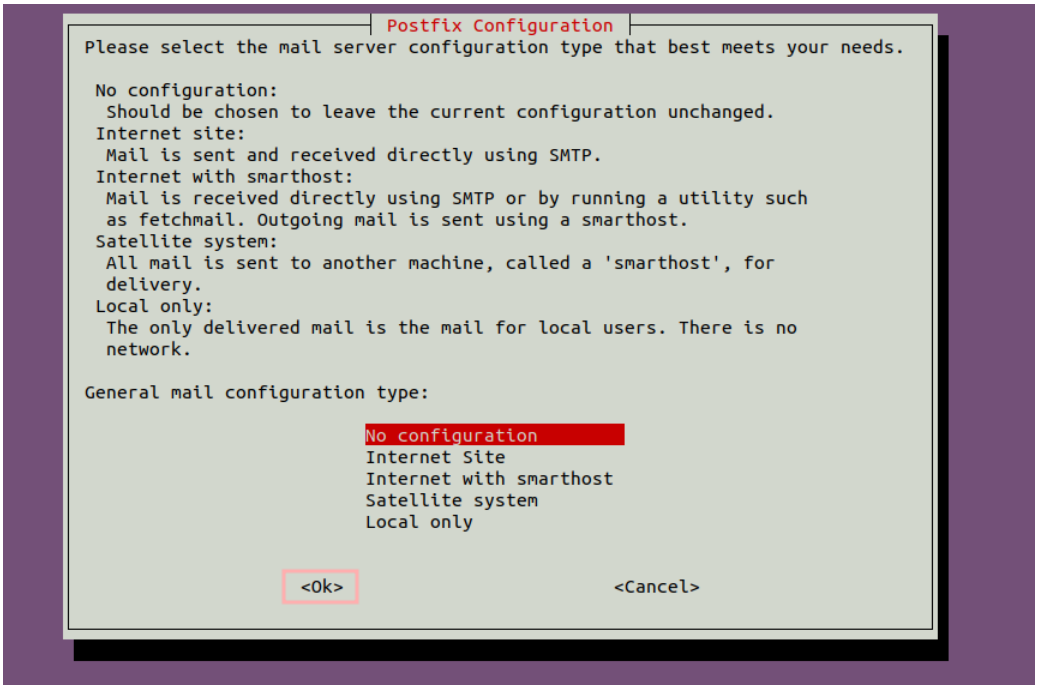
Note:

- Follow the prompts during the upgrade, choose the default options unless otherwise specified.
- Allow services to restart as required.
- The screenshots below are provided for reference.

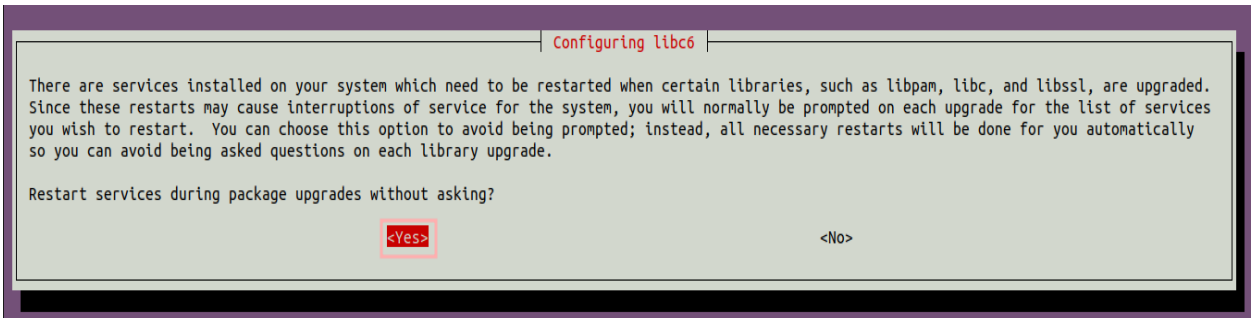
Copyright © 2025 AppViewX Inc. All Rights Reserved.



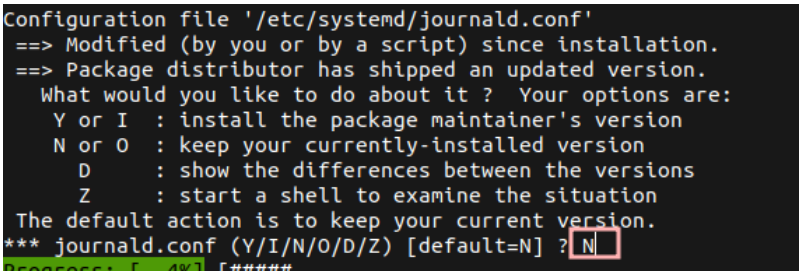


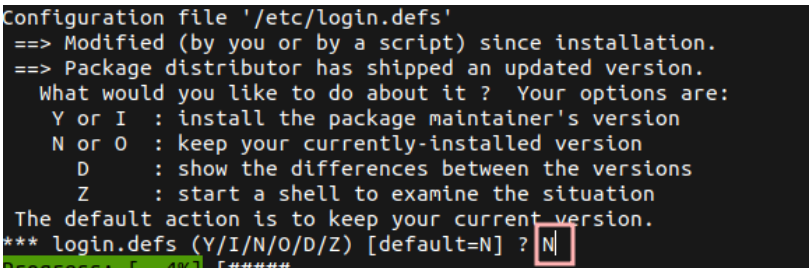


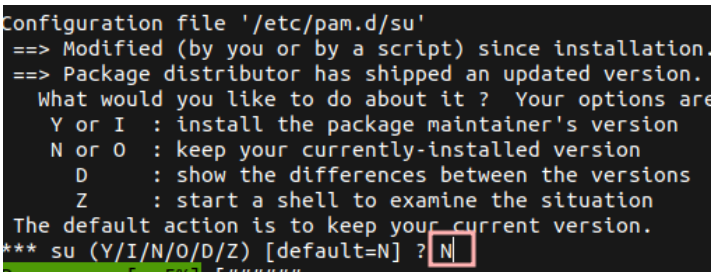
Copyright © 2025 AppViewX Inc. All Rights Reserved.

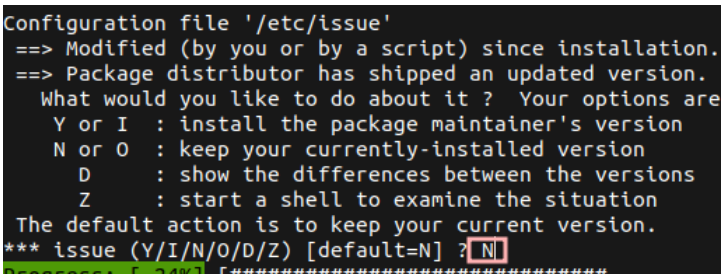


The screenshots below indicate the option to retain the existing configuration without changes, while still upgrading the necessary packages.









Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
Configuration file '/etc/issue.net'
 ==> Modified (by you or by a script) since installation
 ==> Package distributor has shipped an updated version.
   What would you like to do about it ?  Your options ar
    Y or I  : install the package maintainer's version
    N or O  : keep your currently-installed version
      D     : show the differences between the versions
      Z     : start a shell to examine the situation
 The default action is to keep your current version.
*** issue.net (Y/I/N/O/D/Z) [default=N] ? N
```

```
Configuration file '/etc/security/pwquality.conf'
 ==> Modified (by you or by a script) since installation.
 ==> Package distributor has shipped an updated version.
   What would you like to do about it ?  Your options are:
    Y or I  : install the package maintainer's version
    N or O  : keep your currently-installed version
      D     : show the differences between the versions
      Z     : start a shell to examine the situation
 The default action is to keep your current version.
*** pwquality.conf (Y/I/N/O/D/Z) [default=N] ? N
```

```
Configuration file '/etc/sudoers'
 ==> Modified (by you or by a script) since installation.
 ==> Package distributor has shipped an updated version.
   What would you like to do about it ?  Your options are:
    Y or I  : install the package maintainer's version
    N or O  : keep your currently-installed version
      D     : show the differences between the versions
      Z     : start a shell to examine the situation
 The default action is to keep your current version.
*** sudoers (Y/I/N/O/D/Z) [default=N] ? N
```

```
Configuration file '/etc/logrotate.conf'
 ==> Modified (by you or by a script) since installation.
 ==> Package distributor has shipped an updated version.
   What would you like to do about it ?  Your options are:
    Y or I  : install the package maintainer's version
    N or O  : keep your currently-installed version
      D     : show the differences between the versions
      Z     : start a shell to examine the situation
 The default action is to keep your current version.
*** logrotate.conf (Y/I/N/O/D/Z) [default=N] ? N
```

```
Configuration file '/etc/audit/auditd.conf'
 ==> Modified (by you or by a script) since installation.
 ==> Package distributor has shipped an updated version.
   What would you like to do about it ?  Your options are:
    Y or I  : install the package maintainer's version
    N or O  : keep your currently-installed version
      D     : show the differences between the versions
      Z     : start a shell to examine the situation
 The default action is to keep your current version.
*** auditd.conf (Y/I/N/O/D/Z) [default=N] ? N
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
Configuration file '/etc/audit/rules.d/audit.rules'
 ==> Modified (by you or by a script) since installation.
 ==> Package distributor has shipped an updated version.
   What would you like to do about it ?  Your options are:
    Y or I  : install the package maintainer's version
    N or O  : keep your currently-installed version
      D     : show the differences between the versions
      Z     : start a shell to examine the situation
 The default action is to keep your current version.
*** audit.rules (Y/I/N/O/D/Z) [default=N] ? N
```

```
Configuration file '/etc/sysctl.conf'
 ==> Modified (by you or by a script) since installation.
 ==> Package distributor has shipped an updated version.
   What would you like to do about it ?  Your options are:
    Y or I  : install the package maintainer's version
    N or O  : keep your currently-installed version
      D     : show the differences between the versions
      Z     : start a shell to examine the situation
 The default action is to keep your current version.
*** sysctl.conf (Y/I/N/O/D/Z) [default=N] ? N
```

```
Configuration file '/etc/default/chrony'
 ==> Modified (by you or by a script) since installation.
 ==> Package distributor has shipped an updated version.
   What would you like to do about it ?  Your options are:
    Y or I  : install the package maintainer's version
    N or O  : keep your currently-installed version
      D     : show the differences between the versions
      Z     : start a shell to examine the situation
 The default action is to keep your current version.
*** chrony (Y/I/N/O/D/Z) [default=N] ? N
```

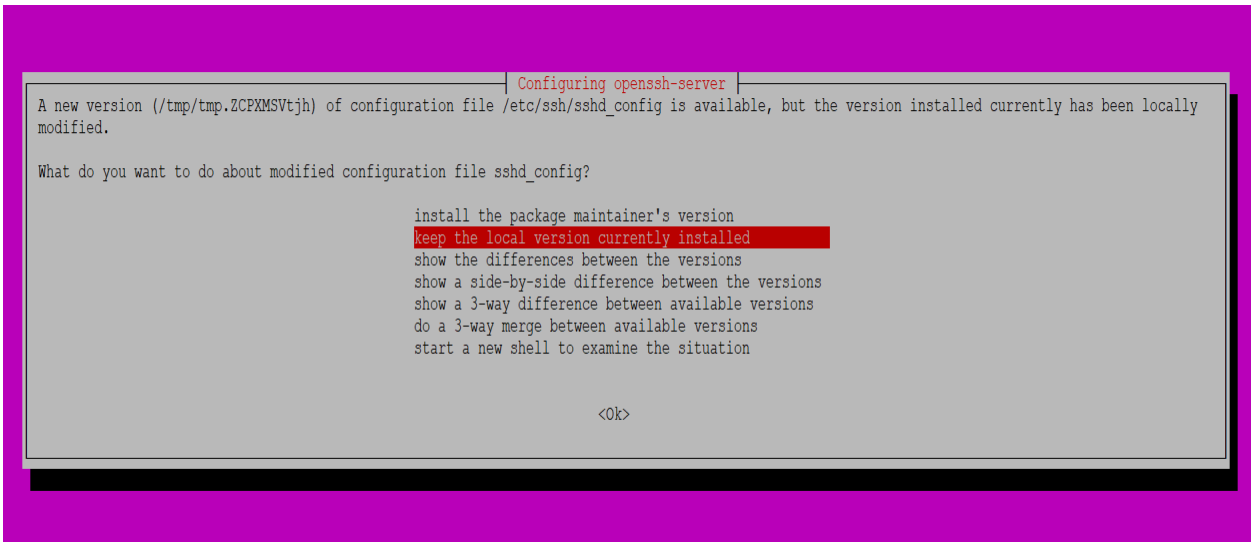
```
                        Modified configuration file
A new version (/usr/share/chrony/chrony.conf) of configuration file /etc/chrony/chrony.conf is available, but the version installed currently
has been locally modified.

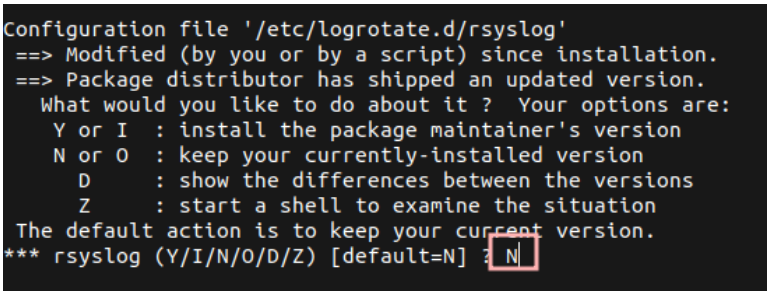
What do you want to do about modified configuration file chrony.conf?

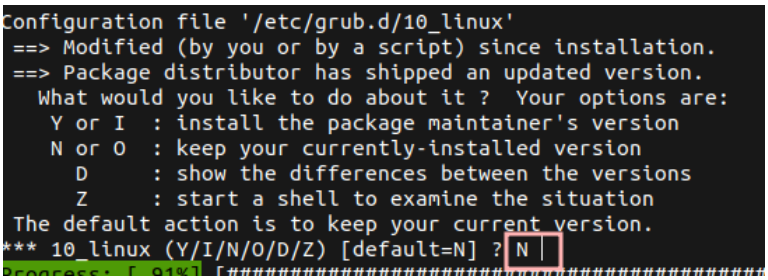
                                    install the package maintainer's version
                                    keep the local version currently installed
                                    show the differences between the versions
                                    show a side-by-side difference between the versions
                                    show a 3-way difference between available versions
                                    do a 3-way merge between available versions
                                    start a new shell to examine the situation

                                          <Ok>
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.







Post Upgrade Actions

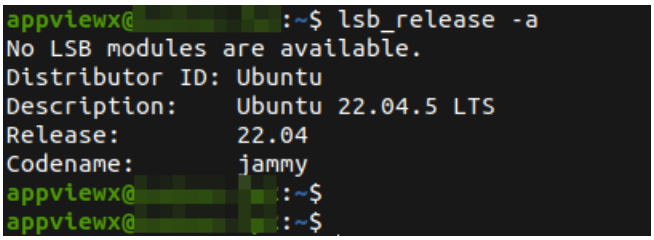
1. **Reboot the System:** Execute the command below.

   ```
   sudo reboot
   ```

2. **Verification and Validation:** After the system is rebooted, confirm the upgrade to Ubuntu 22.04 by executing the command below.

   ```
   lsb_release -a
   ```

Copyright © 2025 AppViewX Inc. All Rights Reserved.



3. **Validation for On-Premise:** After completing the upgrade for all nodes, log in to the CLI and verify that all pods in the cluster are running using the command below.

   ```
   kubectl get pods -A
   ```

   If all components are operational, access the application via the GUI to confirm a successful login.
4. **Validation for Cloud Connectors (CC):** Once the OS upgrade is complete, verify that the application is accessible through the GUI.

The system should now be successfully upgraded to Ubuntu 22.04.

Note:

- Monitor the system to ensure all applications and services function as expected.
- For assistance, contact AppViewX technical support.

Troubleshooting

If the SSH connection to a node is interrupted or disconnected during the upgrade process, access to the server can be restored by logging in again. After reconnecting, the previously active session can be resumed by running the `tmux attach` command, allowing the upgrade to continue from where it was interrupted without any loss of progress. Use the command:

```
tmux attach
```

Upgrading Ubuntu 22.04 to Ubuntu 24.04

This guide is intended for all AppViewX customers that are currently using Ubuntu 22.04 operating system. It offers step-by-step instructions for transitioning from Ubuntu 22.04 to Ubuntu 24.04, covering essential pre-upgrade preparations, updating packages, configuring necessary repositories, and resolving compatibility issues with deprecated modules for a seamless upgrade. This process applies specifically to AppViewX-shipped OVAs.

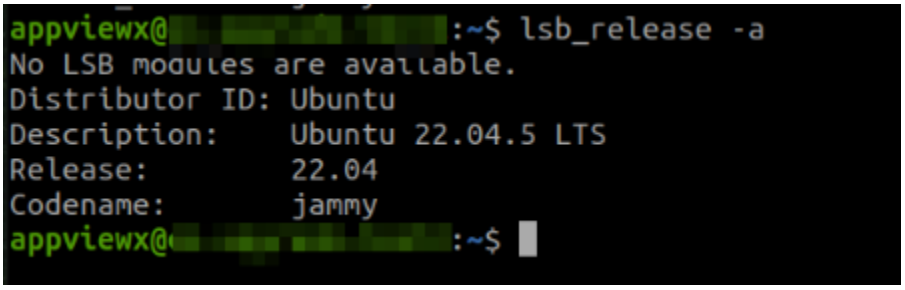
Copyright © 2025 AppViewX Inc. All Rights Reserved.

• Initiating the Upgrade Process
• Post Upgrade Actions
• Troubleshooting

Pre-Upgrade Considerations

Before starting the upgrade, ensure the following are addressed:

- **Backup:** Take a VM snapshot before starting the upgrade to enable rollback in case of issues.
- **Internet Connectivity:** Ensure the node has internet or proxy access to download the OS update from `repos.appviewx.com`.
- **Disk Space:** Verify that at least 3GB of free space is available in the root (*/*) partition.
- **Current Version Check:**Confirm that the system is running Ubuntu 22.04 by executing the following command:
  - `lsb_release -a`



Prerequisite

1. Stop AppViewX on All Nodes (On-Premise Setups Only).

> **Note:** This step is applicable only to on-premise setups. Cloud Connectors (CC) do not require component stoppage.

a. Log in to the AppViewX installer node and navigate to the appviewx_kubernetes/scripts directory.

b. If you are upgrading the OS **with a downtime**, then use the following command to stop the AppViewX on all the nodes.

i. For **single-node**:

```
./appviewx.sh --stop <hostname>
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

ii. For **multi-node**:

```
./appviewx.sh --stop -all
```

c. If you are upgrading the OS **without a downtime**, then use the following command to stop the AppViewX nodes.

- For multi-node - This applies only to multi-node clusters. For multi-master clusters, ensure that the master load balancer is properly configured to prevent downtime.

```
./appviewx.sh –stop <hostname>
```

- Before proceeding to the next node, upgrade the OS and verify that the components are running on the node using:

```
kubectl get pods -A -o wide | grep -i <hostname>
```

> **Note:** Use VMware Console access for this upgrade. If upgrading over SSH, use "**screen**" or "**tmux**" to prevent interruptions in case of disconnection.

2. Check if the **tmux** package is installed on your node. If not, install it using

```
sudo apt install tmux
```

3. Attach to a tmux session by running:

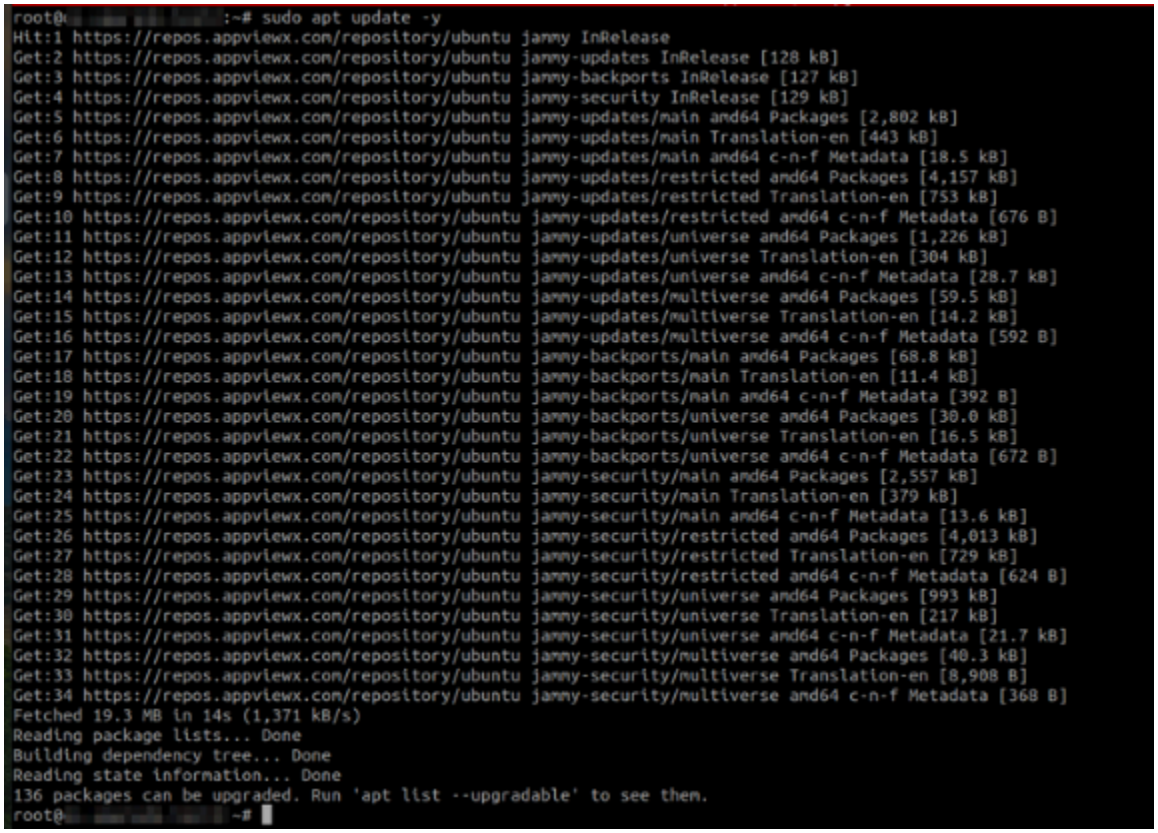
```
tmux
```

Preparing for the Upgrade

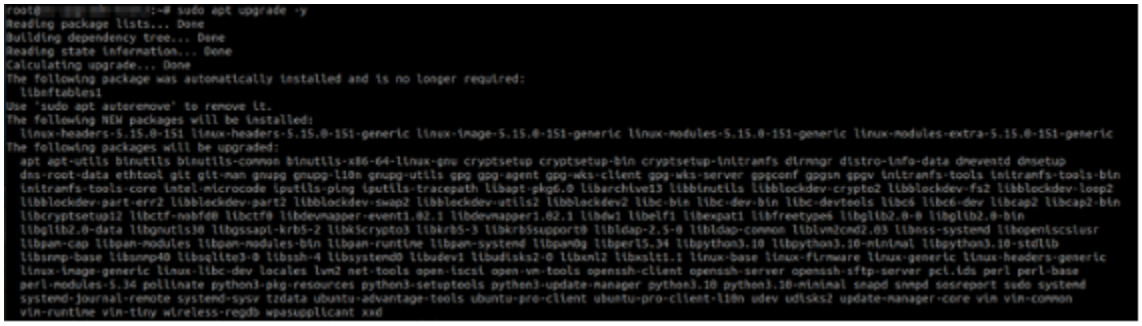
To prepare for the upgrade, update all installed packages on Ubuntu 22.04. Execute the following commands:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- sudo apt update -y



- sudo apt upgrade -y



Copyright © 2025 AppViewX Inc. All Rights Reserved.

Note: If there is an error on executing the above command as shown below then execute the following command:

```
160 packages can be upgraded. Run 'apt list --upgradable' to see them.
appviewx@ts-apvx-labnode-16:~/Thames_Install/appviewx_kubernetes/scripts$ sudo apt upgrade -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
You might want to run 'apt --fix-broken install' to correct these.
The following packages have unmet dependencies:
 libstdc++6 : Depends: gcc-10-base (= 10.5.0-1ubuntu1~20.04) but 10.3.0-1ubuntu1~20.04 is installed
 python3-setuptools : Depends: python3-pkg-resources (= 45.2.0-1ubuntu0.1) but 45.2.0-1 is installed
E: Unmet dependencies. Try 'apt --fix-broken install' with no packages (or specify a solution).
```

```
sudo apt --fix-broken install -y
```

After the `apt --fix-broken install` command executes successfully, execute the `apt upgrade` command once again.

```
sudo apt upgrade -y
```

- `sudo apt full-upgrade`

```
root@                    :~# sudo apt full-upgrade -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  libnftables1 linux-headers-5.15.0-131
Use 'sudo apt autoremove' to remove them.
The following packages will be REMOVED:
  linux-headers-5.15.0-131-generic linux-image-5.15.0-131-generic linux-modules-5.15.0-131-generic linux-modules-extra-5.15.0-131-generic
0 upgraded, 0 newly installed, 4 to remove and 0 not upgraded.
After this operation, 506 MB disk space will be freed.
```

Configuring the AppViewX Repository

1. Validate the authentication for the repos.appviewx.com repository using the command below.

```
cat /etc/apt/auth.conf
```

2. Create a new APT Repository; add the AppViewX repository for Ubuntu 24.04 (Noble Numbat).

```
sudo vi /etc/apt/sources.list.d/appviewx_repo-noble.sources
```

```
root@                    :~# cat /etc/apt/sources.list.d/appviewx_repo-noble.sources
Types: deb
URIs: https://repos.appviewx.com/repository/ubuntu/
Suites: noble noble-updates noble-backports
Components: main restricted universe multiverse
Signed-By: /usr/share/keyrings/ubuntu-archive-keyring.gpg

Types: deb
URIs: https://repos.appviewx.com/repository/ubuntu/
Suites: noble-security
Components: main restricted universe multiverse
Signed-By: /usr/share/keyrings/ubuntu-archive-keyring.gpg
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Add the following lines to the file:

```
# AppViewX APT repository
Types: deb
URIs: https://repos.appviewx.com/repository/ubuntu/
Suites: noble noble-updates noble-backports
Components: main restricted universe multiverse
Signed-By: /usr/share/keyrings/ubuntu-archive-keyring.gpg

Types: deb
URIs: https://repos.appviewx.com/repository/ubuntu/
Suites: noble-security
Components: main restricted universe multiverse
Signed-By: /usr/share/keyrings/ubuntu-archive-keyring.gpg
```

Handling Deprecated Modules

To avoid issues with deprecated pam_tally and pam_tally2 modules, replace them with pam_faillock using the following commands:

```
cd /etc/pam.d
```

```
sudo sed -i 's/pam_tally2\?/pam_faillock/g' /etc/pam.d/*
```

Initiating the Upgrade Process

Execute the following commands:

```
sudo apt update -y
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
root@ :~# sudo apt update -y
Get:1 https://repos.appviewx.com/repository/ubuntu noble InRelease [256 kB]
Get:2 https://repos.appviewx.com/repository/ubuntu noble-updates InRelease [126 kB]
Get:3 https://repos.appviewx.com/repository/ubuntu noble-backports InRelease [126 kB]
Get:4 https://repos.appviewx.com/repository/ubuntu noble-security InRelease [126 kB]
Hit:5 https://repos.appviewx.com/repository/ubuntu jammy InRelease
Get:6 https://repos.appviewx.com/repository/ubuntu jammy-updates InRelease [128 kB]
Hit:7 https://repos.appviewx.com/repository/ubuntu jammy-backports InRelease
Get:8 https://repos.appviewx.com/repository/ubuntu jammy-security InRelease [129 kB]
Get:9 https://repos.appviewx.com/repository/ubuntu noble/main amd64 Packages [1,401 kB]
Get:10 https://repos.appviewx.com/repository/ubuntu noble/main Translation-en [513 kB]
Get:11 https://repos.appviewx.com/repository/ubuntu noble/main amd64 c-n-f Metadata [30.5 kB]
Get:12 https://repos.appviewx.com/repository/ubuntu noble/restricted amd64 Packages [93.9 kB]
Get:13 https://repos.appviewx.com/repository/ubuntu noble/restricted Translation-en [18.7 kB]
Get:14 https://repos.appviewx.com/repository/ubuntu noble/restricted amd64 c-n-f Metadata [416 B]
Get:15 https://repos.appviewx.com/repository/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:16 https://repos.appviewx.com/repository/ubuntu noble/universe Translation-en [5,982 kB]
Get:17 https://repos.appviewx.com/repository/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:18 https://repos.appviewx.com/repository/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:19 https://repos.appviewx.com/repository/ubuntu noble/multiverse Translation-en [118 kB]
Get:20 https://repos.appviewx.com/repository/ubuntu noble/multiverse amd64 c-n-f Metadata [8,328 B]
Get:21 https://repos.appviewx.com/repository/ubuntu noble-updates/main amd64 Packages [1,314 kB]
Get:22 https://repos.appviewx.com/repository/ubuntu noble-updates/main Translation-en [264 kB]
Get:23 https://repos.appviewx.com/repository/ubuntu noble-updates/main amd64 c-n-f Metadata [13.5 kB]
Get:24 https://repos.appviewx.com/repository/ubuntu noble-updates/restricted amd64 Packages [1,645 kB]
Get:25 https://repos.appviewx.com/repository/ubuntu noble-updates/restricted Translation-en [359 kB]
```

```
Get:51 https://repos.appviewx.com/repository/ubuntu noble-security/multiverse Translation-en [4,288 B]
Get:52 https://repos.appviewx.com/repository/ubuntu noble-security/multiverse amd64 c-n-f Metadata [380 B]
Fetched 34.1 MB in 19s (1,774 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
621 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ :~#
```

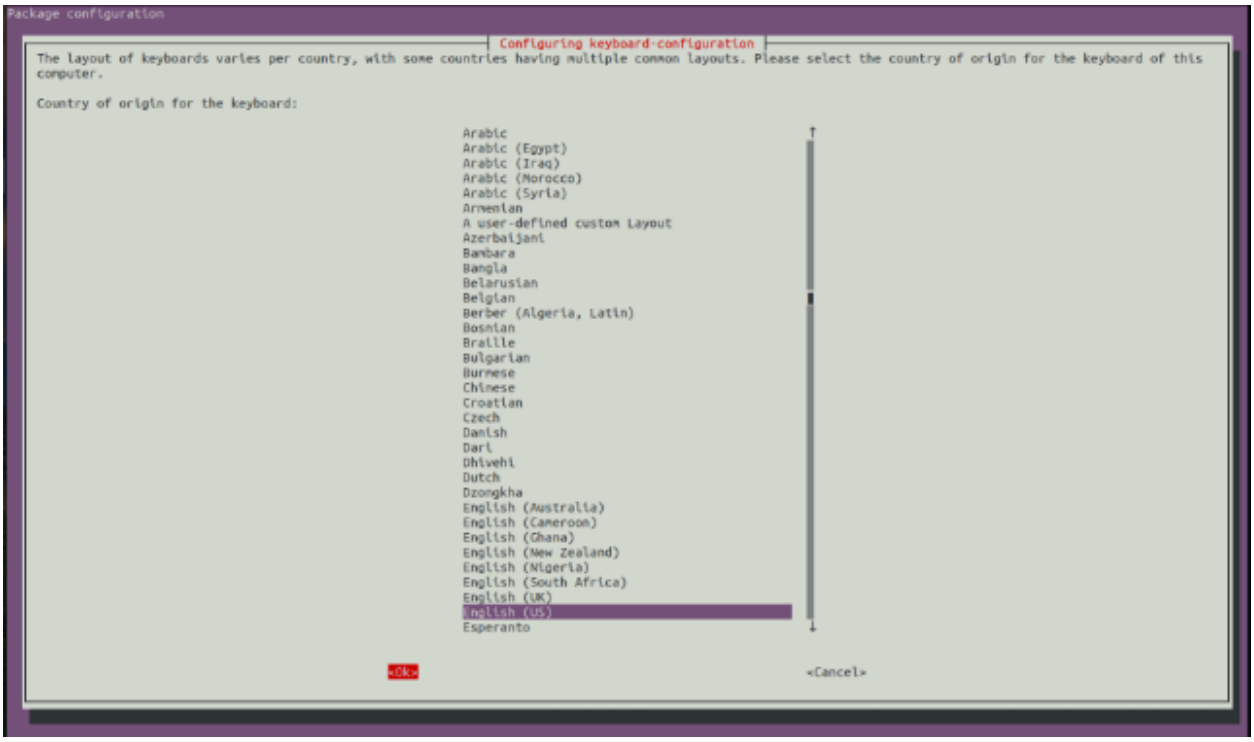
sudo apt upgrade -y

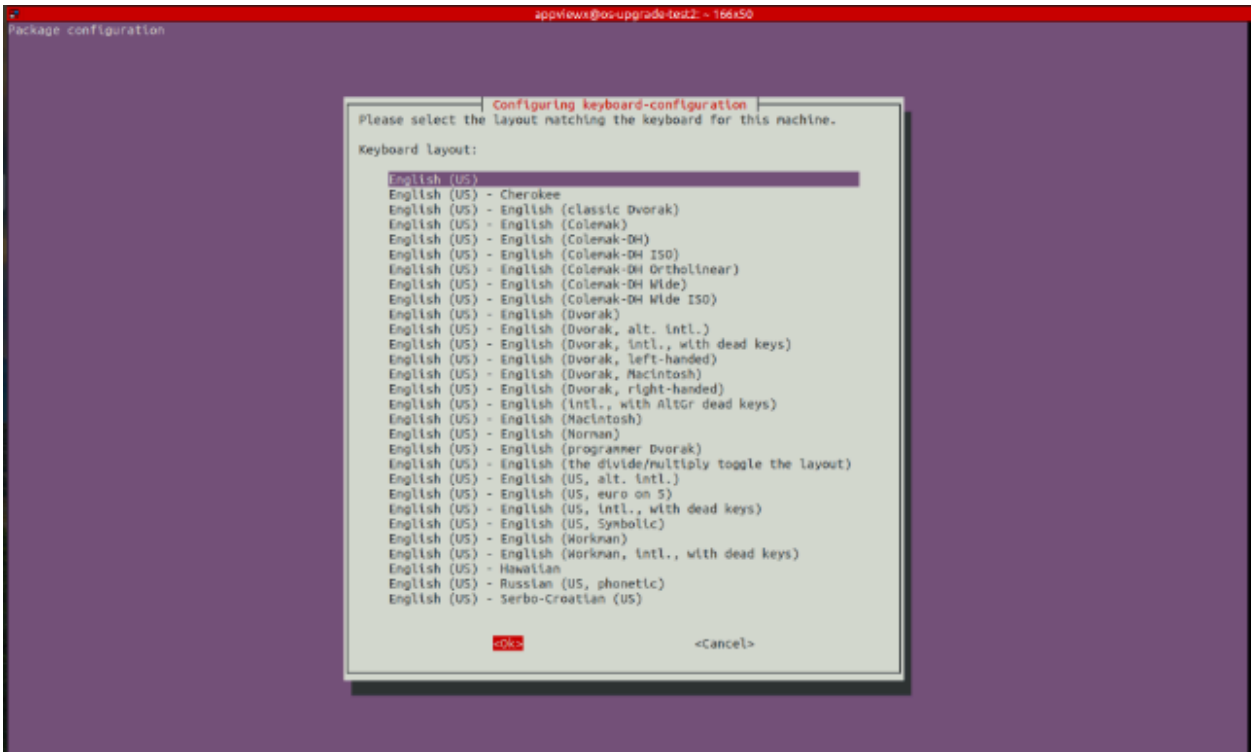
```
root@ :~# sudo apt upgrade -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  libnftables1 linux-headers-5.15.0-131 python3-distutils python3-lib2to3
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  cron-daemon-common fonts-dejavu-mono gcc-14-base libplist-2.0-4 libsharpyuv0 libunistring5 python3-boto3 python3-botocore python3-certifi python3-dateutil
  python3-jmespath python3-requests python3-s3transfer python3-urllib3
The following packages have been kept back:
  acl alde amd64-microcode apparmor apparmor-utils apt apt-utils audispd-plugins auditd bash bc bind9-dnsutils bind9-host bind9-libs bind9-utils binutils
  binutils-common binutils-x86-64-linux-gnu bolt bsd-mailx bsdextrautils bsdutils btrfs-progs build-essential busybox-initramfs chrony coreutils cpio cpp cpp-11
  cryptsetup-bin cryptsetup-initramfs curl dash dbus dbus-user-session debianutils debugedit diffutils dirmngr dmeventd dnsetup dnsmasq-base dosfstools dpkg dwz
  e2fsprogs eject fakeroot fdisk file findutils fuse3 fwupd g++ g++-11 gawk gcc gcc-11 gcc-11-base gcc-12-base gdisk gettext gettext-base gir1.2-glib-2.0
```

Note:

- Follow the prompts during the upgrade, choose the default options unless otherwise specified.
- Allow services to restart as required.
- The screenshots below are provided for reference.

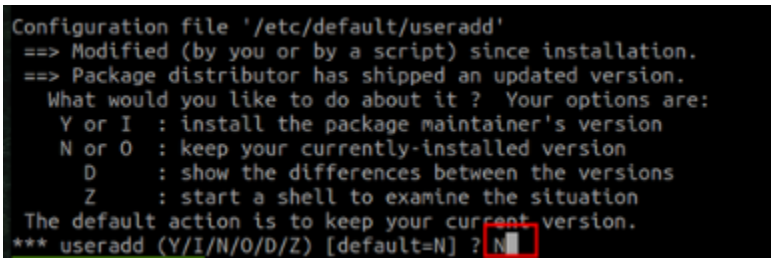
Copyright © 2025 AppViewX Inc. All Rights Reserved.

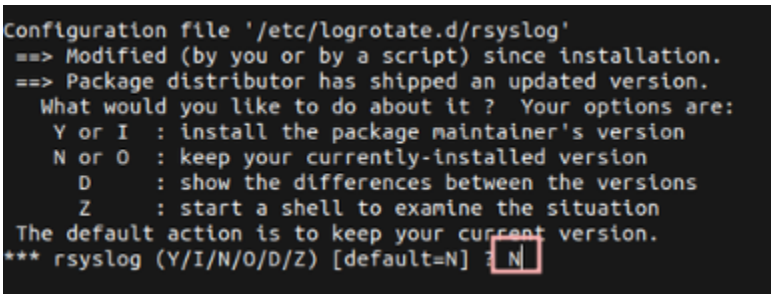


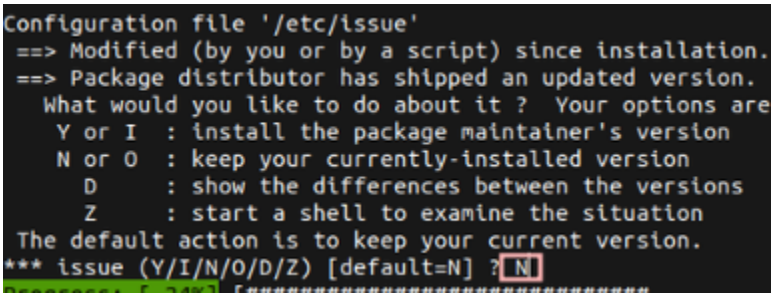


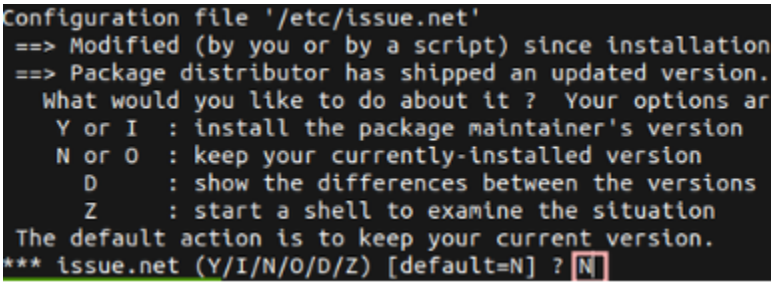
The screenshots below indicate the option to retain the existing configuration without changes, while still upgrading the necessary packages.

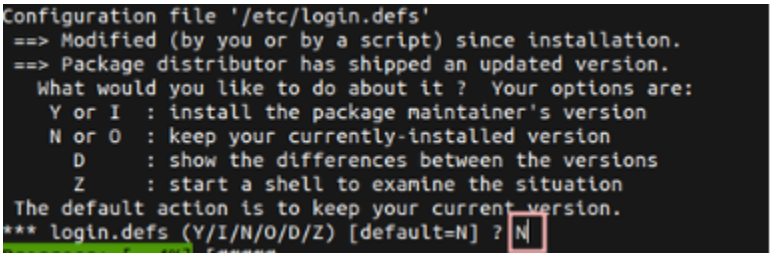
Copyright © 2025 AppViewX Inc. All Rights Reserved.



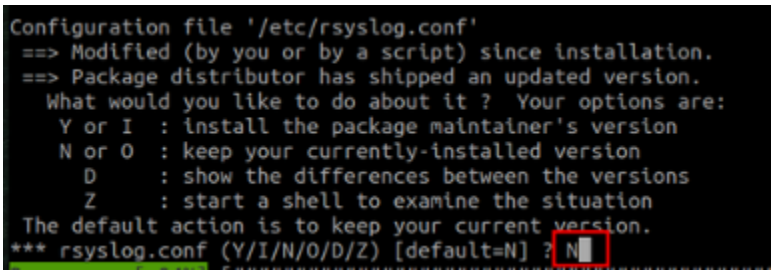


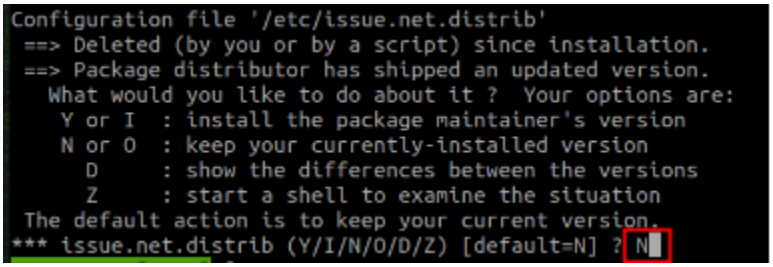


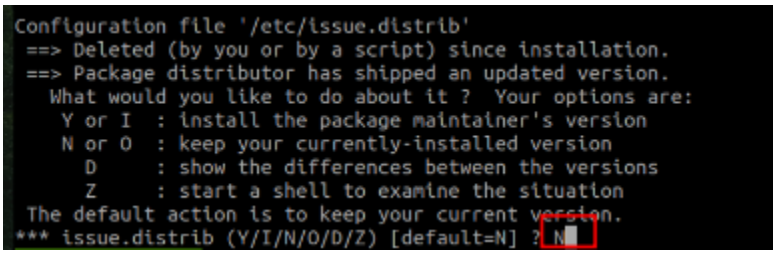


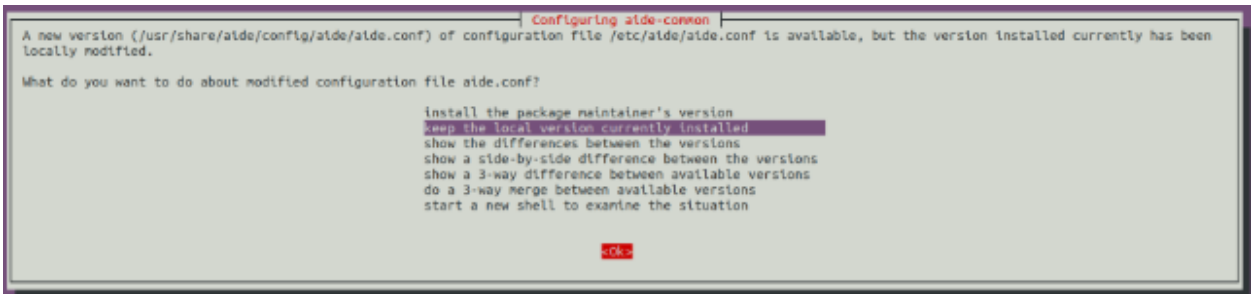


Copyright © 2025 AppViewX Inc. All Rights Reserved.









Post Upgrade Actions

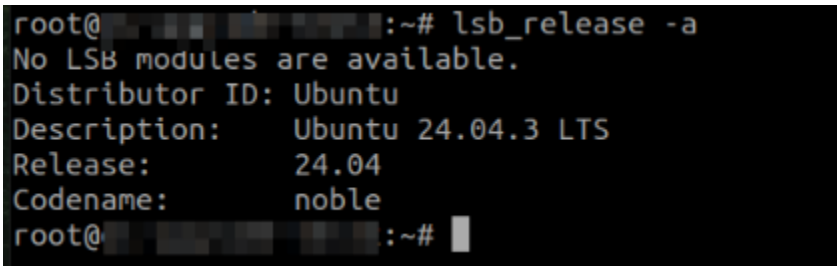
1. **Reboot the System:** Execute the command below.

   ```
   sudo reboot
   ```

2. **Verification and Validation:** After the system is rebooted, confirm the upgrade to Ubuntu 24.04 by executing the command below.

   ```
   lsb_release -a
   ```

Copyright © 2025 AppViewX Inc. All Rights Reserved.



3. **Validation for On-Premise:** After completing the upgrade for all nodes, log in to the CLI and verify that all pods in the cluster are running using the command below.

   ```
   kubectl get pods -A
   ```

   If all components are operational, access the application via the GUI to confirm a successful login.
4. **Validation for Cloud Connectors (CC):** Once the OS upgrade is complete, verify that the application is accessible through the GUI.

The system should now be successfully upgraded to Ubuntu 24.04.

Note:

- Monitor the system to ensure all applications and services function as expected.
- For assistance, contact AppViewX technical support.

Troubleshooting

If the SSH connection to a node is interrupted or disconnected during the upgrade process, access to the server can be restored by logging in again. After reconnecting, the previously active session can be resumed by running the `tmux attach` command, allowing the upgrade to continue from where it was interrupted without any loss of progress. Use the command:

```
tmux attach
```

OS Patching on Ubuntu for Multi-Node Environment

This section provides a step-by-step guide for applying OS patches in an Ubuntu environment while minimizing downtime in a multi-node, multi-data center (DC) Kubernetes setup. By following the rolling update procedure, OS patches can be applied without causing significant disruption to the overall environment. Ensure the GUI load balancer is properly configured to distribute traffic during the

Copyright © 2025 AppViewX Inc. All Rights Reserved.

patching process to maintain high availability. As per our high-availability (HA) claim, a brief downtime of approximately 5 minutes is expected when the secondary DC nodes are down.

• Prerequisites
• OS Patching Procedure for Minimal Downtime

Prerequisites

- Ensure your system has access to the required package repositories by either enabling APT Proxy or whitelisting the AppViewX repository (**repo.appviewx.com**).
- For multi-master Kubernetes environments, configure the TCP load balancer to handle failover effectively, ensuring minimal downtime in case the primary master becomes unavailable during the upgrade.
- Verify that the installed user has **sudo** privileges.
- Take VM snapshots before proceeding with the OS patching.

OS Patching Procedure for Minimal Downtime

To minimize application downtime in environments spanning multiple data centers, it is recommended to perform a rolling update. This process involves stopping the application on one node at a time, applying the OS patches, and then restarting the node before moving to the next one.

1. Log in to the installer node and navigate to <appviewx installer directory>/scripts.
2. *Stop the application on the node* - Before applying the patch, stop the application on the specific node. This ensures that the node is not running any critical services during the patching process.
   Run the following command to stop the application:

   ```
   ./appviewx.sh --stop <node_name>
   ```

   Note: Replace **<node_name>** with the actual name of the node you are stopping.

3. *Apply the OS patch/security update* - Once the application is stopped, proceed with updating the operating system. Run the following commands to apply the necessary patches:
   Run the following command to stop the application:

   ```
   sudo apt update -y
   ```

   ```
   sudo apt upgrade -y
   ```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

This will update the package lists and apply all available OS security patches.

4. *Reboot the node* - After the patching is complete, reboot the node to ensure that all changes are applied properly.

```
sudo reboot
```

5. *Start the application on the node* - After the node has rebooted, start the application back up using the following command:

```
./appviewx.sh --start <node_name>
```

> **Note:** Replace **<node_name>** with the actual name of the node.

6. *Repeat for the remaining nodes in the data center* - Repeat the above steps for all the nodes in the first data center. Ensure that each node is patched one at a time to avoid any application downtime within the data center.
7. *Move to the next data center* - Once all nodes in the first data center have been patched and verified to be working, proceed to the second data center and follow the same steps for each node.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Chapter 2: AppViewX SaaS Setup Guides

- SaaS Architecture Guide
- AppViewX SaaS Onboarding and Getting Started Guide
- AppViewX Cloud Connector User Guide

SaaS Architecture Guide

This guide will walk you through the architecture used by AppViewX to implement SaaS. It covers multi-tenant architecture, network architecture, and cluster architecture. Information with respect to scaling of clusters, DB isolation for each tenant, and high availability of AppViewX with the help of this architecture has been touched upon in this guide.

- Key Highlights of AppViewX Software as a Service
- AppViewX Architecture
- Multi-Tenancy Architecture
- SaaS Deployment Architecture

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 products:

- CERT+, 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.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- Provision certificates for devices and applications.
- Save resources, time, and effort of installation and maintenance.

For details, refer the CERT+ User Guide.

- ADC+, 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+ 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.
- SSH+, which lets you:
  - Discover and display SSH certificates alongside SSH keys, offering a more comprehensive overview of your security credentials.
  - Download keys for key-based access control, ensuring streamlined access management.
  - Specify access duration in either hours or days when requesting access to an infrastructure group, providing enhanced access management control.
  - Use a dynamic access flow that adapts to either key or certificate-based access, depending on the user's selected 'Access Mode' during host addition.
  - Rotate host certificates effortlessly, directly from the host inventory, promoting secure host certificate management.
  - Revoke SSH certificates directly, thus enhancing security control.
  - Choose between 'Key' and 'Certificate' access modes during host addition, with the 'Certificate' option being pre-selected by default.
  - Rotate and delete keys from hosts with multiple keys through the user and host key age report.

  For details, refer the SSH+ User Guide.
- SIGN+, which lets you:
  - Simplify Code Signing Certificate enrollment and Certificate Lifecycle Management (CLM) operations.
  - Customize signing policies according to your requirements
  - Integrate with AppViewX's customized Cryptographic Service Provider (CSP) and PKCS#11 for enhanced security.
  - Manage your code signing inventory with a full suite of tools and features.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- Sign your code effortlessly using a variety of tools including SignTool, JSign, JarSigner, APKSigner, Mage, and Nuget.
- Ensure compatibility with third-party Timestamp Authorities (TSA) for a wider range of options.

For details, refer the SIGN+ User Guide.

- KUBE+, which lets you:
  - Simplify Certificate Lifecycle Management for Kubernetes workloads.
  - Get real-time visibility, central audit, and governance over K8's Certs.
  - Achieve end-to-end automated certificate enrollment process.
  - Have secure and compliant PKI across K8s workloads (secrets, pods, and service mesh).

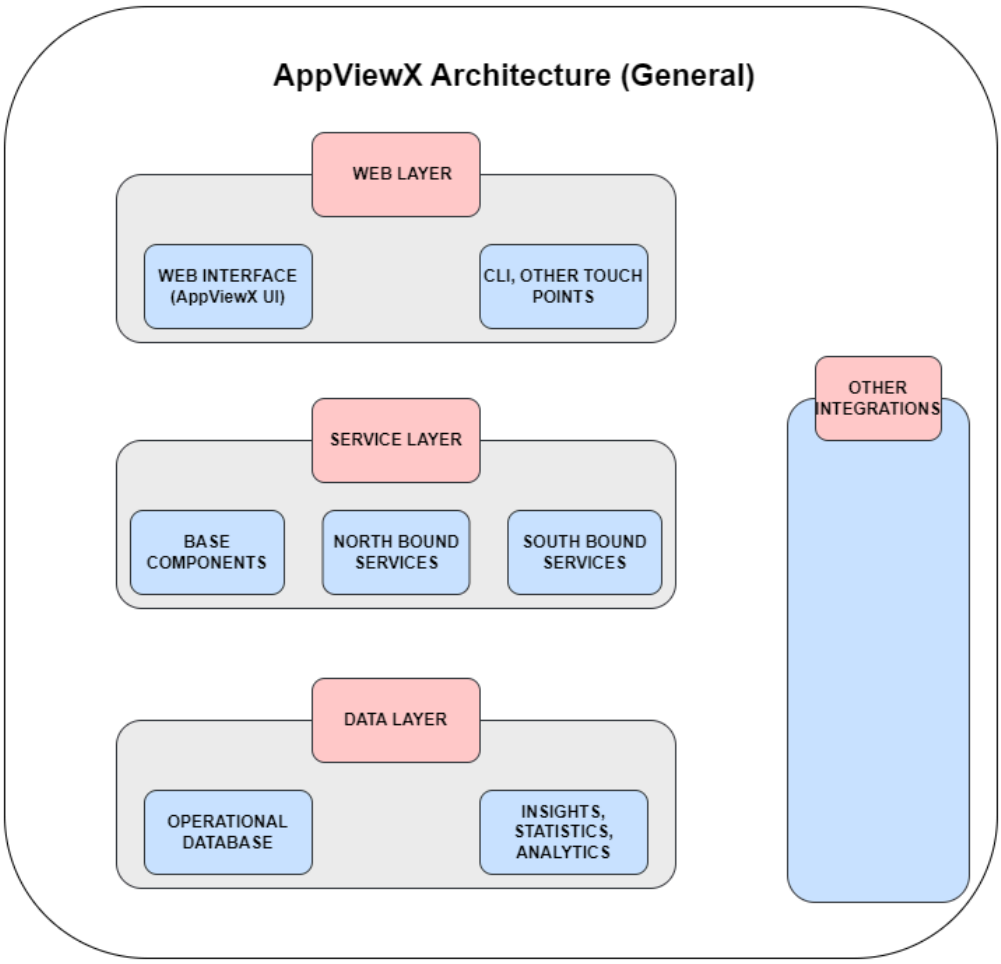
For details, refer the KUBE+ User Guide.

AppViewX Architecture

AppViewX is designed based on microservice architecture and its deployed on Kubernetes, an open-source platform for deploying and managing containers. The microservice architecture of AppViewX makes it easier to move to containerized workloads and the containers being orchestrated using Kuberneters. Kubernetes provides container runtime, orchestration, self-healing mechanisms, service discovery and load balancing and its used for the deployment, scaling, management, and composition of application containers across clusters.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

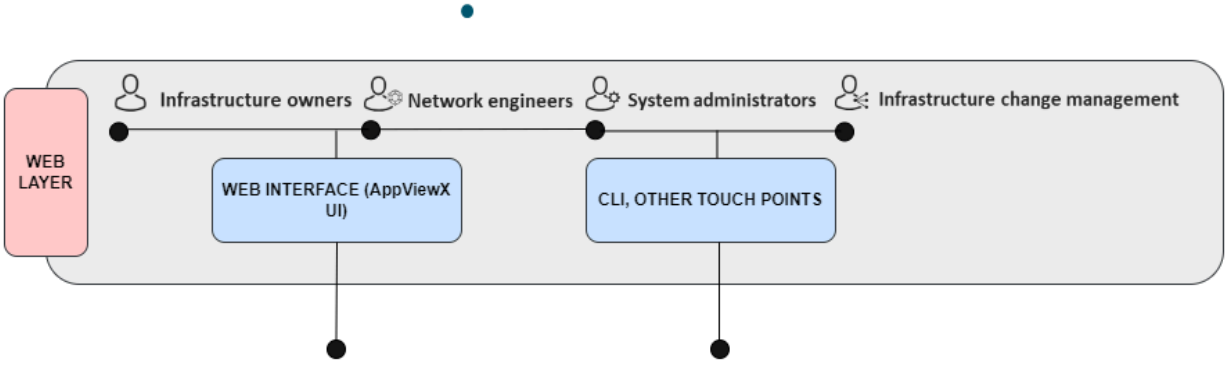
High Level Architecture



Understanding the AppViewX Architecture

Web Layer

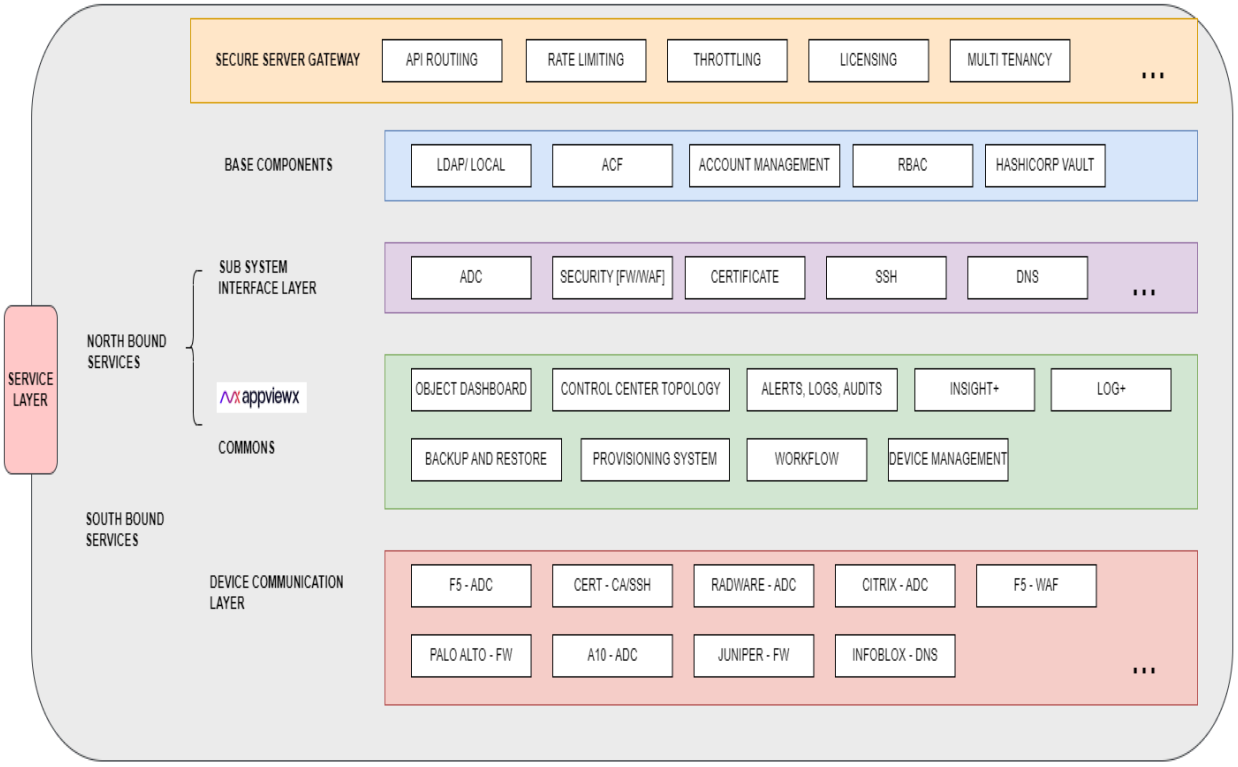
The web layer includes services for user interaction.



Copyright © 2025 AppViewX Inc. All Rights Reserved.

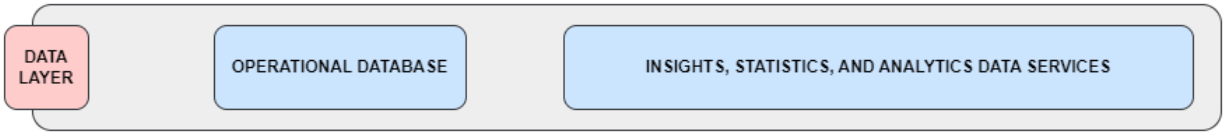
Service Layer

The service layer houses the core AppViewX business logic that is responsible for fetching user inputs from the UI. The AppViewX application then uses these inputs to perform CLM operations for the end devices. The responses thus received are persisted in the database.



Data Layer

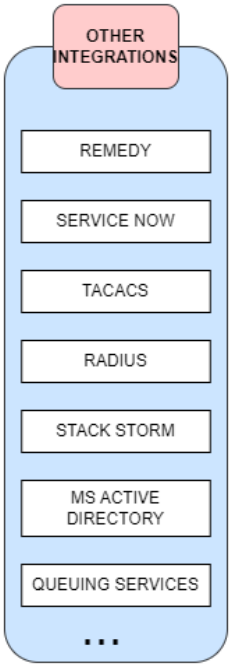
The data layer houses the persistence logic for the application. The data persistence logic is responsible for backing up the data in the application's file system. In events of a data loss, the data can then be retrieved from the file system.



Other Integrations

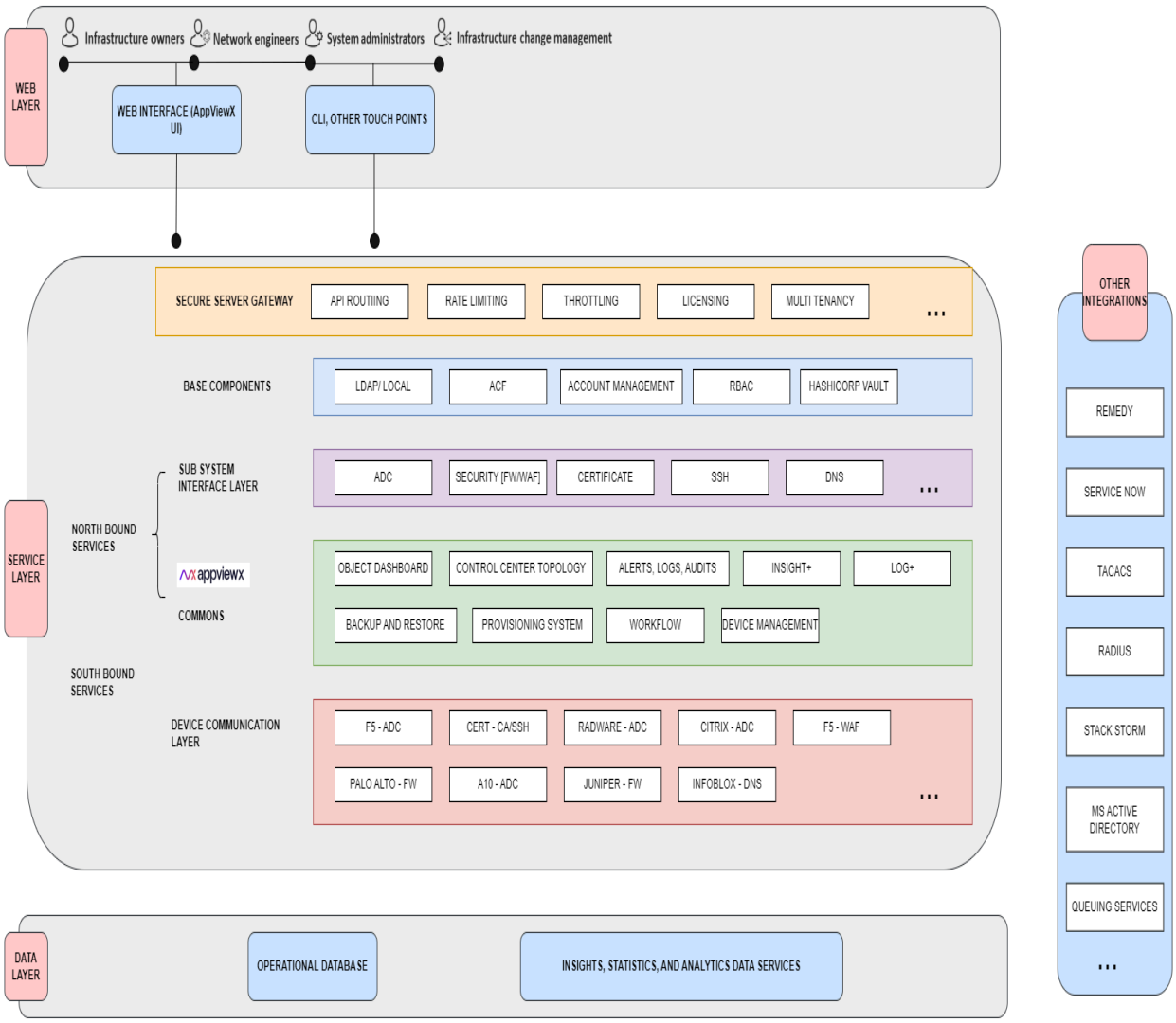
Other Integrations are out-of-the-box ticketing, authorization, and authentication tools that AppViewX supports integration with.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



Copyright © 2025 AppViewX Inc. All Rights Reserved.

Low Level Architecture



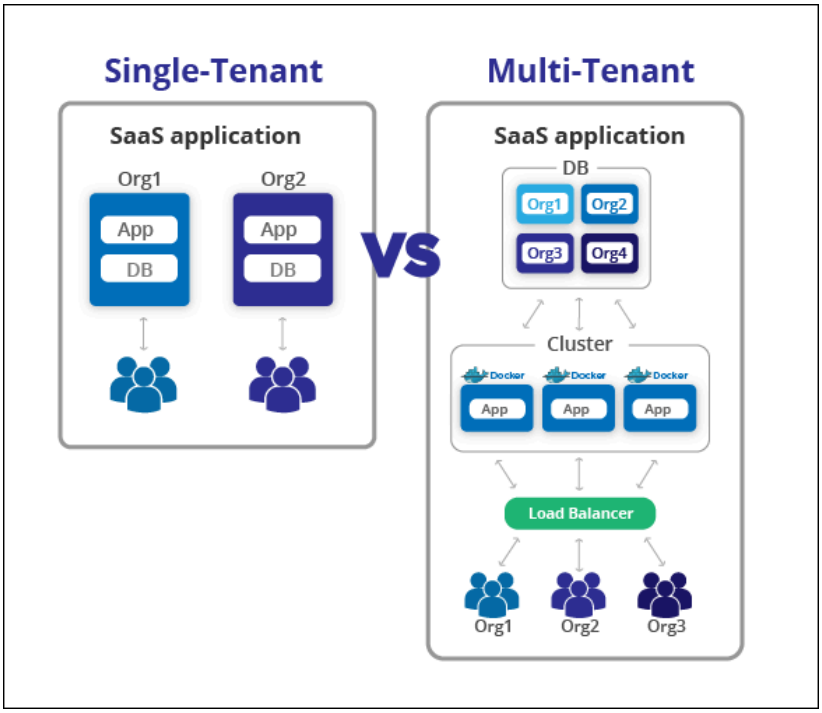
Multi-Tenancy Architecture

In a multi-tenancy architecture, a single instance of the software serves multiple accounts/customers. In this setup, the same resources -compute, networking and storage - are shared on the cloud among tenants.

In this ecosystem, a single environment can serve multiple tenants utilising a scalable, available, and resilient architecture. The underlying infrastructure is completely shared, logically isolated, and with fully centralised services.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

AppViewX multi-tenant architecture is enabled by a shared compute cluster or a workload cluster where the workloads run and a database cluster where the actual tenant isolation happens by allocating a dedicated schema for each and every tenant. The diagram below depicts the multi-tenant implementation.



Multi-Tenant Architecture

SaaS Deployment Architecture

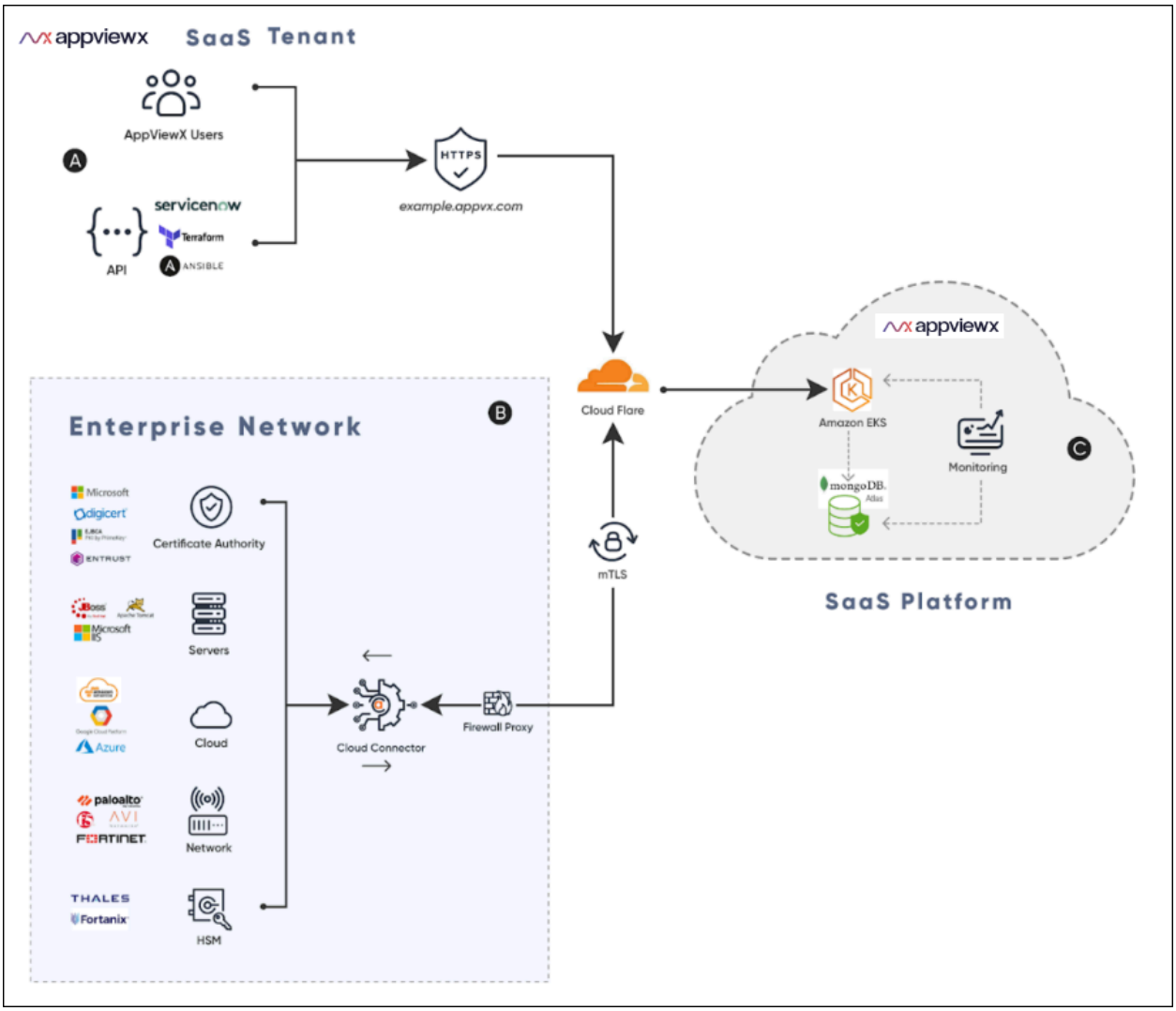
• SaaS Deployment Architecture
• Architecture Components Overview

SaaS Deployment Architecture

The AppViewX SaaS deployment architecture is a cloud-based deployment with the following benefits.

- Lower cost of ownership (TCO), significantly reduced maintenance.
- Guaranteed availability (SLA), and enhanced data security
- Faster release cycles and upgrades to access new offerings.
- Avoid installation of the entire AppViewX infrastructure in the tenant network.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



AppViewX Multi-Tenancy Architecture

At a high level, the SaaS deployment architecture consists of:

- A public access to the AppViewX SaaS products' tenant secured via https.
- AppViewX Cloud Connector (a lightweight proxy) deployment that enables connectivity between the SaaS platform to the Enterprise network thereby ensuring faster value realisation of critical Certificate Lifecycle Management (CLM) functions such as Discovery, visibility, Automation and self-servicing of SSL / TLS certificates from the tenants infrastructure.
- AppViewX SaaS platform that enables the server-side components such as database, compute, monitoring and tenant provisioning (which includes install and upgrades).

Copyright © 2025 AppViewX Inc. All Rights Reserved.

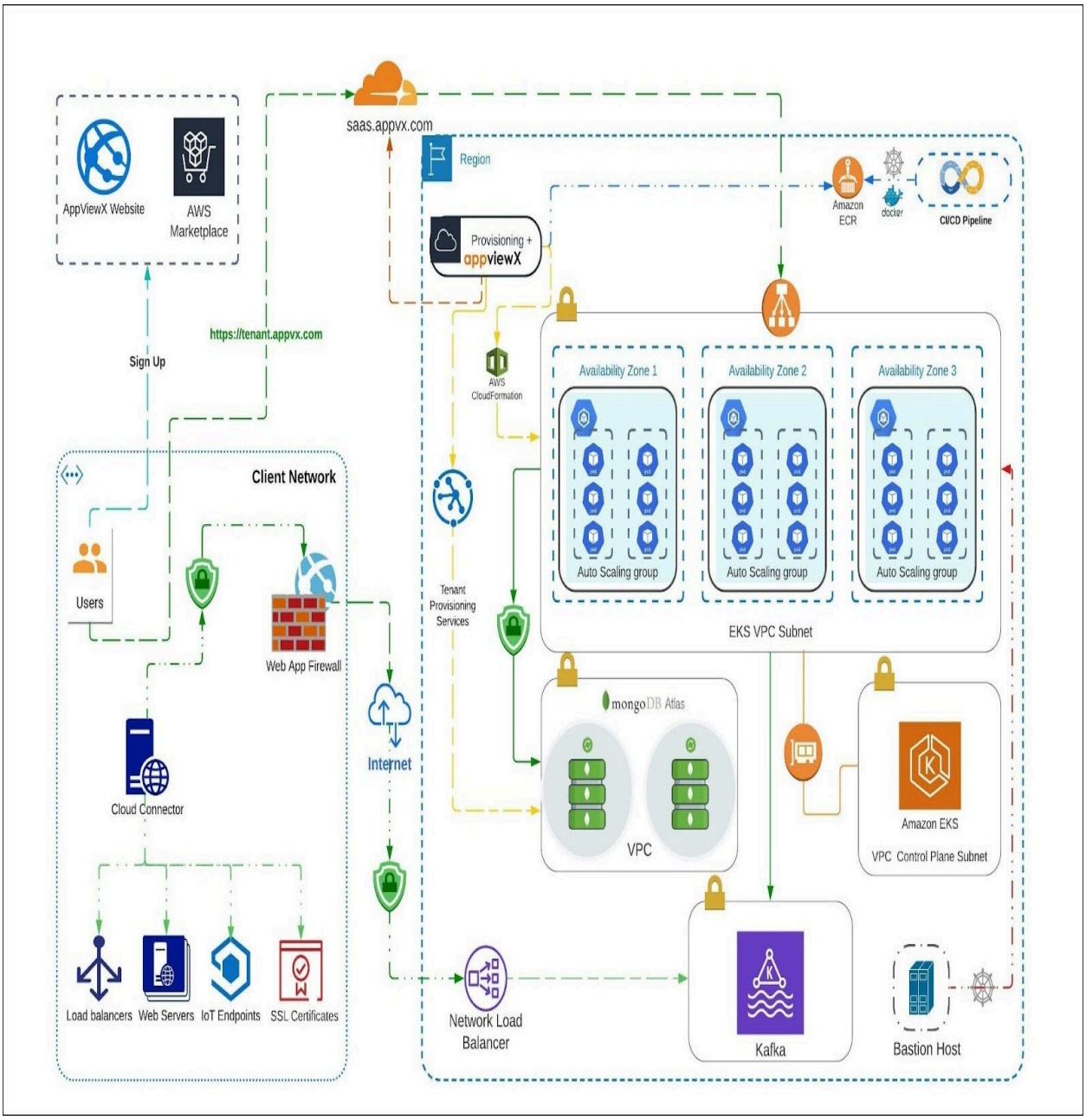
Architecture Components Overview

The AppViewX SaaS platform is enabled with the help of Provisioning, Compute, Database, Monitoring clusters and an AppViewX Cloud Connector.

The key tenets of the platform include:

1. Provisioning Cluster
2. Compute Cluster
3. Database Cluster
4. Monitoring Cluster
5. AppViewX Cloud Connector

Copyright © 2025 AppViewX Inc. All Rights Reserved.



Deployment Architecture

- Provisioning Cluster (SaaS Management Portal)
- Compute Cluster
- Database Cluster

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- Monitoring Cluster
- The AppViewX Cloud Connector

Provisioning Cluster (SaaS Management Portal)

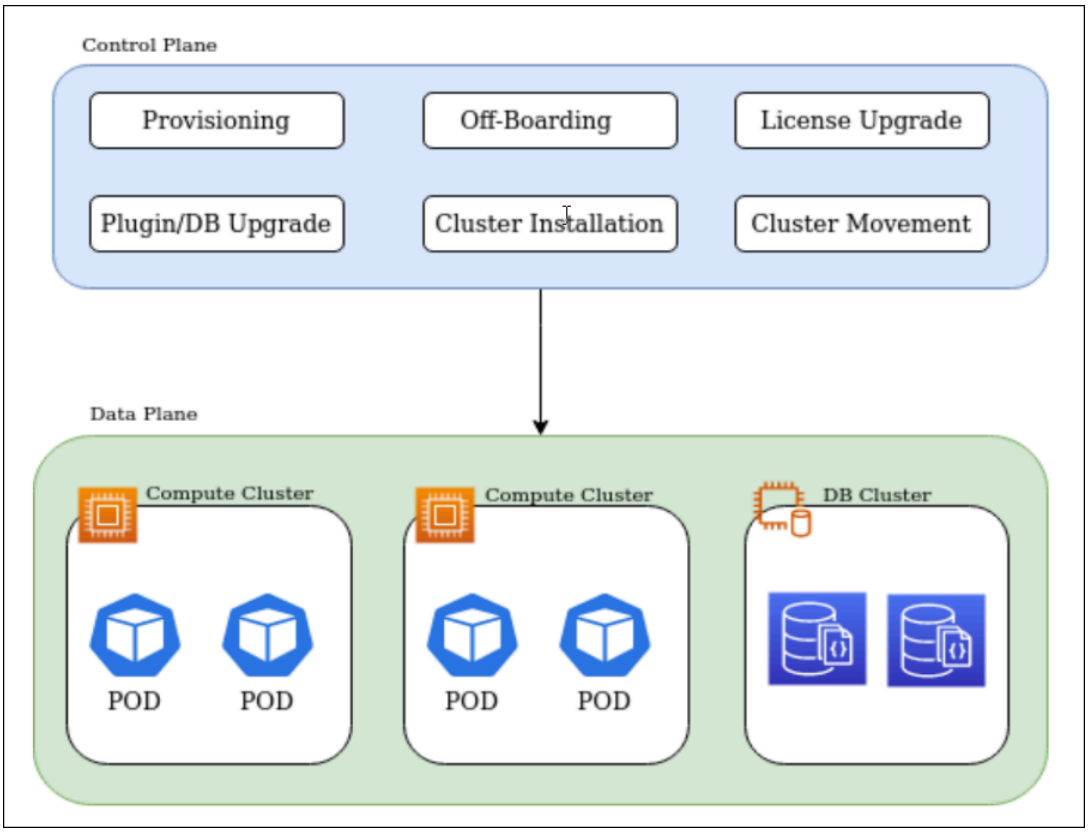
Provisioning Cluster (aka SaaS Management portal) is a cluster used to orchestrate the compute and database instances that powers the AppViewX SaaS. The cluster provides key capabilities like management and visibility into tenants and clusters which includes licensing, upgrading the SaaS tenants and so on, from a Single pane of glass. The granular features and capabilities of the portal are explained below.

- The Provisioning cluster is an internal AppViewX platform for SaaS lifecycle management which can be deployed cross zone or cross region for high availability.

> **Note:** The cluster can be deployed in a dedicated AWS account (not necessary to be deployed in the AWS account where the actual compute cluster is deployed).

- Key features of the provisioning cluster include:
  1. Tenant Life Cycle Management - Onboarding, Offboarding, Licence upgrade.
  2. Cluster Management - Create , Delete , Modify, Upgrade Compute clusters
  3. AppViewX SaaS Life Cycle Management - Install AppViewX on Compute, Upgrade AppViewX (via Canary upgrades), Install Infrastructure components (Istio , ELK etc.,)
- The Provisioning Cluster uses cloudformation templates for creating the Compute cluster and AppViewX automation workflows for mapping the DB clusters with the compute and the other process tied to AppViewX SaaS life cycle management.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



AppViewX Provisioning Cluster Architecture

Compute Cluster

AppViewX compute cluster is a managed-compute infrastructure that runs the AppViewX business logic. The compute cluster is powered by Amazon Elastic Kubernetes Service (Amazon EKS) which is a managed AWS Kubernetes service that scales, manages, and deploys containerized applications.

The compute cluster is deployed via the Provisioning cluster using the cloud formation template. Compute cluster encompasses the below.

- EKS
- AppViewX workloads
- Infrastructure components
- Bastion Host

Copyright © 2025 AppViewX Inc. All Rights Reserved.

• EKS Cluster
• AppViewX Workloads & Infrastructure Components
• Bastion Host

EKS Cluster

EKS clusters are composed of the following main components—a control plane and worker nodes. Each cluster runs in its own, fully managed Virtual Private Cloud (VPC).

The control plane is composed of three master nodes, each running in a different AZ to ensure AWS high availability. Incoming traffic directed to the Kubernetes API passes through the AWS network load balancer (NLB).

Worker nodes run on Amazon EC2 instances located in a VPC. EKS provides managed node groups with automated lifecycle management. This lets users automatically create, update, or shut down nodes with one operation. EKS uses Amazon's latest Linux AMIs optimised for use with EKS. When nodes are terminated, EKS gracefully drains them to make sure there is no interruption of service.

High Availability

Amazon EKS runs and scales the Kubernetes control plane across multiple AWS Availability Zones to ensure high availability. Amazon EKS automatically scales control plane instances based on load, detects and replaces unhealthy control plane instances, and automatically patches the control plane.

The EKS cluster consists of EC2 instances deployed in multiple availability zones within the region. Each instance has replicas of the services and nodes which exist across all the EC2 instances.

Each zone or instance has an active pod listening to other instances. In case of a failure of any instance, the active pod ensures seamless functioning of the application by activating the nodes from any other working cluster.



Note: EKS clusters are deployed within specific regions and each region has multiple availability zones. Example - Region : us-east-1 and the respective zones : us-east-1a, us-east-1b, us-east-1c.

AppViewX Workloads & Infrastructure Components

AppViewX workloads are containerized workloads running as microservices and these containers are orchestrated using Amazon Elastic Kubernetes Service (Amazon EKS).

Copyright © 2025 AppViewX Inc. All Rights Reserved.

The workloads are a mix of AppViewX Business logics that enable communication from User Interface to AppViewX core services and AppViewX SaaS services which is used for enabling the SaaS communication from the AppViewX's SaaS compute to the customer network.

The infrastructure components encompasses third party components that are used for the purpose of service mesh, log aggregation and monitoring the utilisation of the application workloads and so on.

All these workloads, infrastructure components are deployed from the provisioning cluster.

Bastion Host

A bastion host is another EC2 instance based on Linux OS which is created on the same VPC of the wokernodes and this is used for cluster admin operations and troubleshooting the application if required.

The bastion host is accessed via SSH keys generated during the EKS cluster creation and each and every cluster have their own SSH key and the key is downloaded only from the AppViewX SaaS provisioning cluster.

Database Cluster

AppViewX Database cluster is a managed database infrastructure of AppViewX SaaS which holds the customer data. The database cluster is powered by MongoDB Atlas which brings together capabilities that are critical to a modern, cloud-native, microservice-aligned database architecture, including scalability, availability, and uptime.

The AppViewX Database Cluster is enabled via MongoDB Atlas which is a global cloud document database service. The Atlas service MongoDB ensures availability, scalability, and security compliance. The granular features of this cluster are:

- A single database with multiple schemas.
- Individual schemas are generated for each Licensed tenant.
- Snapshots are created for the licensed tenant in the DB cluster and each of the snapshots contains mandatory schemas such as :
  - appSession
  - appviewx
  - appviewxCA
- These schemas are created before the tenants are onboarded.
- Apart from the three mandatory schemas, Snapshot Ids are created for the following schemas:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

  - connectedPlatform
  - imageDetails
  - templateDB
  - workFlowDB
  - workFlowDBEn..
- The tenant data is secured and isolated due to this segregation of schemas.
- It also ensures the singularity of data for each licensed tenant.

The AppViewX Database Cluster is made highly available by enabling the cluster deployment on multiple zones or even more resilient by enabling the cluster deployment on multiple regions. Each of these Clusters have a unique URL and credential associated with it.

Monitoring Cluster

Monitoring Cluster is a managed infrastructure of AppViewX SaaS which caters to monitoring, and understanding the performance of the application and machine critical services which is a condensed form of metadata, metrics, and events about the application and its underlying services. This is enabled with a monitoring stack comprising Prometheus, Grafana, Loki, Promtail, and AlertManager.

The monitoring cluster has a Status dashboard and is deployed in a separate cluster which is again a subset of AppViewX powered with AppViewX monitoring capability enabled via Prometheus, Grafana, Loki, Promtail, and AlertManager and it is deployed on AWS (like an onprem AppViewX deployment) with its own database, compute etc which can be deployed cross zone or cross region for high availability.

The status dashboard application will have three microservices

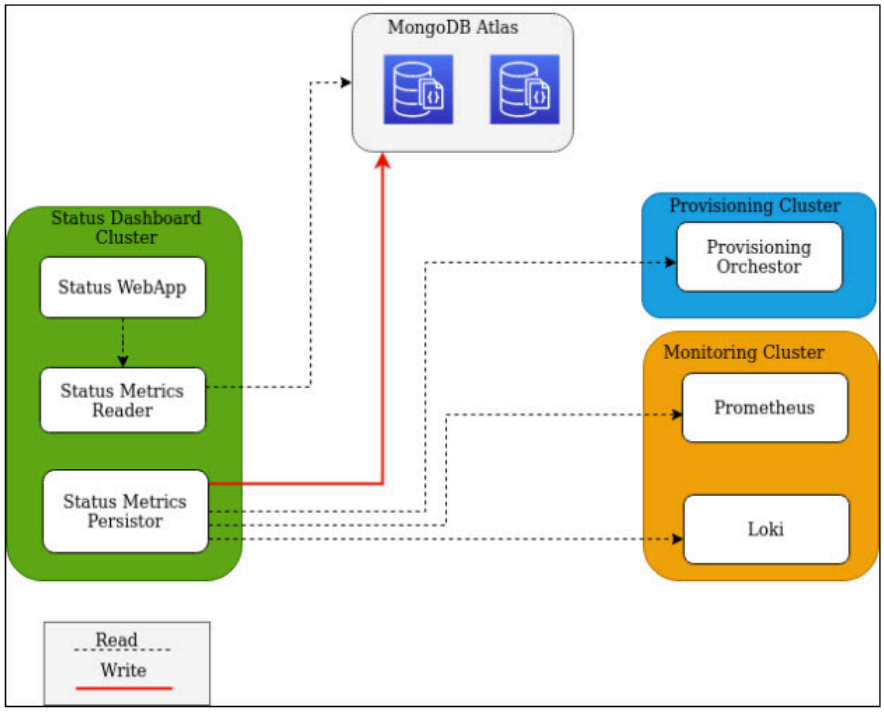
- Status dashboard Webapp
- Status Metrics Reader
- Status Metrics Persistor

The Webapp talks to Status metrics Reader and displays AppViewX services uptime details.

The Status Metrics Reader reads data from a dedicated instance residing in MongoDB Atlas.

The Status Metrics Persitor aggregates data from Monitoring and Provisioning clusters and save them in the dedicated instance which resides in MongoDB Atlas.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



Monitoring Cluster Architecture

The AppViewX Cloud Connector

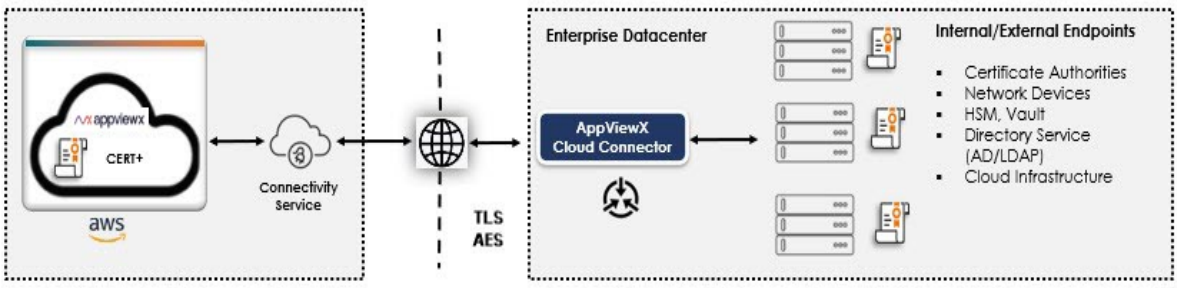
AppViewX Cloud Connector is a lightweight plug-in that establishes connectivity between AppViewX Cloud and the Enterprise Network. The cloud connector serves as a secure channel for communication between AppViewX SaaS and your enterprise network without requiring any complex network or infrastructure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

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

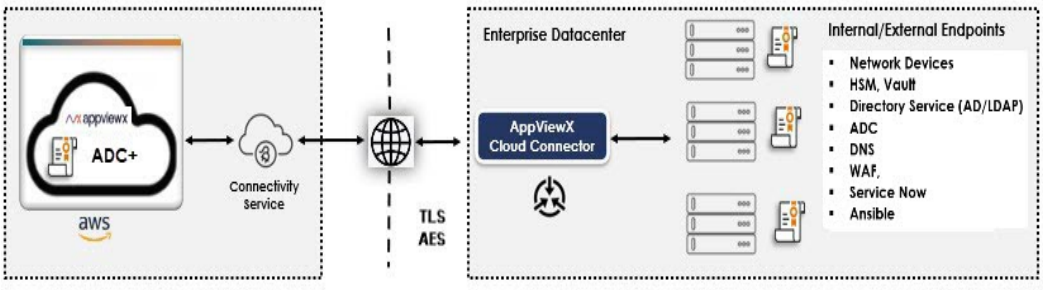
- **CERT+**
  - Discovering certificates from an endpoint within the enterprise network via Smart Network Scan and Managed Device Scan.
  - Discovering certificates from Certificate Authorities (CAs) that are internal to the enterprise. For example : EJBCA.
  - Discovering certificates from public Certificate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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

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



- **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



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)

Copyright © 2025 AppViewX Inc. All Rights Reserved.

AppViewX SaaS Onboarding and Getting Started 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.

- Key Highlights of AppViewX Software as a Service
- Introduction to the AppViewX Cloud Connector
- Prerequisites for Setting up AppViewX Cloud Connector
- Getting Started with the AppViewX Free Trial
- Signing Up for the Free Trial via the AppViewX Website
- Signing Up for the Free Trial via the AWS Marketplace

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 products:

- CERT+, 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+ User Guide.
- ADC+, 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.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

For details, refer the ADC+ 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.
- SSH+, which lets you:
  - Discover and display SSH certificates alongside SSH keys, offering a more comprehensive overview of your security credentials.
  - Download keys for key-based access control, ensuring streamlined access management.
  - Specify access duration in either hours or days when requesting access to an infrastructure group, providing enhanced access management control.
  - Use a dynamic access flow that adapts to either key or certificate-based access, depending on the user's selected 'Access Mode' during host addition.
  - Rotate host certificates effortlessly, directly from the host inventory, promoting secure host certificate management.
  - Revoke SSH certificates directly, thus enhancing security control.
  - Choose between 'Key' and 'Certificate' access modes during host addition, with the 'Certificate' option being pre-selected by default.
  - Rotate and delete keys from hosts with multiple keys through the user and host key age report.

  For details, refer the SSH+ User Guide.
- SIGN+, which lets you:
  - Simplify Code Signing Certificate enrollment and Certificate Lifecycle Management (CLM) operations.
  - Customize signing policies according to your requirements
  - Integrate with AppViewX's customized Cryptographic Service Provider (CSP) and PKCS#11 for enhanced security.
  - Manage your code signing inventory with a full suite of tools and features.
  - Sign your code effortlessly using a variety of tools including SignTool, JSign, JarSigner, APKSigner, Mage, and Nuget.
  - Ensure compatibility with third-party Timestamp Authorities (TSA) for a wider range of options.

  For details, refer the SIGN+ User Guide.
- KUBE+, which lets you:
  - Simplify Certificate Lifecycle Management for Kubernetes workloads.
  - Get real-time visibility, central audit, and governance over K8’s Certs.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- Achieve end-to-end automated certificate enrollment process.
- Have secure and compliant PKI across K8s workloads (secrets, pods, and service mesh).

For details, refer the KUBE+ User Guide.

Introduction to 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.

Key features of the AppViewX Cloud Connector include:

- Data center-based routing
- High availability
- Integrated gatway functionality
- Custom certificates for core communication
- Communication authentication and encryption

Refer to the AppViewX Cloud Connector User Guide, to read more on the features of the AppViewX Cloud Connector and the services it supports.

Prerequisites for Setting up AppViewX Cloud Connector

The AppViewX Cloud Connector can be set up in two ways: using the **virtual image** and via the **native OS**.

For the complete list of system requirements that are minimum prerequisites for setting up and operating the AppViewX Cloud Connector, click here.

Important: For installation via the virtual image, only the **hardware** and **server and network** prerequisites have to be ensured. The operating system and Docker prerequisites are packaged as part of the OVA.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Getting Started with the AppViewX Free Trial

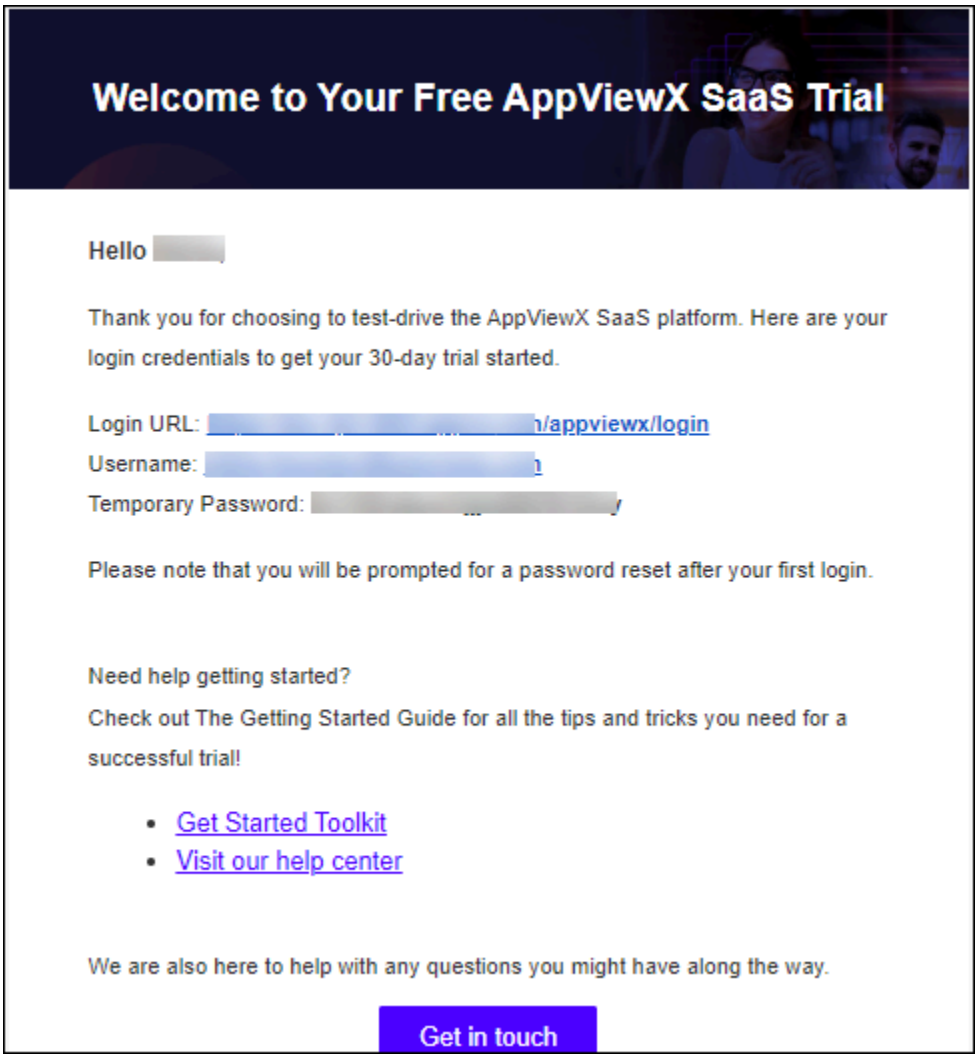
For users evaluating the AppViewX SaaS solution, which enables turnkey Certificate Lifecycle Management, ADC management and automation, and PKI, AppViewX enables two channels to onboard you for a free trial of the product:

- via the AppViewX website
- via the AWS Marketplace

Signing Up for the Free Trial via the AppViewX Website

1. Go to the SaaS free trial page.
2. On the landing page, enter your details and click **Get Stared**.
   A confirmation message that you will receive an email from AppViewX on your registered email ID shortly is displayed. You will receive a verification email shortly.The welcome email you receive will include your login URL and temporary password.
3. In the verification email, click **Verify your Email Address**.
   A welcome email, with your login URL and temporary credentials, is sent to your registered email ID.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



Troubleshooting: If you have not received the welcome email, please get in touch with **sales@appviewx.com**.

4. Go to the login URL.
5. On the login page, enter the credentials you received in the welcome email.
   The OTP verification screen is displayed and an OTP is sent to your registered email ID.
6. Enter the OTP and click **Continue**.
   The **Change password** screen is displayed.
7. Enter and reenter your new password in the **Enter New Password** and **Confirm New Password** fields respectively, and click **Continue**.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Note: Your 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 notifying successful password change is displayed.

You will be redirected to the login page.

8. Login to the application with your new credentials.
   An OTP is sent to your registered email ID.
9. On the **OTP Verification** screen, enter the OTP received and click **Continue**.
10. On the **Terms of Service** screen, select the **I accept the terms and conditions** checkbox and click **Continue**.
    The AppViewX Platform landing page is displayed.
11. To try a product, click **Try Now** to start your 30-day trial of the product.
    You will be redirected to the **GET STARTED** page of the selected product.

Note: The 30-day trial period starts from the day you receive the welcome email. The trial period can be extended by 60 more days (which makes the trial duration 90 days). For more details on how you can extend your trial, please reach out to AppViewX Support.

What to do next:

- For instructions on installing the AppViewX Cloud Connector, refer to the corresponding documentation here.
- To simplify your interaction with the product features, refer to AppViewX's exhaustive documentation in the form of the following guides:
  - AppViewX Cloud Connector User Guide
  - CERT+ User Guide
  - CERT+ Admin Guide
  - Platform User Guide
  - ADC+ User Guide

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- ADC+ Admin Guide
- PKI+ User Guide
- SIGN+ User Guide
- SIGN+ Admin Guide
- KUBE+ User Guide

You can access the complete AppViewX documentation here.

Signing Up for the Free Trial via the AWS Marketplace

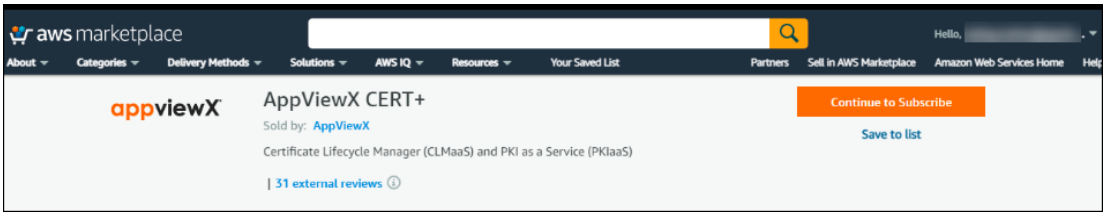
Note: Currently, you can sign up only for the CERT+ SaaS trial via the AWS Marketplace.

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:

Step 1: Accessing the AWS Marketplace Sign Up Page

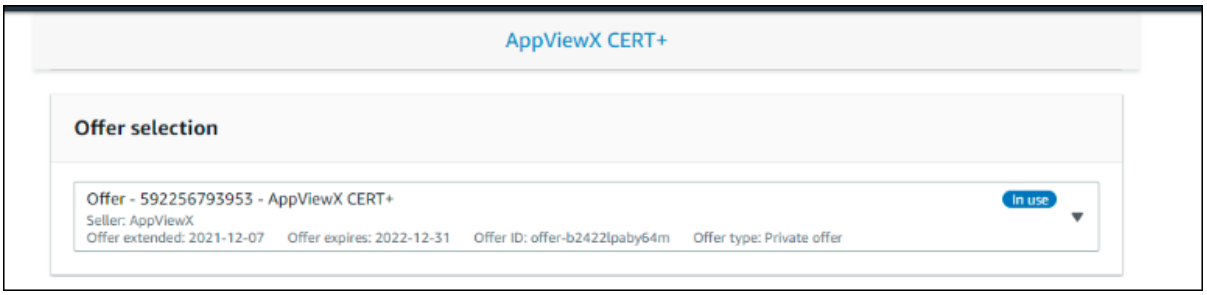
1. Navigate to the AWS Marketplace page.

   The **AppViewX CERT+** page is displayed.



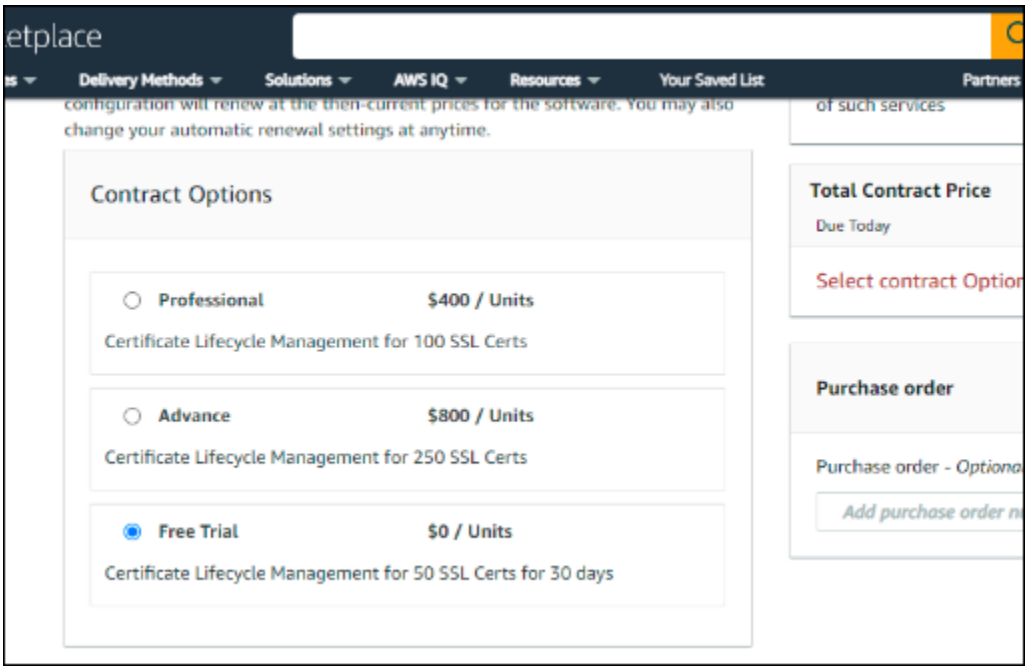
2. Sign into your account or create your account if you are new to AWS Marketplace.
3. Click **Continue to Subscribe.**

   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.



Copyright © 2025 AppViewX Inc. All Rights Reserved.

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.

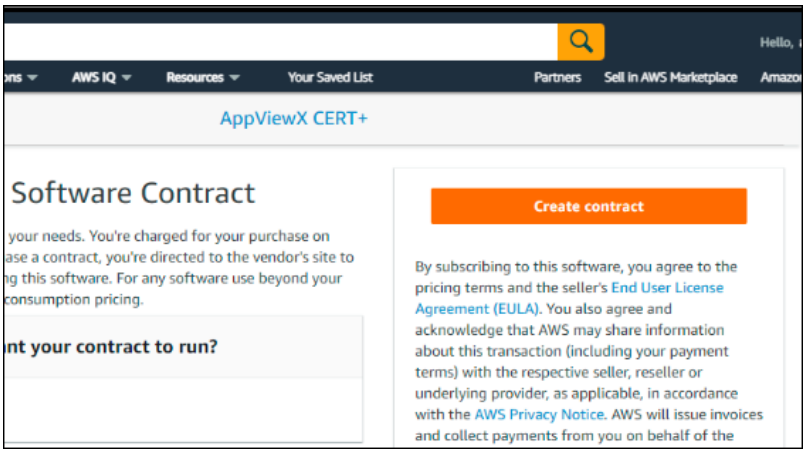


4. Set **Auto Renew** to **No.**



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

Copyright © 2025 AppViewX Inc. All Rights Reserved.



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

   A confirmation dialog box is displayed.

6. In the confirmation dialog box, click **Setup your account** to complete the signup.

   You will be redirected to the AppViewX SaaS registration page.

Step 2: Filling the Sign Up Form

1. To get started with your free trail, enter the following details:

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Field	Description
Select Country*	Select the country from the dropdown list.

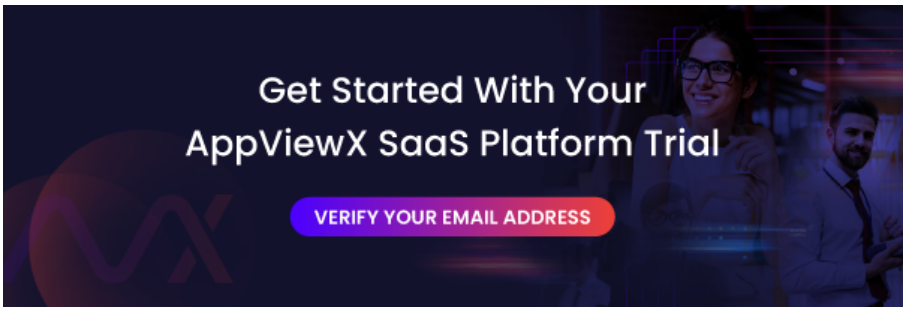
Note:

- 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.

Step 3: 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.

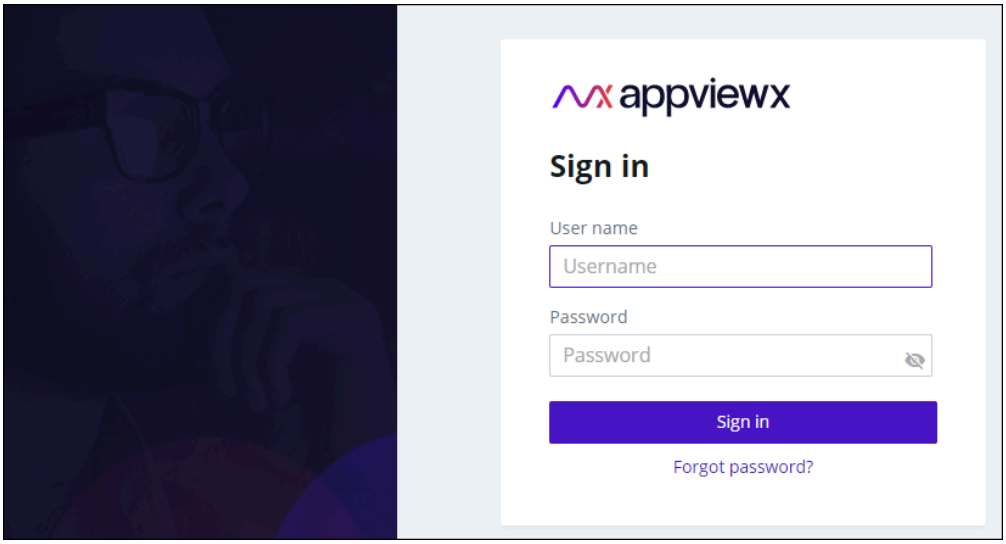
Copyright © 2025 AppViewX Inc. All Rights Reserved.

Step 4: Logging in to your SaaS Account

Based on the details entered, you will receive a welcome email on your registered email ID. The email includes your login URL and temporary credentials.

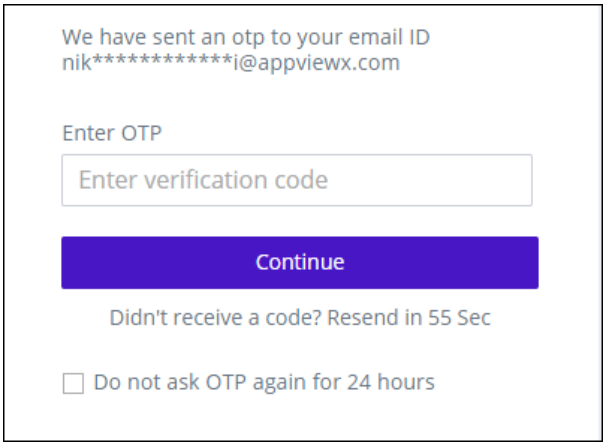
1. Navigate to the login URL.

   The login page is displayed.



2. From the welcome email, login using the credentials provided.

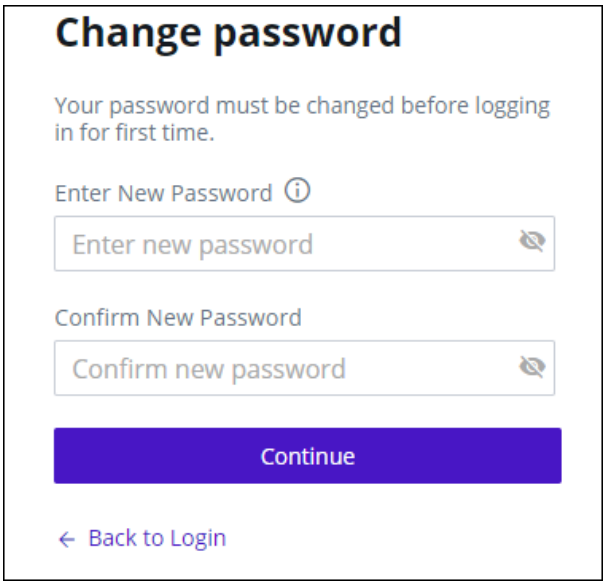
   On successful login, the OTP verification screen is displayed. You will receive the OTP on your registered email address.



3. Enter the OTP received.

   On entering the correct OTP, the **Change password** screen is displayed.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



4. Enter and reenter your new password in the **Enter New Password** and **Confirm New Password** fields respectively.

> **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

5. Click **Continue**.

   A message notifying successful password change is displayed.

   You will be redirected to the login page again.

6. Sign in with your new credentials.
7. n the **OTP Verification** screen, enter the OTP received on your registered email and click **Continue**.
8. On the **Terms of Service** screen, select the **I accept the terms and conditions** checkbox and click **Continue**.

   The AppViewX Platform landing page is displayed.

9. To try a product, click **Try Now** to start your 30-day trial of the product.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

You will be redirected to the **GET STARTED** page of the selected product.

> **Note:** The 30-day trial period starts from the day you receive the welcome email. The trial period can be extended by 60 more days (which makes the trial duration 90 days). For more details on how you can extend your trial, please reach out to AppViewX Support.

Step 5: Setting up the AppViewX Cloud Connector

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

- Using the virtual machine

  For instructions on setting up the AppViewX Cloud Consumer using a virtual image, click here.

- Via the native OS

  For instructions on setting up the AppViewX Cloud Consumer using the native OS, click here.

To understand the difference between the two methods, click here.

Step 6: Getting Started with AppViewX SaaS

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

- AppViewX Cloud Connector User Guide
- CERT+ User Guide
- CERT+ Admin Guide
- Platform User Guide
- ADC+ User Guide
- ADC+ Admin Guide
- PKI+ User Guide
- SIGN+ User Guide
- SIGN+ Admin Guide
- KUBE+ User Guide

You can access the complete AppViewX documentation here.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

AppViewX Cloud Connector User Guide

The guide introduces you to the features of the AppViewX Cloud Connector, the component that facilitates a SaaS deployment of AppViewX's flagship products.

The guide includes steps for installing, configuring, managing, and troubleshooting your AppViewX Cloud Connector instance.

- AppViewX Software as a Service
- Features of the AppViewX Cloud Connector
- System Requirements for Setting up the AppViewX Cloud Connector
- Setting Up the AppViewX Cloud Connector
- Managing ADC Devices
- Installing the AppViewX Windows Gateway
- Troubleshooting the AppViewX Cloud Connector
- Managing the AppViewX Cloud Connector
- Frequently Asked Questions
- Appendix A: Network Scan Recommendations
- Appendix B: Automated Installation without Internet
- Appendix C: CIS Benchmarking for AppViewX Cloud Connector
- Appendix D: Verifying and Renewing Kubernetes Certificates in the AppViewX Cloud Connector

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.

AppViewX Cloud Connector is a lightweight plug-in that establishes connectivity between AppViewX Cloud and the Enterprise Network, and is the enabler for AppViewX's SaaS-based deployment. 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.

The AppViewX SaaS platform offers the following three products:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- CERT+, 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+ User Guide.

- ADC+, 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+ 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.

- SSH+, which lets you:
  - Discover and display SSH certificates alongside SSH keys, offering a more comprehensive overview of your security credentials.
  - Download keys for key-based access control, ensuring streamlined access management.
  - Specify access duration in either hours or days when requesting access to an infrastructure group, providing enhanced access management control.
  - Use a dynamic access flow that adapts to either key or certificate-based access, depending on the user's selected 'Access Mode' during host addition.
  - Rotate host certificates effortlessly, directly from the host inventory, promoting secure host certificate management.
  - Revoke SSH certificates directly, thus enhancing security control.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- Choose between 'Key' and 'Certificate' access modes during host addition, with the 'Certificate' option being pre-selected by default.
- Rotate and delete keys from hosts with multiple keys through the user and host key age report.

For details, refer the SSH+ User Guide.

- SIGN+, which lets you:
  - Simplify Code Signing Certificate enrollment and Certificate Lifecycle Management (CLM) operations.
  - Customize signing policies according to your requirements
  - Integrate with AppViewX's customized Cryptographic Service Provider (CSP) and PKCS#11 for enhanced security.
  - Manage your code signing inventory with a full suite of tools and features.
  - Sign your code effortlessly using a variety of tools including SignTool, JSign, JarSigner, APKSigner, Mage, and Nuget.
  - Ensure compatibility with third-party Timestamp Authorities (TSA) for a wider range of options.

  For details, refer the SIGN+ User Guide.
- KUBE+, which lets you:
  - Simplify Certificate Lifecycle Management for Kubernetes workloads.
  - Get real-time visibility, central audit, and governance over K8’s Certs.
  - Achieve end-to-end automated certificate enrollment process.
  - Have secure and compliant PKI across K8s workloads (secrets, pods, and service mesh).

  For details, refer the KUBE+ User Guide.
- The AppViewX Cloud Connector

The AppViewX Cloud Connector

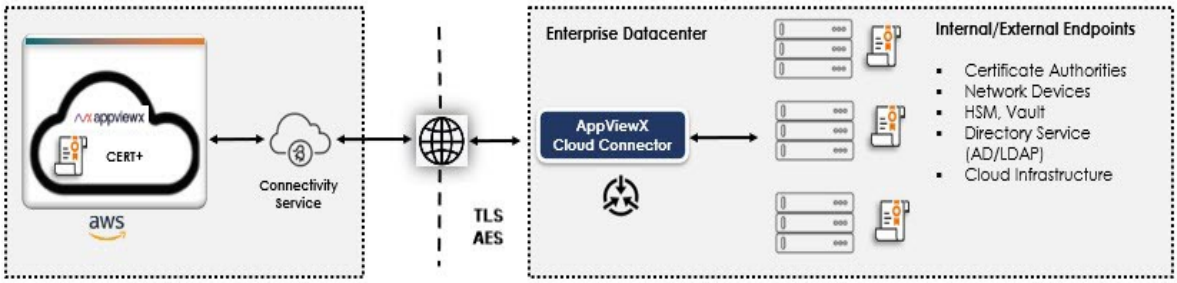
AppViewX Cloud Connector is a lightweight plug-in that establishes connectivity between AppViewX Cloud and the Enterprise Network. The cloud connector serves as a secure channel for communication between AppViewX SaaS and your enterprise network without requiring any complex network or infrastructure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

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

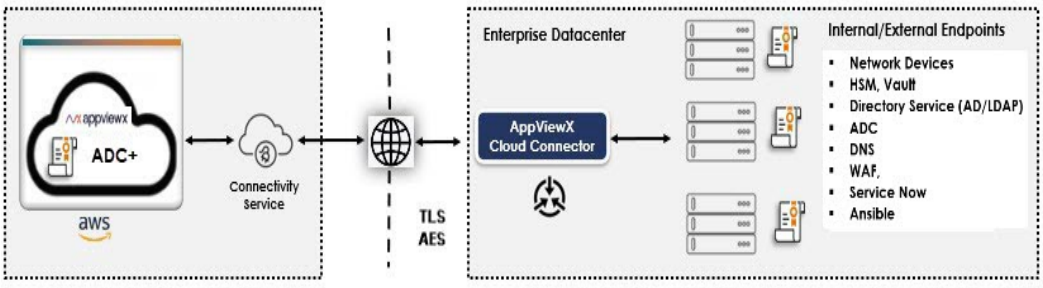
Copyright © 2025 AppViewX Inc. All Rights Reserved.

- **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**.



- **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



Key features of the AppViewX Cloud Connector:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- 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

- AppViewX Cloud Connector DataCenter Significance
- Cloud Connector High Availability
- Integrated Gateway
- Custom Certificates for Core Communication
- Communication Authentication and Encryption
- Auto Enrollment with the AppViewX Cloud Connector
- Enabling Proxy for End Point Communication
- mTLS Enablement

AppViewX Cloud Connector DataCenter Significance

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

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

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

You can also choose to have a dedicated cloud connector instance for specific endpoints. To do this, at the time of onboarding the endpoint (Managed Devices, Network Scan, Certificate Authority, and so

Copyright © 2025 AppViewX Inc. All Rights Reserved.

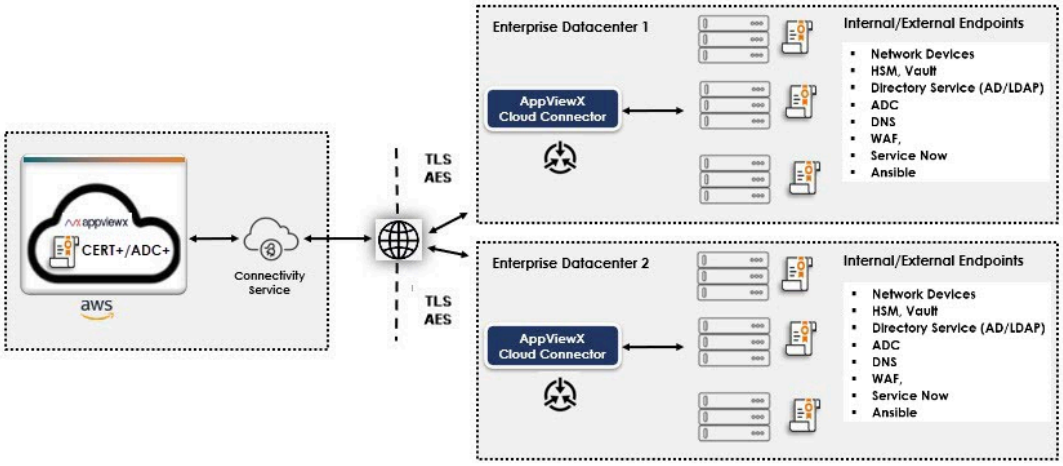
on), select the AppViewX Cloud Connector DataCenter Name from the **DataCenter** field in each of the endpoint onboarding pages.

• Data Center-based Routing

Data Center-based Routing

The AppViewX Cloud Connector instances that need to connect to the network endpoints are deployed inside a specific DataCenter in an enterprise's premises. Based on the DataCenter in which the cloud connector is added, the calls to manage the end points are routed to the specific cloud connector inside a DataCenter.

Figure 2. 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.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

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

Integrated Gateway

To eliminate the need for a separate installation, AppViewX has introduced the integrated gateway functionality (tech preview). The integrated gateway is a Windows Gateway Agent that is packaged along with the AppViewX Cloud Connector.

Important:

- Ensure that you upgrade your AppViewX Cloud Connector to be able to apply this functionality.
- Integrated gateway is supported with HTTP and HTTPs modes using winRM.

The integrated gateway is currently implemented for the following vendors:

- **Devices**
  - Microsoft IIS
  - Microsoft PC
  - Microsoft Server
  - Microsoft SQL
  - Apache (Windows)
  - Tomcat (Windows)

Copyright © 2025 AppViewX Inc. All Rights Reserved.

For instructions on using the internal gateway for the supported device vendors, click the vendor name from the above list.

- **CAs**
  - Microsoft Enterprise CA
  - Microsoft Standalone CA



Note: The following features are now available for Microsoft devices added through the Integrated Gateway.

- MS devices (Integrated gateway) can be added and exported using Export & Import option (excel sheet upload). The following column must be added to make it work with integrated gateway.
  - **Windows Gateway mode** (the column value should be *cloudConnector*).
- All Microsoft devices (added via the integrated gateway) can be used for endpoint CSR generation.
- Pre and Post push script executions can be performed on devices configured using Integrated gateway.

For instructions on using the internal gateway for the supported Certificate Authorities, click the CA name from the above list.

For prerequsistes of enabling the Integrated Gateway refer to the section Appendix A: Prerequisites for enabling the Integrated Windows Gateway Mode.

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.



Note: This is explained in detail as part of the insrtuctions for setting up the AppViewX Cloud Connector.

Communication Authentication and Encryption

Copyright © 2025 AppViewX Inc. All Rights Reserved.

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

Auto Enrollment with the AppViewX Cloud Connector

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

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

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

Note: For complete documentation of the auto enrollment protocols supported in AppViewX, click here.

Enabling Proxy for End Point Communication

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

Note: This is an optional enablement and should be performed only is a proxy setup is required. For instructions on configuring the network proxy settings, click here.

mTLS Enablement

Overview

The AppViewX SaaS platform utilizes a lightweight agent called the Cloud Connector to facilitate secure communication between the customer's on-premises environment and the SaaS infrastructure. This communication was previously secured using standard TLS. For customers requiring mutual TLS (mTLS),

Copyright © 2025 AppViewX Inc. All Rights Reserved.

the configuration used to be performed manually by the AppViewX SRE team. To streamline this process, mTLS enablement has now been automated, thereby removing the dependency on manual intervention.

Prerequisites

Before enabling mTLS for existing cloud connectors, ensure the following prerequisites are met:

- Ensure Cloud Connectors are upgraded to the latest version to enable seamless and automated mutual TLS (mTLS) configuration.
- The provisioning cluster must have Cloudflare integration configured, including API credentials in the Integration Settings, to facilitate automated root certificate upload and DNS record mapping via the Cloudflare API.
- Each cluster is required to have an internal tenant that is up and running, as the Cloudflare WAF rule configuration is initiated through this tenant.
- SMTP settings must be configured for the internal tenant, as mTLS client certificate expiration motifications will be sent via the internal tenant.
- Customers who are currently using mTLS and who wish to switch to the automated mTLS enablement process are required to upgrade to the latest version. After the upgrade, the AppViewX SRE team will assist in enabling the mTLS using the latest root and client certificates.

General Constraints

- mTLS is not supported for Cloud Connectors deployed via HELM.
- mTLS will be enabled only for active and running Cloud Connectors. Connectors that are in a stopped or paused state will continue operating with default TLS communication.
- Only self-signed certificates are supported for mTLS. Customer-provided or third-party CA certificates are not currently supported.
- Automated renewal of the root certificate is not supported. In the event of root certificate expiration, the SRE team must be contacted.
- By default, the root certificate is valid for 10 years, while the mTLS certificate is valid for 1 year
- Expiry notifications for the mTLS client certificate will be trigerred 90 day prior, and the renewal process will be handled 30 days before expiry.

mTLS Enablement for Existing Cloud Connectors

In this method the mTLS is enabled for existing Cloud Connectors within the customer's environment. The AppViewX SRE team internally utilizes an automated workflow to perform mTLS enablement for a tenant’s Cloud Connectors.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

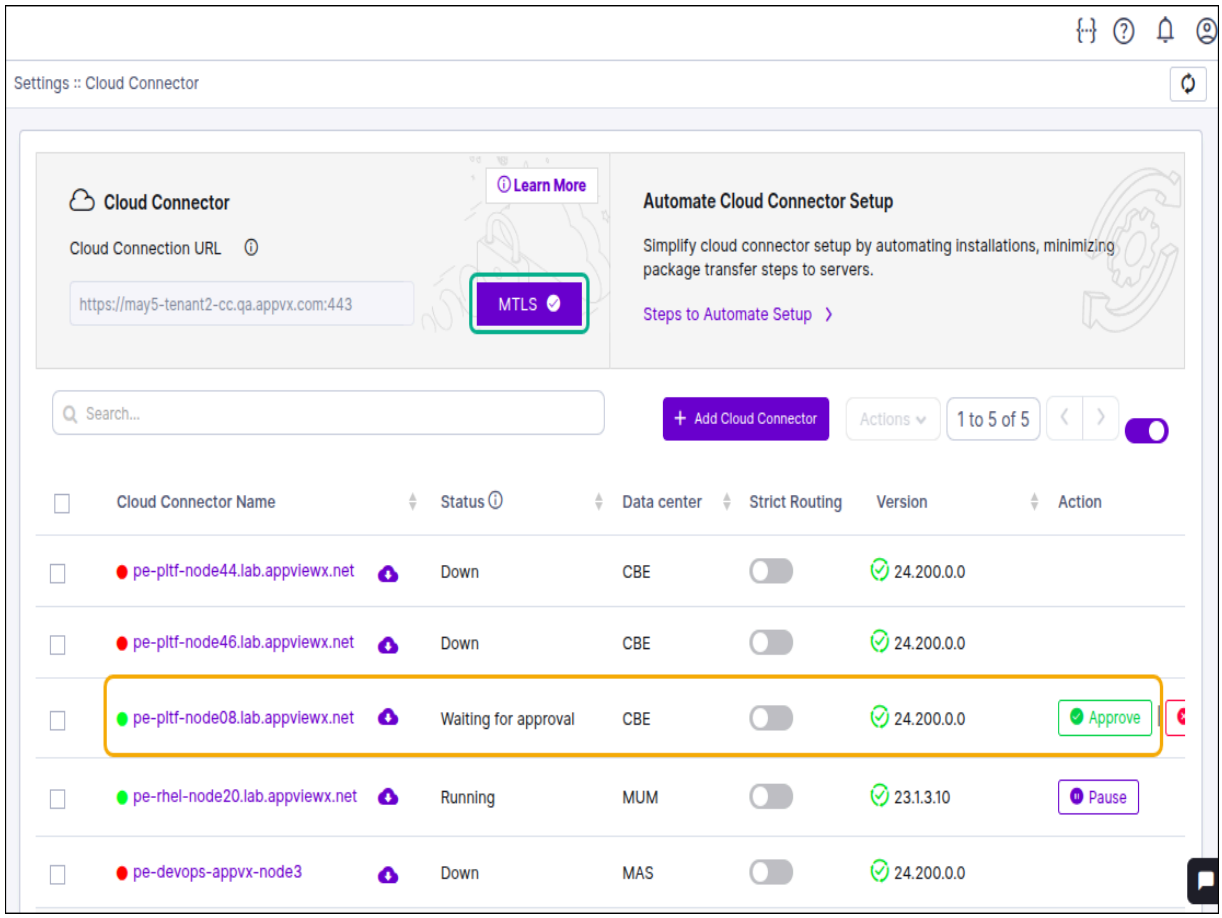
mTLS Enablement for New Cloud Connector

This method details the mTLS enablement process for newly deployed Cloud Connectors. As noted in the prerequisites, Cloud Connectors can be installed via multiple methods—Manual, Automated, Native OS-based, or HELM-based. mTLS enablement is supported across all installation types except HELM-based deployments.

Once the Cloud Connector is installed in the customer's environment, the customer must:

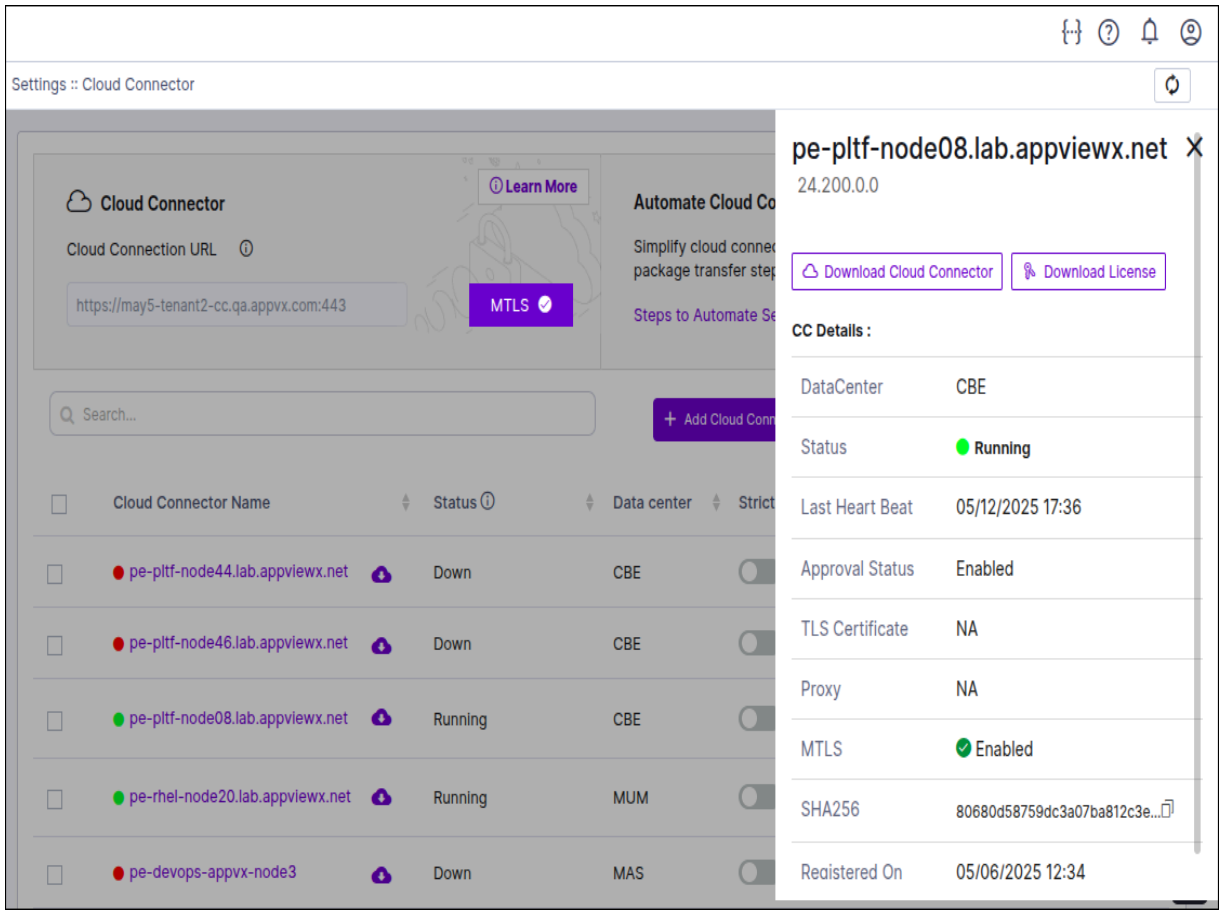
1. Log in to the AppViewX UI.
2. Go to **Menu** > **Platform** > **CONNECTVITY** > **Cloud Connectors**. (*The Settings :: Cloud Connector page is displayed.)*
3. For the connector status that is "Waiting for Approval", click **Approve** to initiate communication.

> **Note:** If any one cloud connector is enabled with mTLS communication, then at the inventory page level, the mTLS enabled information icon will be displayed.



Copyright © 2025 AppViewX Inc. All Rights Reserved.

4. To view the status of the cloud connector, click on the Cloud Connector Name. The mTLS status will be shown individually for each cloud connector.



If mTLS has been selected, the enablement process is automatically triggered during approval. A client certificate is generated and pushed to the Cloud Connector. Upon successful certificate deployment, the system attempts to enable the corresponding WAF rule. If WAF rule activation fails, up to three retry attempts are made.

After the WAF rule is enabled, the system waits two minutes for a heartbeat signal from the connector. If no signal is received, mTLS is considered unsuccessful. The WAF rule is then disabled automatically to avoid disruption, and the connector falls back to TLS communication. The failure is logged and notified to the AppViewX SRE team for further investigation.

Auto Renewal process of mTLS Certificate

To maintain uninterrupted and secure mTLS communication, client certificate renewal is fully automated—no manual intervention is required from the customer.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

A scheduled job runs daily to identify certificates nearing expiration (within 30 days). Upon detection, the system automatically triggers the renewal process.

- **Client Certificate Validity Period**: 1 year.
- **Expiry Notification:** Sent 90 days before the certificate expires.
- **Renewal Process:** Automatically managed by the system ahead of expiration.
- **Auto-Renewal:** Performed 30 days prior to expiry without customer involvement.

If any failure occurs during the certificate renewal process, a notification will be sent to the AppViewXSRE team via email.

In case the renewal fails on a given day, the system will automatically retry the process the next day. This retry mechanism continues daily for up to 30 days or until the renewal is successfully completed.

System Requirements for Setting up the AppViewX Cloud Connector

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
- Server and Network Prerequisites
- Docker Prerequisites

Hardware

Each AppViewX Cloud Connector instance requires the following minimum configuration:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- 4vCPU
- 8 GB memory
- 16 GB (Minimum free disk space required for the installation directory)
- x86 64 bit architecture

Note:

- If **/var/lib** is going to be a separate mount, ensure that it has more than 5 GB of free space. In case of restrictions in meeting this requirement, it is recommended to change the data root directory from **/var/lib** to another dedicated directory. For instructions on changing the data root directory, click here.
- For all k3s installation modes, make sure that the **/run** has more than 3 GB of free space.

Operating System

- Ubuntu version 22.04
- RHEL versions 8.6, 8.7, 8.8, 8.10, 9.2, 9.3 and 9.4 (Refer the prerequisites for RHEL below)
- Amazon Linux 2
- Rocky Linux 8.10 (Green Obsidian)

Note:

- If **/var/lib** is going to be a separate mount, ensure that it has minimum 5 GB of free space. In case of restrictions in meeting this requirement, it is recommended to change the data root directory from **/var/lib** to another dedicated directory. For instructions on changing the data root directory, click here.
- For a RHEL8+ node, ensure that **/run** has minimum 3 GB of free space.

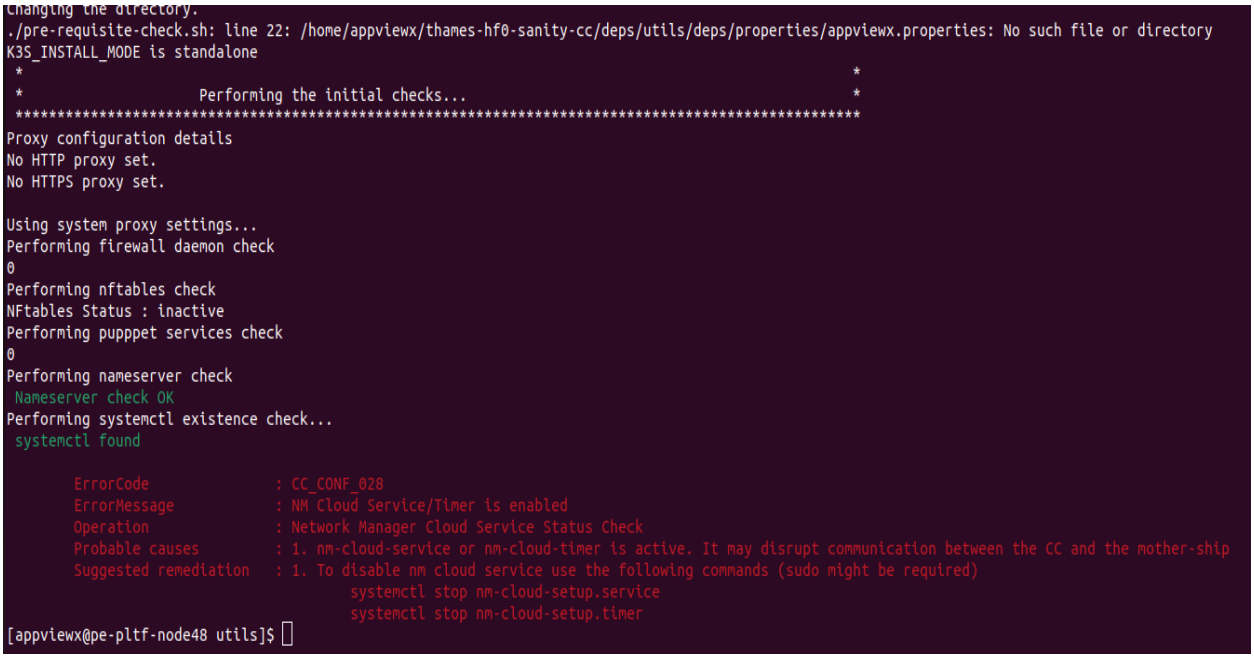
Prerequisites for RHEL OS

Before installing the Cloud Connector on RHEL OS, ensure to disable the following network manager services:

- Network manager cloud service (nm-cloud-setup.service)
- Network manager cloud timer (nm-cloud-setup.timer)

Copyright © 2025 AppViewX Inc. All Rights Reserved.

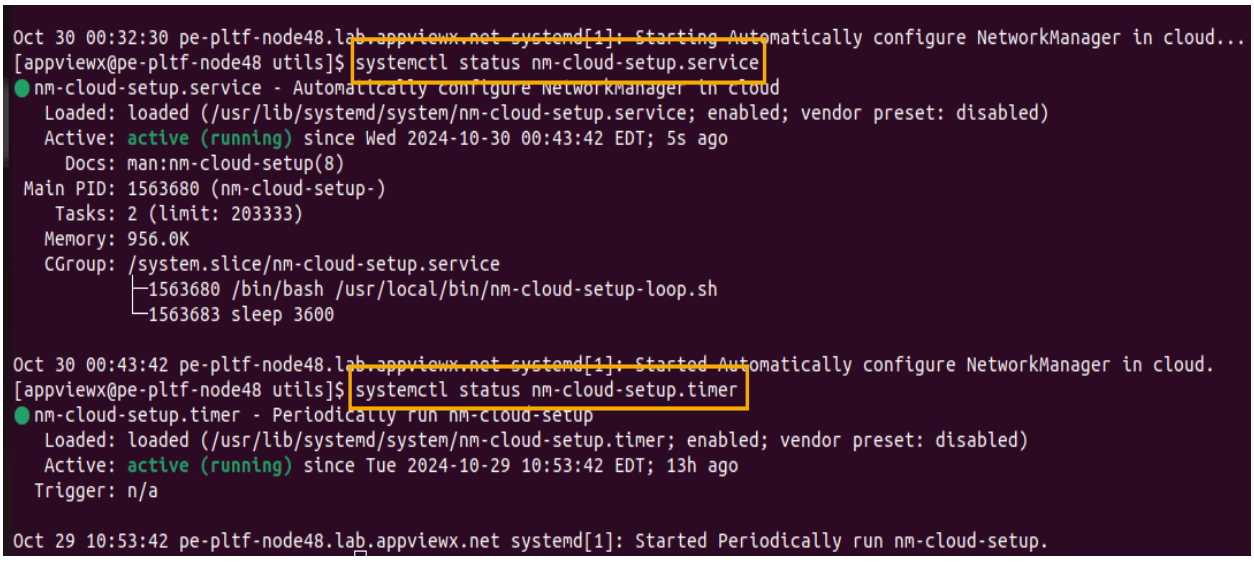
The CC installer verifies this prerequisite by checking the services and displays an error message if they are running, as shown below.



Hence, to perform a smooth installation manually verify if the services are running in the given node by executing the following commands each for both services.

```
systemctl status nm-cloud-setup.service
```

```
systemctl status nm-cloud-setup.timer
```



To disable both the services, execute the following command:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
systemctl stop nm-cloud-setup.service
```

```
systemctl stop nm-cloud-setup.timer
```

Server and Network Prerequisites

General Prerequisites

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

  For the **ntpd** package (for **CentOS**, **RHEL**, and **Amazon Linux 2**), execute the following sequence of commands:

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

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

  - For **CentOS**, **RHEL**, and **Amazon Linux 2**

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

  OR

  - For **Ubuntu**

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

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

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/
```

Also, ensure that the cloud connector URL is not blocked by your puppet scripts, anti-virus software settings, and/or firewall rules.

Tip: : To verify connectivity with the AppViewX SaaS servers, use the **cURL** utility.

1. Install the **cURL** utility.
   - On **Ubuntu**: `sudo apt-get install curl`
   - On **CentOS**, **RHEL**, and **Amazon Linux 2**: `sudo yum install curl`
2. To check connectivity, execute the following command:
   - If proxy is not enabled:

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

   - If proxy is enabled:

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

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

- Disable the firewalld in the tenant's node (**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
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- Disable the firewalld in the tenant's node (**CentOS**, **RHEL**, and **Amazon Linux 2**) 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
  ```

- Disable the **nftables** service.
  - For **Ubuntu**

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

  - Fo**CentOS**, **RHEL**, and **Amazon Linux 2**

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

- If you are utilizing the IP ranges **10.42.0.0/16** and **10.43.0.0/16**, modify them before installing the cloud connector.

  (These are the default CIDR ranges for the AppViewX Cloud Connector and should be modified to prevent IP conflicts).

  To adjust the IP range:

  1. To modify **install.sh**, execute the command: `vi install.sh`.
  2. **For a k3d installation**:
     a. In the **install.sh** file, search for the text **k3d cluster create**.
     b. At the end of this line, paste the following: **--k3s-arg '--cluster-cidr=<preferredcidr>/16' --k3s-arg '--service-cidr=<preferredcidr>/16'**.

        Here, replace **<preferredcidr>** with your preferred CIDR address.

     OR

     For a standard k3s installation:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

a. In the **install.sh** file, search for the text **--disable=metrics-server**.
b. Add a space and paste the following text: **--cluster-cidr=<preferredcidr>/16 --service-cidr=<preferredcidr>/16**.

   Here, replace **<preferredcidr>** with your preferred CIDR address.

3. Save the file.
4. Execute the following command: `./install.sh`

Additional Prerequisites for Installation on RHEL8+

- Ensure that the user has access to perform any action (create, start, stop, and so on) on the **systemd** services through **systemctl**.
- The AppViewX Cloud Connector is installed on top of a Kubernetes engine. To install the underlying Kubernetes engine directly on the host, the user must have **sudo** access with **read/write/execute** permissions for the following directories at the least:
  - **/var/lib**
  - **/etc**
  - **/run**
  - **/usr/local/bin**
  - **/tmp**
- If **nm-cloud-setup** is enabled, disable it and reboot the node.

```
systemctl disable nm-cloud-setup.service nm-cloud-setup.timer

reboot
```

Note: Since RHEL8+ does not include Docker support, these additional prerequisites are necessary for a Docker-less installation of the AppViewX Cloud Connector.

Additional Prerequisites for Installation on Amazon Linux 2

- Deploy an EC2 instance with type C5.Xlarge 4vcpu and 8GB RAM.

Docker Prerequisites

Note: Since RHEL8+ does not include Docker support, Docker prerequisites are not applicable when the AppViewX Cloud Connector is being installed on a RHEL8+ node.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- Docker version 20.10.5, 27.1.1 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.

> **Note:** If **/var/lib** is going to be a separate mount, ensure that it has minimum 5 GB of free space.
>
> In case of restrictions in meeting this requirement, it is recommended to change the data root directory from **/var/lib** to another dedicated directory. For instructions on changing the data root directory, click here.

- Bash shell support in the node for the installation of the AppViewX Cloud Connector Connectivity Service
- Changing the Data Root Directory

Changing the Data Root Directory

> **Note:** The following are one-time setup steps for Docker. Ensure that you execute these steps:
>
> - as the **root** user
> - are performed in each AppViewX Cloud Connector node to change the Docker data root

To change the data root directory (if **/var/lib** has insufficient space to hold the images):

1. View the current root directory using the following command: `docker info -f '{{ .DockerRootDir}}'`
   The name of the current root directory is displayed. For example: **/var/lib/docker**
2. Stop the processes that currently running using the following command: `systemctl stop docker`
3. Verify Docker status using the following command: `systemctl status docker`

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Since Docker processes were stopped in step 2, the status will be displayed as **Active: inactive (dead)**

4. Create the Docker directory, which will be your new data root directory, using the following command:
   `mkdir /new/path/docker-data-root`
   For example, you can choose **/home/appviewx/docker-data-root** as the new path for the data root directory.
5. Copy the content from **/var/lib** to the new data root directory using the following command:`rsync -avxP /var/lib/docker/ /new/path/docker-data-root`
6. To update the path of the Docker daemon file, create or edit the **/etc/docker/daemon.json** configuration file to add the following:

```
{
    "data-root": "/new/path/docker-data-root"
}
```

7. Restart Docker services using the following commands:

```
systemctl daemon-reload
systemctl start docker
```

8. Verify if the new data root directory has been set as the root directory using the following command:
   `docker info -f '{{ .DockerRootDir}}'`
   The output should display the new root directory (for example, **/new/path/docker-data-root**).

Setting Up the AppViewX Cloud Connector

The AppViewX Cloud Connector can be set up by the following methods:

- **Via a Open Virtual Appliance (OVA)**

  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.

  When setting up the cloud connector via a virtual image, you will be required to download only the license file.

- **Via the Native OS**

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- **With Docker runtime**

  Tenant/Administrator/User to provision a Linux machine with docker installed fulfilling 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.

  When setting up the cloud connector via the native OS, you will be required to download a package that contains the cloud connector installer and the license file.

- **RHEL 8+ without Docker runtime**

  Tenant/Administrator/User to provision a RHEL8+ machine fulfilling prerequisites (generic as well as those exclusive for RHEL 8+) across the following categories: hardware, operating system, Docker, and server and network.

  Optional (If required) - To install RHEL OS in the K3s setup with minimal sudo access, refer to the section CC Installation with Minimal Sudo Access in RHEL OS.

- **Helm model in a Kubernetes cluster**

- Setting up the AppViewX Cloud Connector via OVA using the Automated Script
- Setting up the AppViewX Cloud Connector via OVA using AppViewX User Interface (Manual)
- Setting up the AppViewX Cloud Connector by Quick Installation using CLI
- Setting up the AppViewX Cloud Connector using a Packaged Installer
- Setting up the AppViewX Cloud Connector using Helm model in a Kubernetes Cluster

CC Installation with Minimal sudo Access in RHEL

This section enables the cloud connector installation with minimal sudo access in the RHEL environments.

To configure the minimal permissions for the user installing the cloud connector,

1. Copy the below content to a text file/editor and do the following:

```
appviewx_user> ALL=(ALL) NOPASSWD: <cc_tar_file_location>/deps/tools/k3s ctr
images import
<cc_tar_file_location>/deps/tools/mid-server-docker-image/avx-mid-server-base-*
.tar --data-dir=<cc_tar_file_location>/k3s-data-dir/k3s-state/k3s
<appviewx_user> ALL=(ALL) NOPASSWD: /usr/local/bin/k3s-uninstall.sh
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/mkdir -p
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
<cc_tar_file_location>/k3s-data-dir/k3s-state/k3s/agent/images/
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/mkdir -p /etc/rancher/k3s
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/ln -sf k3s /usr/local/bin/kubectl
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/ln -sf k3s /usr/local/bin/crictl
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/ln -sf k3s /usr/local/bin/ctr
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/rm -f
/etc/yum.repos.d/rancher-k3s-common*.repo
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/rm -f /etc/systemd/system/k3s.service
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/rm -f
/etc/systemd/system/k3s.service.env
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/systemctl enable
/etc/systemd/system/k3s.service
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/systemctl daemon-reload
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/systemctl restart k3s
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/systemctl disable k3s
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/touch
/etc/systemd/system/k3s.service.env
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/tee
/etc/yum.repos.d/rancher-k3s-common.repo
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/tee /usr/local/bin/k3s-killall.sh
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/tee /usr/local/bin/k3s-uninstall.sh
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/tee
/etc/systemd/system/k3s.service.env
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/tee -a
/etc/systemd/system/k3s.service.env
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/tee /etc/systemd/system/k3s.service
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/tee /etc/logrotate.d/k3s
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/cp
<cc_tar_file_location>/deps/tools/mid-server-docker-image/rancher-mirrored-core
dns-coredns-*.tar
<cc_tar_file_location>/k3s-data-dir/k3s-state/k3s/agent/images/
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/cp
<cc_tar_file_location>/deps/tools/mid-server-docker-image/rancher-local-path-pr
ovisioner-*.tar <cc_tar_file_location>/k3s-data-dir/k3s-state/k3s/agent/images/
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/cp
<cc_tar_file_location>/deps/tools/mid-server-docker-image/rancher-mirrored-paus
e-*.tar <cc_tar_file_location>/k3s-data-dir/k3s-state/k3s/agent/images/
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/cp
<cc_tar_file_location>/deps/tools/k3s /usr/local/bin/
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/grep '^\\s*SELINUX=enforcing'
/etc/selinux/config
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/chmod 755
/usr/local/bin/k3s-killall.sh
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/chmod 755 /usr/local/bin/k3s
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/chmod 755
/usr/local/bin/k3s-uninstall.sh
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/chmod 0600
/etc/systemd/system/k3s.service.env
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/chmod 0755
/etc/systemd/system/k3s.service
<appviewx_user> ALL=(ALL) NOPASSWD: /usr/bin/sh -c touch
/usr/local/bin/k3s-ro-test && rm -rf /usr/local/bin/k3s-ro-test
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/sha256sum /usr/local/bin/k3s
/etc/systemd/system/k3s.service /etc/systemd/system/k3s.service.env
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/chcon -u system_u -r object_r -t
container_runtime_exec_t /usr/local/bin/k3s
<appviewx_user> ALL=(ALL) NOPASSWD: /sbin/iptables-restore
<appviewx_user> ALL=(ALL) NOPASSWD: /sbin/iptables-save
<appviewx_user> ALL=(ALL) NOPASSWD: /sbin/ip6tables-restore
<appviewx_user> ALL=(ALL) NOPASSWD: /sbin/ip6tables-save
<appviewx_user> ALL=(ALL) NOPASSWD: /sbin/restorecon -R -i
/etc/systemd/system/k3s.service
<appviewx_user> ALL=(ALL) NOPASSWD: /sbin/restorecon -R -i
/etc/systemd/system/k3s.service.env
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/rm -rf
<cc_tar_file_location>/k3s-data-dir
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/sh -c umount "$0" && rm -rf "$0"
<cc_tar_file_location>/k3s-data-dir/kubelet/pods/*
<appviewx_user> ALL=(ALL) NOPASSWD: /bin/sh -c rm -rf
<cc_tar_file_location>/deps/logs && rm -rf
<cc_tar_file_location>/deps/properties/inventory && rm -rf
<cc_tar_file_location>/deps/*.log && rm -rf
<cc_tar_file_location>/deps/utils/*.yaml
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

a. Replace <appviewx_user> with the username intended for installing the cloud connector.
b. Replace <cc_tar_file_location> with the full path to the directory where the *install.sh* script or the *cloud connector tar file* is located.

2. Log in as the root user on the machine where the Cloud Connector is to be installed.
3. Execute the following command:

```
visudo
```

4. Copy the content of the text file (in step 1) and paste it into the /etc/sudoers file, directly below the line "root ALL=(ALL) ALL".
5. Save the **/etc/sudoers** file and switch back to the intended user (which was replaced in the text file). Execute the command:

```
su - <appviewx_user>isudo
```

Installing the Cloud Connector

1. Create and download a cloud connector package tar from the UI and move the tar, license files to the cloud connector installation environment.
2. Switch to the directory where the Cloud connector tar is present.
3. To extract the installer, from the downloaded package, extract the tar.gz file using the command given below:

```
tar -zxvf <filename>.tar.gz
```

Example:

```
tar - zxvf pesrv07-test-94-99-appviewx-appviewx-net-cloud-connector.tar.gz
```

4. Execute the following command:

```
vi deps/tools/install_k3s.sh
```

5. Search for chown in the file and comment the lines adding a “#” at the start of each line.

There will be 3 occurrences of chown, and all of these should be commented as below:

```
# $SUDO chown root:root ${TMP_BIN}
```

```
# $SUDO chown root:root ${KILLALL_K3S_SH}
```

```
# $SUDO chown root:root ${UNINSTALL_K3S_SH}
```

6. Save and exit the **install_k3s.sh** file.
7. To install the cloud connector, trigger the **install.sh** script below.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
./install.sh
```

The cloud connector is installed with the specified user with the minimal sudo privileges.

Setting up the AppViewX Cloud Connector via OVA using the Automated Script

Note: The steps outlined in the following subsections are **specifically** for creating a virtual machine for OVA deployment for setting up the AppViewX Cloud Connector. For instructions on OVA deployment for setting up the AppViewX product, click here.

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 Ubuntu version 22.04
- Hardened OVA with all security issues addressed

Note: Detailed instructions for updating the AppViewX virtual image from the AppViewX repository are documented here.

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

The following sections will guide you through the steps for deploying the AppViewX virtual machine and installing the AppViewX Cloud Connector for

- an on-prem deployment
- AWS
- Azure
- GCP

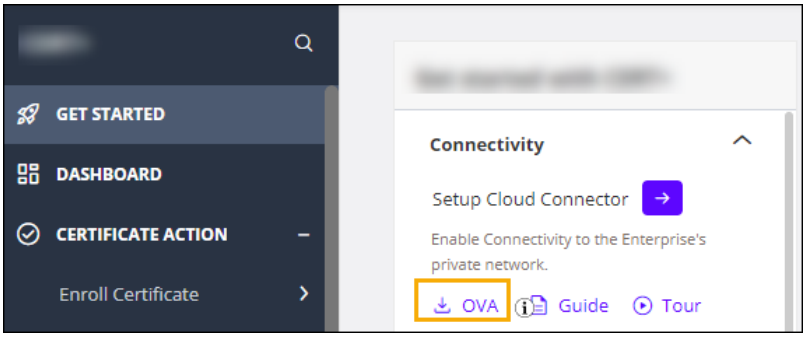
- Setting up the AppViewX Cloud Connector using a Virtual Image for an On-Prem Deployment
- Setting up the AppViewX Cloud Connector using a Virtual Image on AWS

Copyright © 2025 AppViewX Inc. All Rights Reserved.

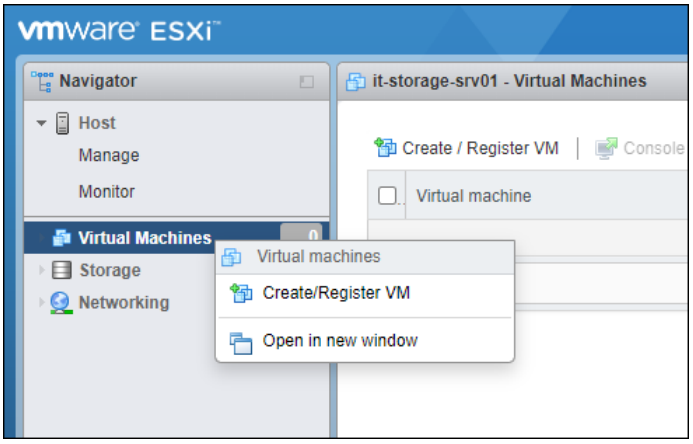
- Setting up the AppViewX Cloud Connector using a Virtual Image on Azure
- Setting up the AppViewX Cloud Connector using a Virtual Image on GCP

Setting up the AppViewX Cloud Connector using a Virtual Image for an On-Prem Deployment

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



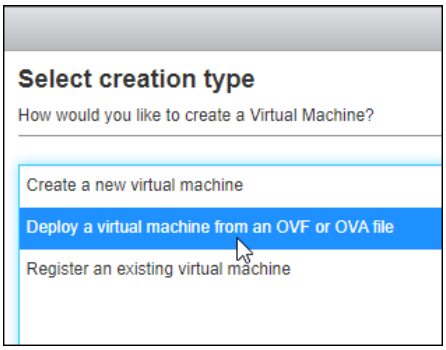
2. Log in to the **vmware** client.
3. From the **Navigation** pane on the left, right click **Virtual Machines**.
4. Click **Create/Register VM**.



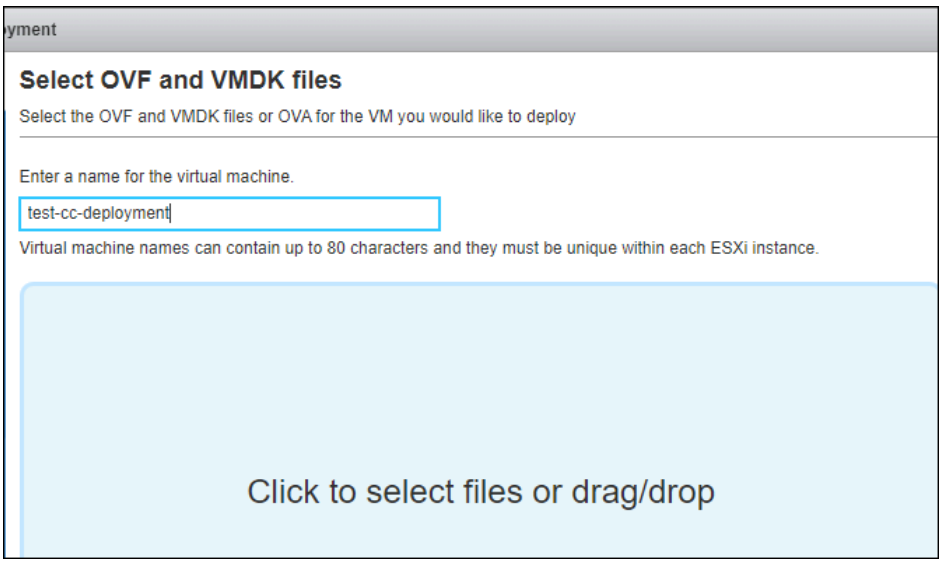
The **New Virtual machine** window is displayed.

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.



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

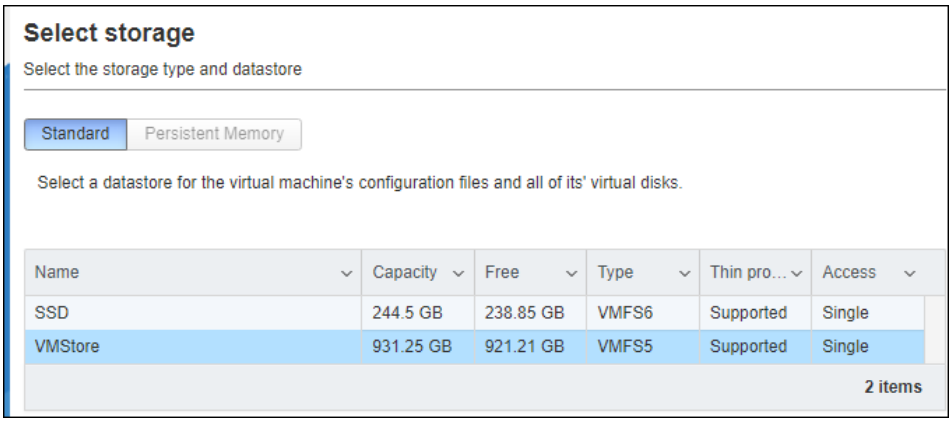


   a. Enter a name for the virtual machine.
      For the purpose of this document, we will name it **test-cc-deployment**.

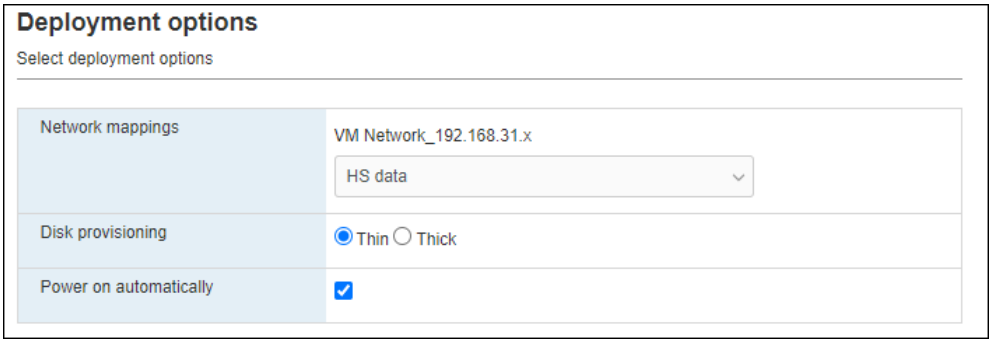
   b. In the **Click to select files or drag/drop** area, click and, from the file explorer, navigate to the location of the file, select the file, and click **Open**.

9. Click **Next**.
10. In the **Select storage** window, from the available options, select a datastore for storing the virtual machine's files and all of its virtual disks.

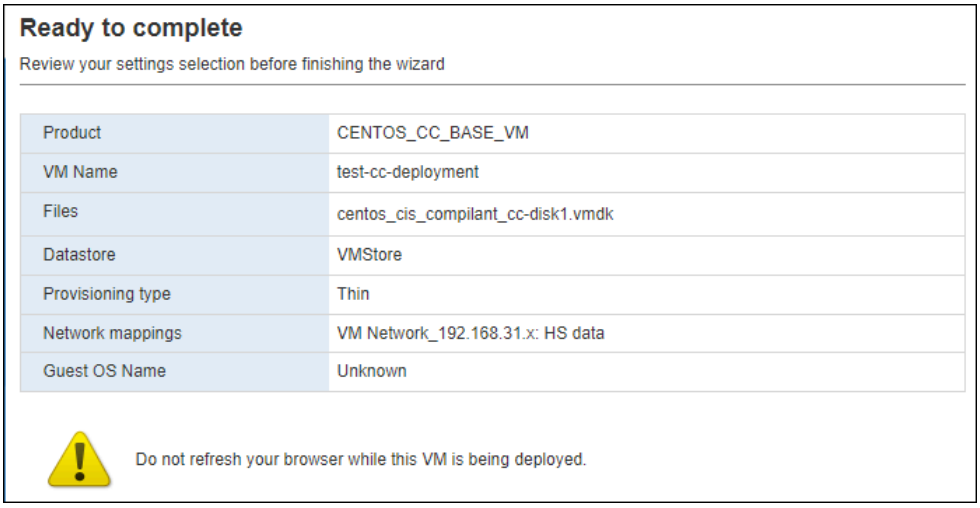
Copyright © 2025 AppViewX Inc. All Rights Reserved.



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

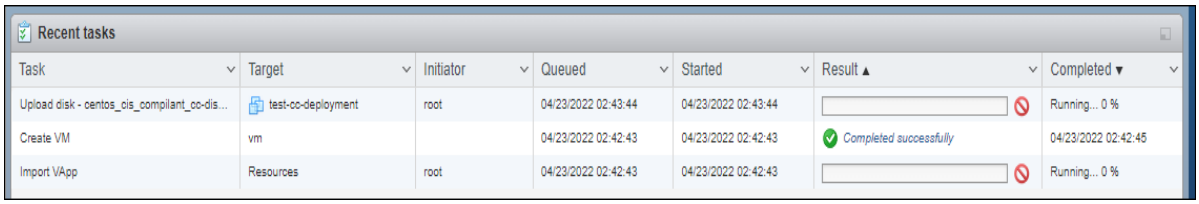


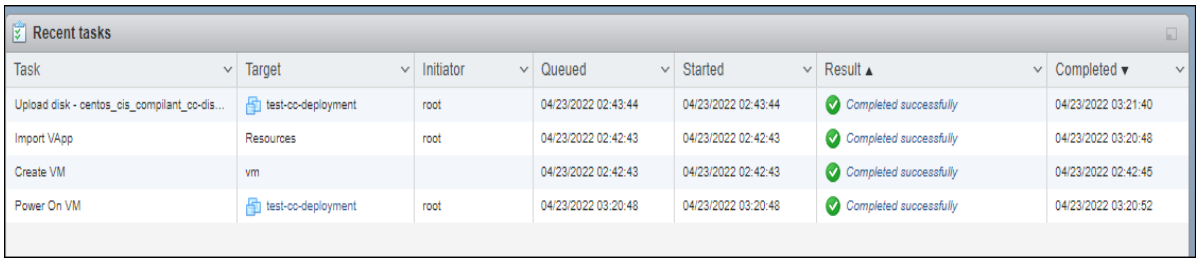
    a. Select the network mapping.
    b. Select the disk provisioning required.
    c. Select the **Power on automatically** checkbox.
13. Click **Next**.
14. In the **Ready to complete** window, review your settings.



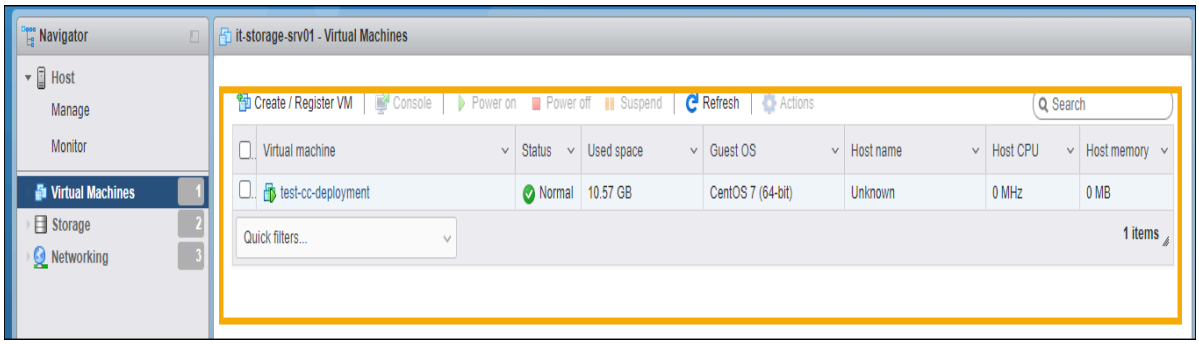
Copyright © 2025 AppViewX Inc. All Rights Reserved.

15. 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:



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

```
==========================================

IPADDR            =  XXX.XXX.XXX.XXX

GATEWAY           =  XXX.XXX.XXX.XXX

NAMESERVER        =  XXX.XXX.XXX.XXX

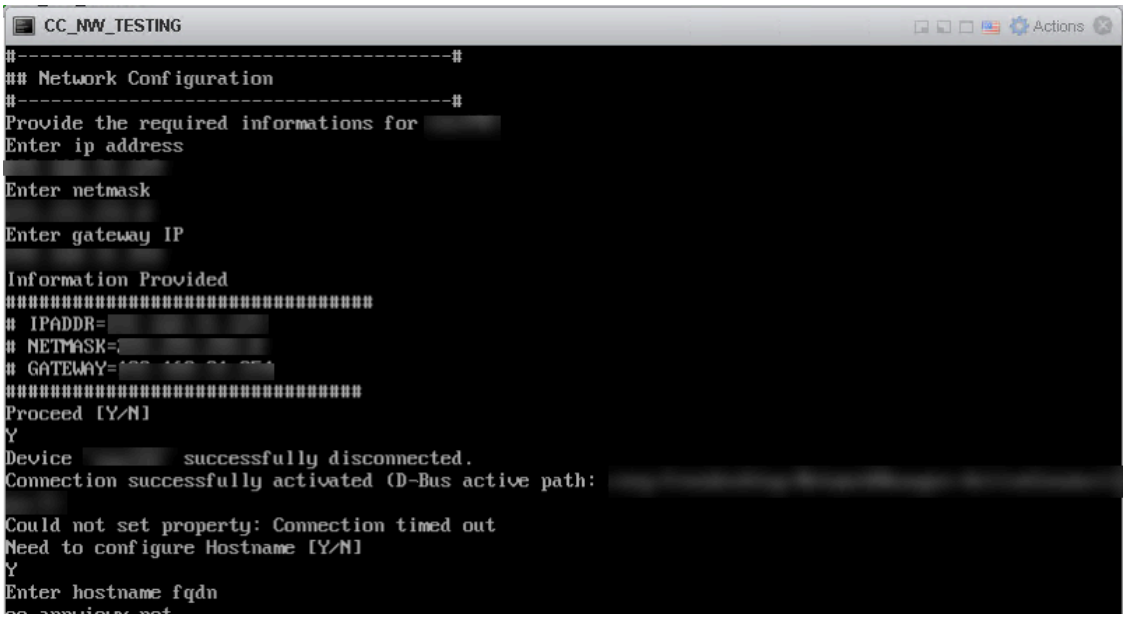
HOSTNAME WITH FQDN   =  server1.example.com

HOST SHORTNAME       =  server1

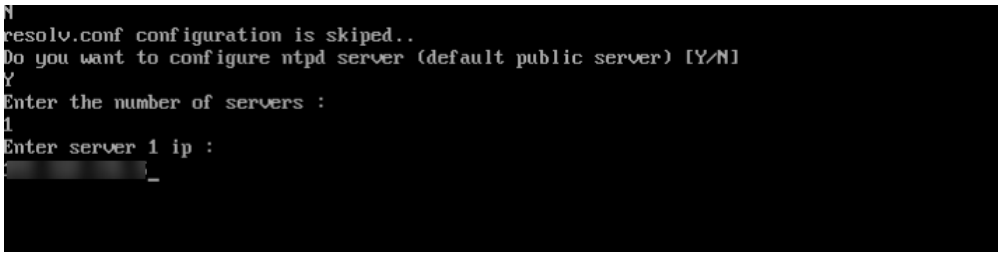
================================
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

For example, refer to the sample screenshot below:



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

    After the execution of this step, if the cloud connector does not exist on this host machine, the following prompt is displayed: **Would you like to opt for automated installation of the cloud connector? (y/n):**.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Note:

- Proceeding with the cloud connector installation at this point will require you to enter the tenant ID and master key for the installation **manually** (you will not be able to copy the details and paste them in response to the prompt).
- By default, the Standalone mode of CC installation is followed, which requires sudo privileges.

20. To install the AppViewX cloud connector at this point, press **y** and then press **Enter**.

Note: If you enter **n** here (that is, you choose to **not** go ahead with the installation here), **skip steps 21 to 25**. You will be reprompted for the automated installation after you login to the VM as the appviewx user.

21. When prompted **Please enter your AppViewX Cloud tenant name or ID**, enter the required details. Internet connectivity on the host machine is validated. The installation proceeds only if the host machine has internet access.

Note: If internet access is unavailable, you will be prompted to proceed with installation using a proxy. Enter **y** and for instructions on proceeding with the installation with a proxy, click here.

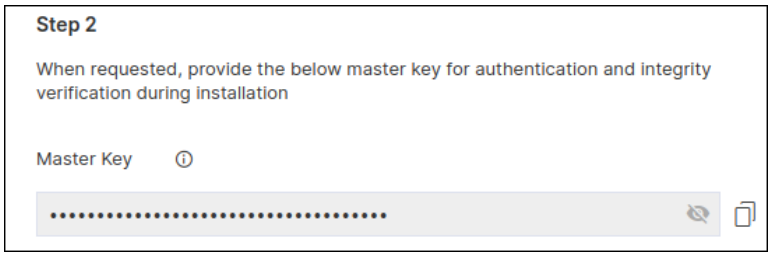
22. When prompted to **Please provide the master key**, enter the required details.
    To retrieve the master key:

    a. Go to (**Menu**) > **Platform** > **Connectivity** > **Cloud Connector**.
    The **Settings :: Cloud Connector** inventory page is displayed.

    b. From the cloud connector details banner, under **Automate Cloud Connector Setup**, click **Steps to Automate Setup**.

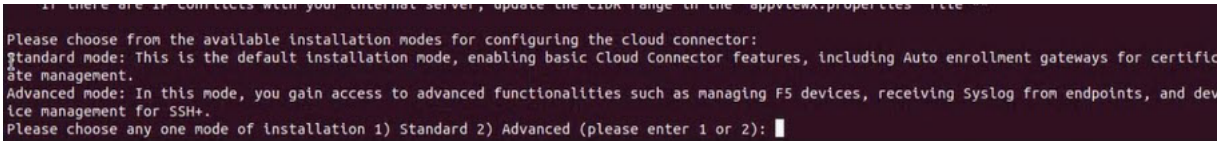
    c. From the **Automated Cloud Connector Setup** window, under **Step 2**, copy the **Master Key**.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



d. Paste the master key in the terminal window and press **Enter**.

23. When prompted to **Please enter Datacenter**, enter the name of the data center on which this cloud connector will be deployed.
    The following prompt is displayed:



    AppViewX offers the following two modes for installing the cloud connector:

    - **Standard** (default): This installation mode will, by default, trigger the prerequisite check and then automatically enable only the basic cloud connector features including the auto enrollment gateways for certificate management and excluding advanced features such as managing F5 devices, receiving syslogs from endpoints, and device management for SSH+.

      Note: To enable these advanced features after the installation is done, use the avxctl command line tool and run `./avxctl enable –help` to see the syntax and possible values. For example, to enable SSH, use the command `./avxctl --ssh`.

      Important: Enabling these advanced features will cause a downtime of 3-5 minutes, because the cloud connector will restart.

    - **Advanced**: Along with checking prerequisites and enabling the basic cloud connector features, this installation mode will display prompts for you to choose which of the above listed advanced features you want to enable for the cloud connector.

      Important: Enabling these advanced features will cause a downtime of 3-5 minutes, because the cloud connector will restart.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

24. Read the descriptions for the **standard** and **advanced** modes, and enter your choice. (Enter **1** for standard mode and **2** for advanced mode.) If you've selected **1**, the cloud connector is installed with its basic features and, on successful completion of the setup, a corresponding instance of this cloud connector is displayed in the inventory. Here, skip step **25**.

25. **Note:** Enabling auto-enrollment protocols is recommended only if the cloud connector is being installed in a demilitarized zone (DMZ) or devices in a restricted environment (that disables them from connecting to the <tenant>-aep directly.

    When prompted to enable auto-enrollment protocols, enter **y** and enter the protocol name(s) you want to enable. For instructions on enabling auto-enrollment protocols, click here.

    Note: By default, only the AppViewX certificate is enabled for auto-enrollment. To enable custom certificate for auto-enrollment:
    a. Execute the command `./avxctl upgrade gateway-cert`.
    b. When prompted, enter the location of the custom certificate.

    On successful completion of the setup, a corresponding instance of this cloud connector is displayed in the inventory.

    Note: The cloud connector installation on a OVA-based host machine will not prompt you to select if you want to manage F5 Big-IP devices. However, after the cloud connector has been installed you can copy the **iControl jar** and **axis jar** in the deps/external_libs folder and restart the starter and platform pods (click here for instructions), to enable this feature.

26. After the script is executed, when prompted, login to the VM as the appviewx user, using the credentials shared by AppViewX's customer support team.

    Note: Root user access is required for maintaining the OS configuration and for patching security updates. Since direct root access is not provided, you can:
    a. Login as the **appviewx** user.
    b. Switch to the root user by executing the command `sudo -i`.

    Note: It is recommended to change the default credentials after the first login.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

If the cloud connector has not been installed already, you will be reprompted to proceed with the installation.

27. When prompted **Would you like to opt for automated installation of the cloud connector? (y/n):**, enter **y**.
    If you enter **n** here (that is, you choose to not go ahead with the installation here), **skip step 27**. You will be prompted again for the automated installation every time you login again to the VM as the appviewx user (until the cloud connector is installed).
28. Repeat steps **21** to **24** above.
29. When prompted **Are you sure you want to continue connecting**, press **y** and then press **Enter**.
30. Login to the host machine on which the OVA has been deployed.
31. If the Cloud Connector has been installed, go to **step 39** to approve the cloud connector installation.
    If the AppViewX Cloud Connector has not been installed on the host machine till this point, the following prompt will be displayed again: **Would you like to opt for automated installation of the cloud connector? (y/n):**
32. Enter **y**.
33. When prompted **Please enter your AppViewX Cloud tenant name or ID**, enter the required details.
    Internet connectivity on the host machine is validated. The installation proceeds only if the host machine has internet access.

    Note: If internet access is unavailable, you will be prompted to proceed with installation using a proxy. Enter **y** and for instructions on proceeding with the installation with a proxy, click here.

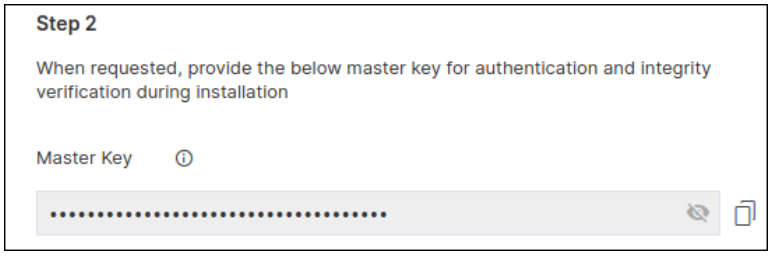
34. When prompted to **Please provide the master key**, enter the required details.
    To retrieve the master key:

    a. Go to (**Menu**) > **Platform** > **Connectivity** > **Cloud Connector**.
    The **Settings :: Cloud Connector** inventory page is displayed.

    b. From the cloud connector details banner, under **Automate Cloud Connector Setup**, click **Steps to Automate Setup**.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

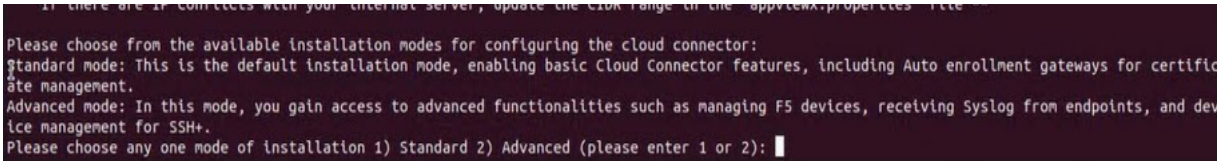
c. From the **Automated Cloud Connector Setup** window, under **Step 2**, copy the **Master Key**.



35. Paste the master key in the terminal window and press **Enter**.
    The cloud connector installation script will check for the prerequisites and trigger the cloud connector installation.

    When the cloud connector instance is successfully installed, a corresponding entry will be listed in the cloud connector inventory.

36. When prompted, **Please enter Datacenter**, enter the name of the data center on which this cloud connector will be deployed. The following prompt is displayed:



    AppViewX offers the following two modes for installing the cloud connector:

    - **Standard** (default): This installation mode will, by default, automatically enable only the basic cloud connector features including the auto enrollment gateways for certificate management and excluding advanced features such as managing F5 devices, receiving syslogs from endpoints, and device management for SSH+.
    - **Advanced**: Along with enabling the basic cloud connector features, this installation mode will display prompts for you to choose which of the above listed advanced features you want to enable for the cloud connector.

37. Read the descriptions for the standard and advanced modes, and enter your choice. (Enter **1** for standard mode and **2** for advanced mode.)

    If you've selected **1**, the cloud connector is installed with its basic features and, on successful completion of the setup, a corresponding instance of this cloud connector is displayed in the inventory. Here, skip step **39**. If you've selected **2**, you will be prompted to enable auto-enrollment protocols.

38. **Note:** This prompt is displayed only for the advanced installation mode

Copyright © 2025 AppViewX Inc. All Rights Reserved.

> **Note:** Enabling auto-enrollment protocols is recommended only if the cloud connector is being installed in a demilitarized zone (DMZ) or devices in a restricted environment (that disables them from connecting to the <tenant>-aep directly.

When prompted to enable auto-enrollment protocols, enter **y**. If you've entered **y**, enter the required protocol name(s). .

> **Note:** By default, only the AppViewX certificate is enabled for auto-enrollment. To enable custom certificate for auto-enrollment:
>
> a. Execute the command `./avxctl upgrade gateway-cert`.
>
> b. When prompted, enter the location of the custom certificate.

On successful completion of the setup, a corresponding instance of this cloud connector is displayed in the inventory.

> **Note:** The cloud connector installation on a OVA-based host machine will not prompt you to select if you want to manage F5 Big-IP devices. However, after the cloud connector has been installed you can copy the **iControl jar** and **axis jar** in the deps/external_libs folder and restart the starter and platform pods (click here for instructions), to enable this feature.

39. To approve the cloud connector installation:

    a. Go to (**Menu**) > **Platform** > **Connectivity** > **Cloud Connector**.
    The **Settings :: Cloud Connector** inventory page is displayed.

    b. For the cloud connector instance installed, from the **Actions** column, click **Approve**.

> **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 using a Virtual Image on AWS

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Prerequisites

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

  When the AppViewX AMI is successfully shared with your customer account, AppViewX will notify you of this development via email.
- From your AppViewX Onboarding Engineer, get the default password for the **appviewx** user.

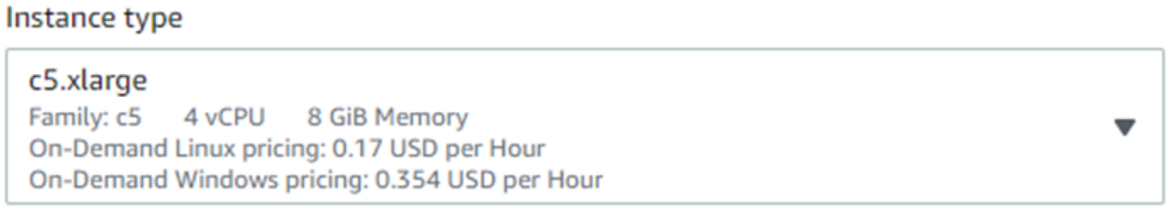
To install the AppViewX Cloud Connector on AWS, you will need a virtual machine that is preconfigured for the operating system and software stack prerequisites.AWS uses AMI to create pre-configured EC2 instances as per AppViewX standards and requirements.

To create an EC2 instance using the AppViewX AMI:

1. Login to the AWS Management Console and go to **EC2** > **Images** > **AMIs**.
   The **Amazon Machine Images (AMIs)** page is displayed.
2. On the **Amazon Machine Images (AMIs)** page, from the **Owned by me** dropdown list, select **Private images**.
   All AMIs with visibility set to private are listed. This list will also have the AMI that is created and shared by AppViewX for your requirements.
3. From this list, select the checkbox for the AMI shared by AppViewX.
4. Click **Launch instance from AMI**.
   The **EC2 > Instances > Launch an instance** page is displayed.
5. Enter the **Name and tags** to be associated with this EC2 instance.

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

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



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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

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

OR

a. To create a new key pair, Click **Create new key pair**.

8. In the **Network settings** section, under **Firewall (security groups)**, as required, create a new security group or select an existing security group.
9. If you select **Select existing security group** in the previous step, from the **Common security groups** dropdown list, select the required security group.
10. From the bottom-right corner of the screen, click **Launch instance**.

    The **Launching instance** page is displayed, which shows you the progress of the launch. As soon as the launch is initiated, you will get a success message.
11. Under **Success**, click **Launch log** to review the instance details.
12. From the page name (**EC2 > Instances > Launch an instance**), click **Instances** to go back to the previous page.
13. From the list of instances, select the AWS instance just created.
14. To login to this AWS instance using the key pair .pem file:

    a. Execute one of the following commands:

    • For Ubuntu:

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

    • For Amazon Linux 2:

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

    b. To switch to the **sudo** user, execute the following command: `sudo -i`

    The following prompt is displayed: **Would you like to opt for automated installation of the cloud connector? (y/n):**

    c. Enter **n**.

    This question is asked here by default. However, since the nameserver has not been configured yet, it is **not recommended** to proceed with the installation because the connection with the cloud connector and mothership will not be established.

    d. To add an entry for the nameserver in the **resolv.conf** file, execute the following command: `echo "nameserver <IP of nameserver>" > /etc/resolv.conf`

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
vi /etc/hosts


hostnamectl set-hostname “hostname-of-the-vm”
```

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

```
hostname -i

hostname -f

hostname
```

g. To switch to the **appviewx** user, execute the following command: `sudo su – appviewx`
After the execution of this step, if the cloud connector does not exist on this host machine, the following prompt is displayed: **Would you like to opt for automated installation of the cloud connector? (y/n):**.

> **Note:** Proceeding with the cloud connector installation at this point will require you to enter the tenant ID and master key for the installation **manually** (you will not be able to copy the details and paste them in response to the prompt). It is, therefore, recommended to **not opt** for the installation at this point in the deployment process.

To install the AppViewX cloud connector at this point, press **y** and then press **Enter**.

After the execution of this step, a script is executed to validate if the cloud connector exists on this host machine and the following prompt is displayed: **Would you like to opt for automated installation of the cloud connector? (y/n):**.

> **Note:** Proceeding with the cloud connector installation at this point will require you to enter the tenant ID and master key for the installation **manually** (you will not be able to copy the details and paste them in response to the prompt). It is, therefore, recommended to **not opt** for the installation at this point in the deployment process.

To install the AppViewX cloud connector at this point, press **y** and then press **Enter**.

15. To install the AppViewX cloud connector at this point, enter **y**.

> **Note:** If you enter **n** here (that is, you choose to **not** go ahead with the installation here), **skip steps 16 to 20**. You will be reprompted for the automated installation after you login to the VM as the appviewx user.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

16. When prompted **Please enter your AppViewX Cloud tenant name or ID**, enter the required details. Internet connectivity on the host machine is validated. The installation proceeds only if the host machine has internet access.

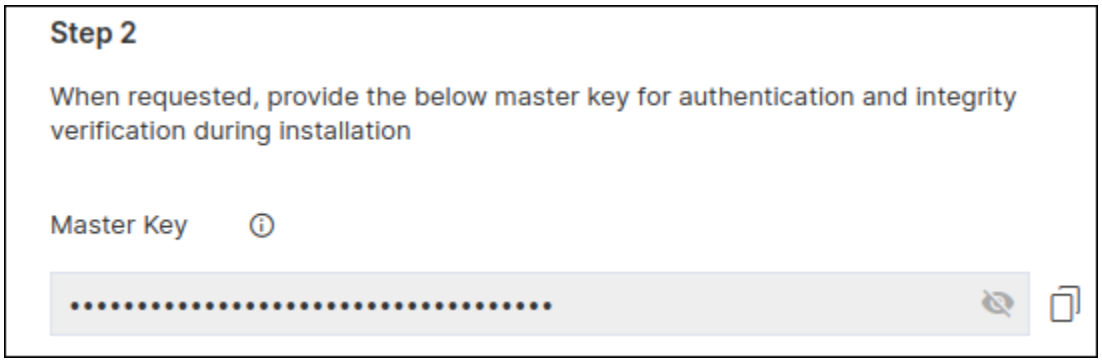
> **Note:** If internet access is unavailable, you will be prompted to proceed with installation using a proxy. Enter **y** and for instructions on proceeding with the installation with a proxy, click here.

17. When prompted to **Please provide the master key**, enter the required details.
    To retrieve the master key:

    a. Go to (**Menu**) > **Platform** > **Connectivity** > **Cloud Connector**.
    The **Settings :: Cloud Connector** inventory page is displayed.

    b. From the cloud connector details banner, under **Automate Cloud Connector Setup**, click **Steps to Automate Setup**.

    c. From the **Automated Cloud Connector Setup** window, under **Step 2**, copy the **Master Key**.



    d. Paste the master key in the terminal window and press **Enter**.

    The cloud connector installation script will check for the prerequisites and trigger the cloud connector installation.

    When the cloud connector instance is successfully installed, a corresponding entry will be listed in the cloud connector inventory.

18. When prompted, **Please enter Datacenter**, enter the name of the data center on which this cloud connector will be deployed.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

19. **Note:** Enabling auto-enrollment protocols is recommended only if the cloud connector is being installed in a demilitarized zone (DMZ) or devices in a restricted environment (that disables them from connecting to the <tenant>-aep directly.

    When prompted to enable auto-enrollment protocols, enter **y** and enter the protocol name(s) you want to enable. For instructions on enabling auto-enrollment protocols, click here.

    Note: By default, only the AppViewX certificate is enabled for auto-enrollment. To enable custom certificate for auto-enrollment:
    a. Execute the command `./avxctl upgrade gateway-cert`.
    b. When prompted, enter the location of the custom certificate.

    On successful completion of the setup, a corresponding instance of this cloud connector is displayed in the inventory.

    Note: The cloud connector installation on a OVA-based host machine will not prompt you to select if you want to manage F5 Big-IP devices. However, after the cloud connector has been installed you can copy the iControl jar in the deps/external_libs folder (click here for instructions) and restart the starter and platform pods (click here for instructions), to enable this feature.

    Note: (Optional, required only for password authentication) In order to successfully execute the installation, AppViewX needs to run a script for which authentication via the **.pem** file needs to be bypassed. To do this, execute the following commands:

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

    ```
    sudo systemctl restart sshd
    ```

20. To approve the cloud connector installation:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

a. Go to (**Menu**) > **Platform** > **Connectivity** > **Cloud Connector**.
The **Settings :: Cloud Connector** inventory page is displayed.

b. For the cloud connector instance installed, from the **Actions** column, click **Approve**.

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 using a Virtual Image on Azure

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

To deploy the AppViewX virtual machine for Azure:

1. Go to https://release.appviewx.com/Login and, from **Overview**, navigate to release Example: **20XX.0.0**.
2. Scroll down to Production Images and download the latest artifact of Azure CC VHD, **AppViewX-20XX.X.X.10-CC-Ubuntu-Azure-ddmmmyyy-vhd.tar.gz**.
3. Untar the downloaded artifact.
   ```
   tar -xvf AppViewX-20XX.X.X.10-CC-Ubuntu-Azure-ddmmmyyy-vhd.tar.gz
   ```
4. Download the **Azure Storage Explorer** from here.
   The **Azure Storage Explorer** is a desktop application that provides you with a GUI for easily managing your Azure resources.

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

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Descriptions for the resources and their corresponding values

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

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

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

    `ssh appviewx@<ip/hostname>`
13. To add an entry for the nameserver in the **resolv.conf** file, execute the following command: `echo "nameserver <IP of nameserver>" > /etc/resolv.conf`
14. Update the /etc/hosts file for the IP and the hostname of the VM created, using the following commands:

    ```
    vi /etc/hosts


    hostnamectl set-hostname “hostname-of-the-vm”
    ```
15. To validate the update to the **/etc/hosts** file, execute the following commands:

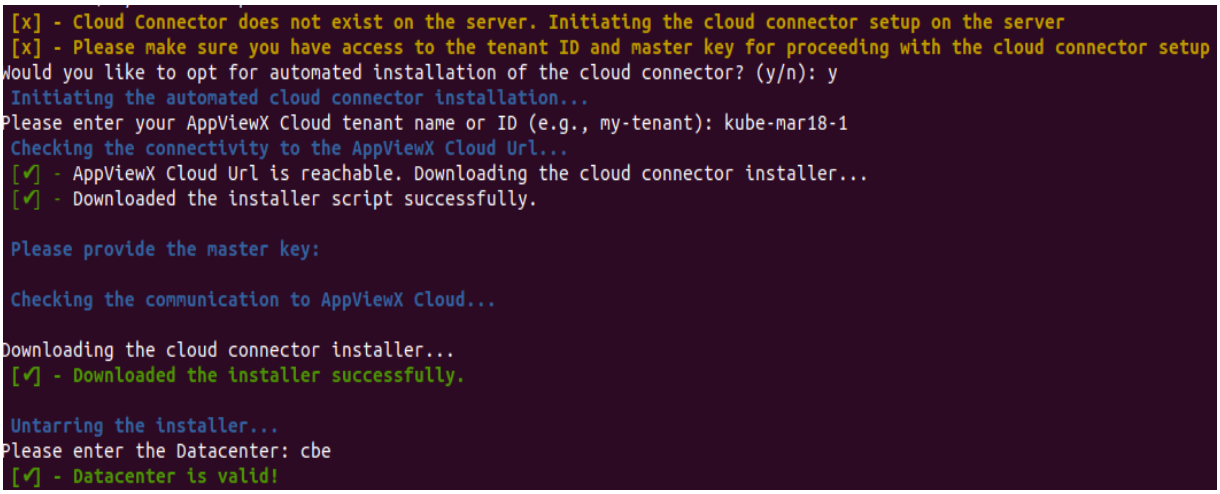
Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
hostname -i

hostname -f

hostname
```

16. To switch to the **appviewx** user, execute the following command: `sudo su – appviewx`
    After the execution of this step, if the cloud connector does not exist on this host machine, the following prompt is displayed: **Would you like to opt for automated installation of the cloud connector? (y/n):**.
17. To install the AppViewX cloud connector at this point, enter **y**.



18. When prompted **Please enter your AppViewX Cloud tenant name or ID**, enter the required details.
    Internet connectivity on the host machine is validated. The installation proceeds only if the host machine has internet access.

> **Note:** If internet access is unavailable, you will be prompted to proceed with installation using a proxy. Enter **y** and for instructions on proceeding with the installation with a proxy, click here.

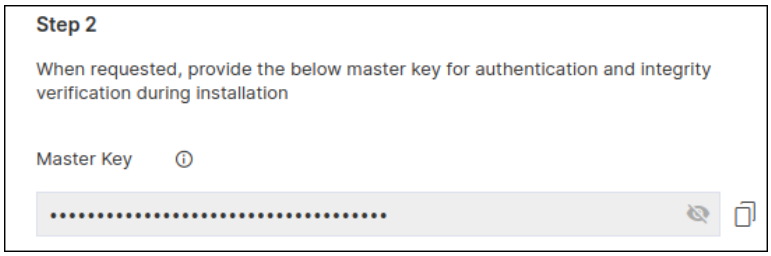
19. When prompted to **Please provide the master key**, enter the required details.
    To retrieve the master key:

    a. Go to (**Menu**) > **Platform** > **Connectivity** > **Cloud Connector**.
    The **Settings :: Cloud Connector** inventory page is displayed.

    b. From the cloud connector details banner, under **Automate Cloud Connector Setup**, click **Steps to Automate Setup**.

    c. From the **Automated Cloud Connector Setup** window, under **Step 2**, copy the **Master Key**.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



d. Paste the master key in the terminal window and press **Enter**.

The cloud connector installation script will check for the prerequisites and trigger the cloud connector installation.

When the cloud connector instance is successfully installed, a corresponding entry will be listed in the cloud connector inventory.

20. When prompted, **Please enter Datacenter**, enter the name of the data center on which this cloud connector will be deployed.

21. **Note:** Enabling auto-enrollment protocols is recommended only if the cloud connector is being installed in a demilitarized zone (DMZ) or devices in a restricted environment (that disables them from connecting to the <tenant>-aep directly.

When prompted to enable auto-enrollment protocols, enter **y** and enter the protocol name(s) you want to enable. For instructions on enabling auto-enrollment protocols, click here.

Note: By default, only the AppViewX certificate is enabled for auto-enrollment. To enable custom certificate for auto-enrollment:

a. Execute the command `./avxctl upgrade gateway-cert`.

b. When prompted, enter the location of the custom certificate.

On successful completion of the setup, a corresponding instance of this cloud connector is displayed in the inventory.

Note: The cloud connector installation on a OVA-based host machine will not prompt you to select if you want to manage F5 Big-IP devices. However, after the cloud connector has been installed you can copy the iControl jar in the deps/external_libs folder (click here for instructions) and restart the starter and platform pods (click here for instructions), to enable this feature.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

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

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

22. To approve the cloud connector installation:

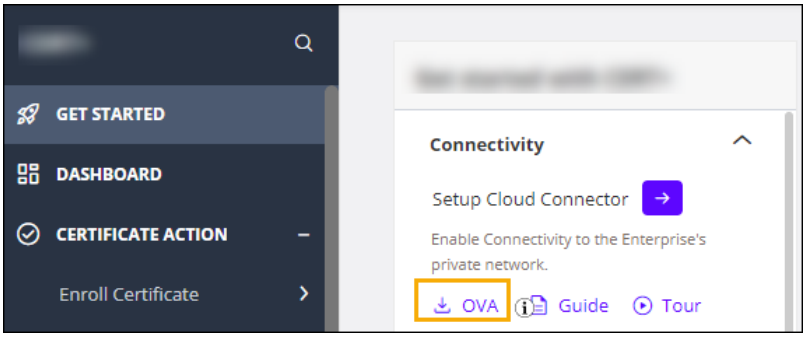
    a. Go to (**Menu**) > **Platform** > **Connectivity** > **Cloud Connector**.
    The **Settings :: Cloud Connector** inventory page is displayed.

    b. For the cloud connector instance installed, from the **Actions** column, click **Approve**.

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 using a Virtual Image on GCP

1. Log in to the GCP console and start a **Cloud Shell** instance.
2. From the command line terminal, create a bucket with the default settings.
   ```
   gsutil mb gs://my-virtual-appliances-bucket
   ```
3. Download the GCP OVA from the AppViewX release portal.
   Alternatively, you can Download the release package in the OVA format, from the respective AppViewX's product line landing page, under **GET STARTED** menu > **Connectivity** section, click OVA.



Copyright © 2025 AppViewX Inc. All Rights Reserved.

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

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

5. Create a virtual instance from the OVA.

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

6. Login to the GCP node using AppViewX credentials.
   `ssh appviewx@<node IP>`
7. Enter the password when prompted.
   The following prompt is displayed: **Would you like to opt for automated installation of the cloud connector? (y/n):**
8. Enter **n**.
   This question is asked here by default. However, since the nameserver has not been configured yet, it is **not recommended** to proceed with the installation because the connection with the cloud connector and mothership will not be established.
9. To switch to the root user, execute the following command: `sudo -i`.
   The following prompt is displayed: **Would you like to opt for automated installation of the cloud connector? (y/n):**
10. Enter **n**.
    This question is asked here by default. However, since the nameserver has not been configured yet, it is **not recommended** to proceed with the installation because the connection with the cloud connector and mothership will not be established.
11. To add an entry for the nameserver in the **resolv.conf** file, execute the following command: `echo "nameserver <IP of nameserver>" > /etc/resolv.conf`
12. Update the /etc/hosts file for the IP and the hostname of the VM created, using the following commands:

```
vi /etc/hosts


hostnamectl set-hostname “hostname-of-the-vm”
```

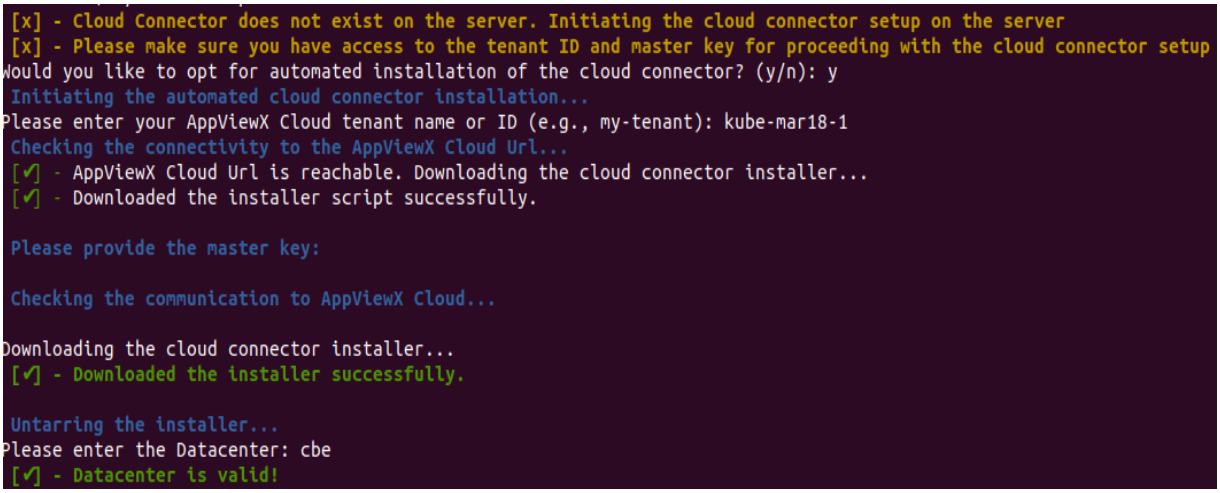
13. To validate the update to the **/etc/hosts** file, execute the following commands:

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
hostname
```

14. To switch to the **appviewx** user, execute the following command: `sudo su – appviewx`
    On login, if the cloud connector does not exist on this host machine, the following prompt is displayed: **Would you like to opt for automated installation of the cloud connector? (y/n):**
15. To install the AppViewX cloud connector at this point, enter **y**.



16. When prompted **Please enter your AppViewX Cloud tenant name or ID**, enter the required details.
    Internet connectivity on the host machine is validated. The installation proceeds only if the host machine has internet access.

> **Note:** If internet access is unavailable, you will be prompted to proceed with installation using a proxy. Enter **y** and for instructions on proceeding with the installation with a proxy, click here.

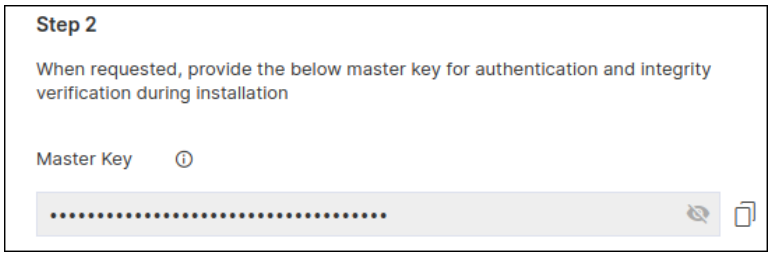
17. When prompted to **Please provide the master key**, enter the required details.
    To retrieve the master key:

    a. Go to (**Menu**) > **Platform** > **Connectivity** > **Cloud Connector**.
    The **Settings :: Cloud Connector** inventory page is displayed.

    b. From the cloud connector details banner, under **Automate Cloud Connector Setup**, click **Steps to Automate Setup**.

    c. From the **Automated Cloud Connector Setup** window, under **Step 2**, copy the **Master Key**.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



d. Paste the master key in the terminal window and press **Enter**.

The cloud connector installation script will check for the prerequisites and trigger the cloud connector installation.

When the cloud connector instance is successfully installed, a corresponding entry will be listed in the cloud connector inventory.

18. When prompted, **Please enter Datacenter**, enter the name of the data center on which this cloud connector will be deployed.

19. **Note:** Enabling auto-enrollment protocols is recommended only if the cloud connector is being installed in a demilitarized zone (DMZ) or devices in a restricted environment (that disables them from connecting to the <tenant>-aep directly.

When prompted to enable auto-enrollment protocols, enter **y** and enter the protocol name(s) you want to enable. For instructions on enabling auto-enrollment protocols, click here.

Note: By default, only the AppViewX certificate is enabled for auto-enrollment. To enable custom certificate for auto-enrollment:

a. Execute the command `./avxctl upgrade gateway-cert`.

b. When prompted, enter the location of the custom certificate.

On successful completion of the setup, a corresponding instance of this cloud connector is displayed in the inventory.

Note: The cloud connector installation on a OVA-based host machine will not prompt you to select if you want to manage F5 Big-IP devices. However, after the cloud connector has been installed you can copy the iControl jar in the deps/external_libs folder (click here for instructions) and restart the starter and platform pods (click here for instructions), to enable this feature.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

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

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

```
sudo systemctl restart sshd
```

20. To approve the cloud connector installation:

    a. Go to (**Menu**) > **Platform** > **Connectivity** > **Cloud Connector**.
    The **Settings :: Cloud Connector** inventory page is displayed.

    b. For the cloud connector instance installed, from the **Actions** column, click **Approve**.

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 OVA using AppViewX User Interface (Manual)

Note: The steps outlined in the following subsections are **specifically** for creating a virtual machine for OVA deployment for setting up the AppViewX Cloud Connector. For instructions on OVA deployment for setting up the AppViewX product, click here.

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:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- Built with Ubuntu version 22.04
- Docker 20.10.5, 27.1.1 pre installed with all required permissions
- Hardened OVA with all security issues addressed

Note: Detailed instructions for updating the AppViewX virtual image from the AppViewX repository are documented here.

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

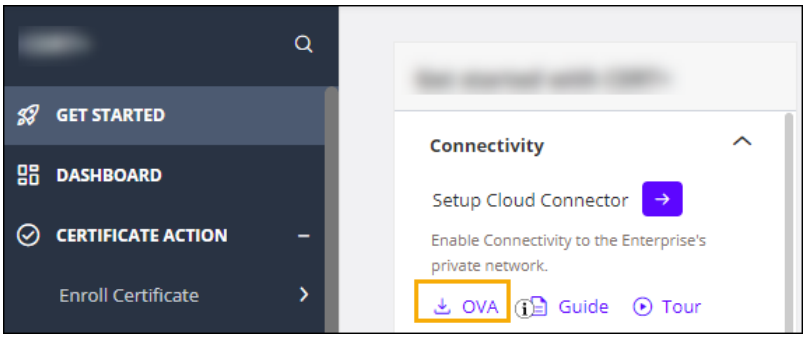
The following sections will guide you through the steps for deploying the AppViewX virtual machine and installing the AppViewX Cloud Connector for

- An on-prem deployment
- AWS
- Azure
- GCP

- Setting up the AppViewX Cloud Connector using a Virtual Image for an On-prem Deployment
- Setting up the AppViewX Cloud Connector using a Virtual Image on AWS
- Setting up the AppViewX Cloud Connector using a Virtual Image on Azure
- Setting up the AppViewX Cloud Connector using a Virtual Image for GCP

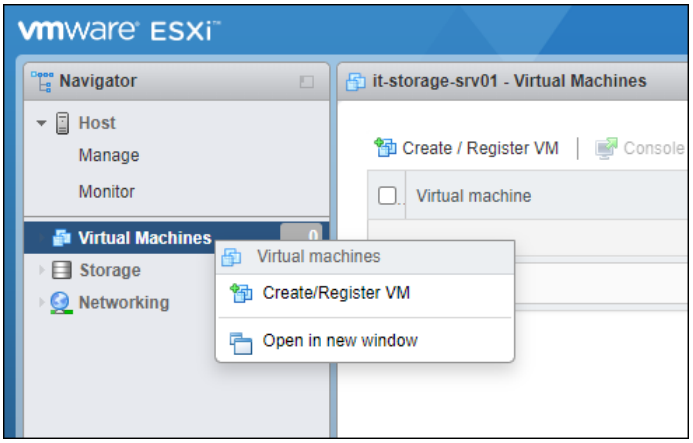
Setting up the AppViewX Cloud Connector using a Virtual Image for an On-prem Deployment

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



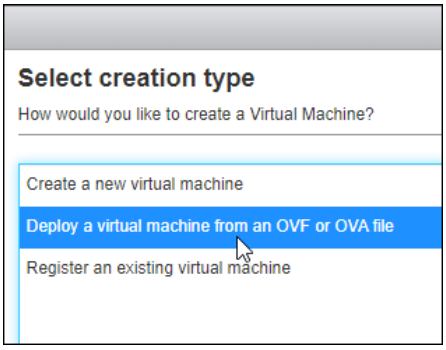
Copyright © 2025 AppViewX Inc. All Rights Reserved.

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



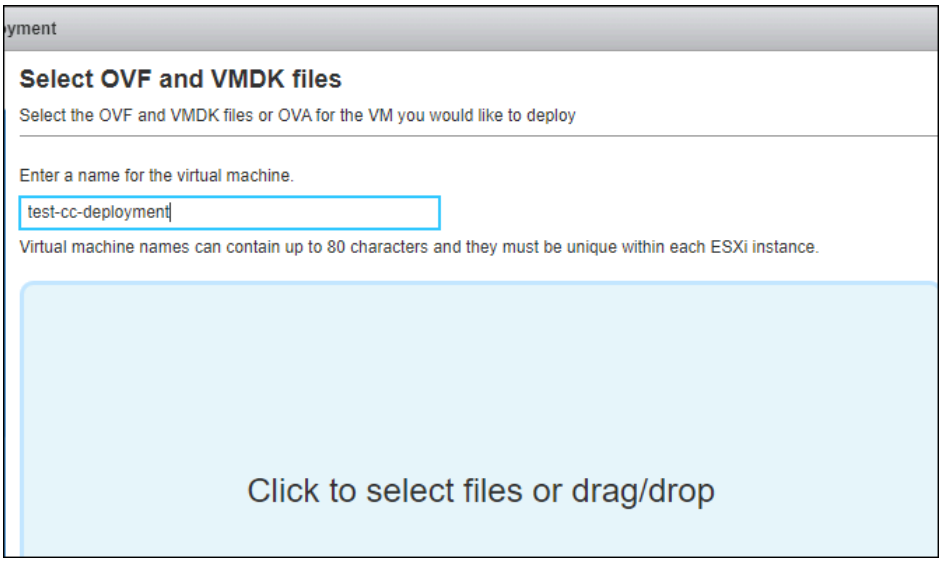
The **New Virtual machine** window is displayed.

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



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

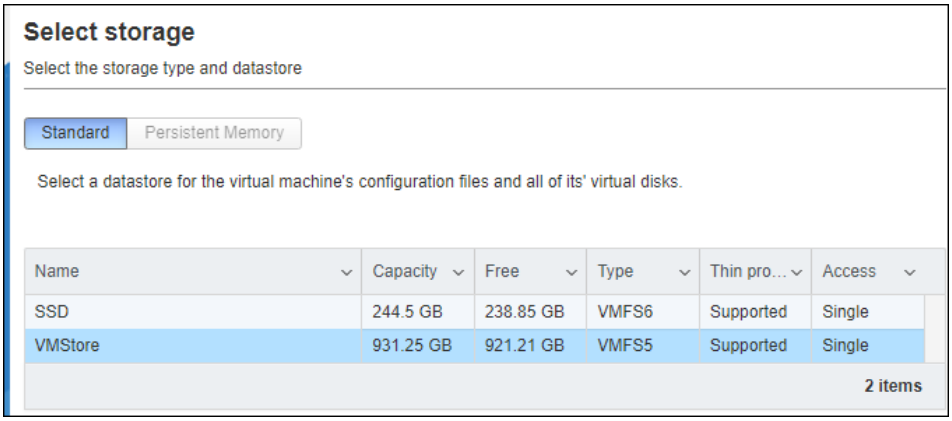
Copyright © 2025 AppViewX Inc. All Rights Reserved.



a. Enter a name for the virtual machine.
   For the purpose of this document, we will name it **test-cc-deployment**.

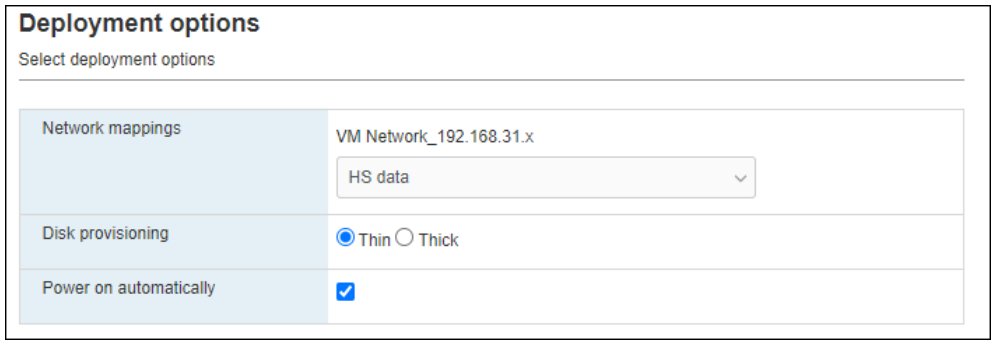
b. In the **Click to select files or drag/drop** area, click and, from the file explorer, navigate to the location of the file, select the file, and click **Open**.

9. Click **Next**.
10. In the **Select storage** window, from the available options, select a datastore for storing the virtual machine's files and all of its virtual disks.



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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

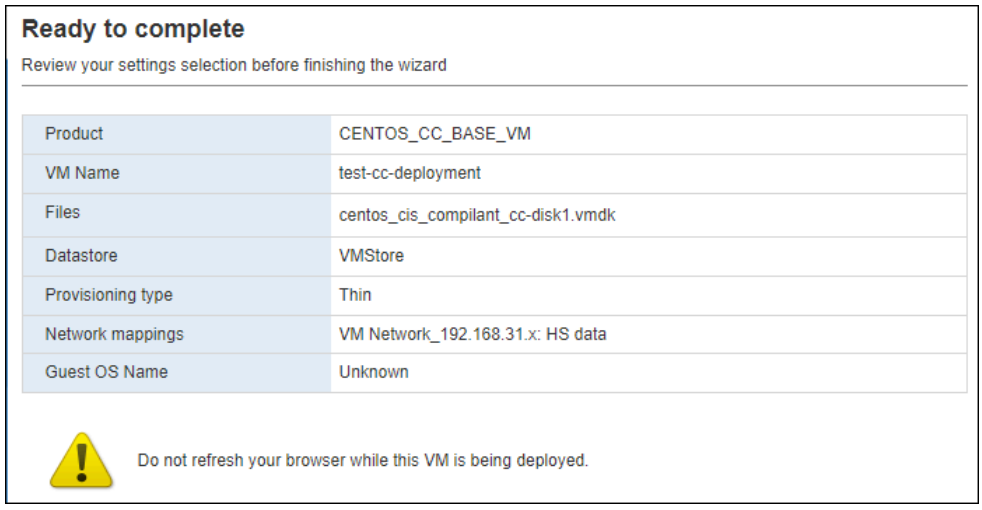


a. Select the network mapping.

b. Select the disk provisioning required.

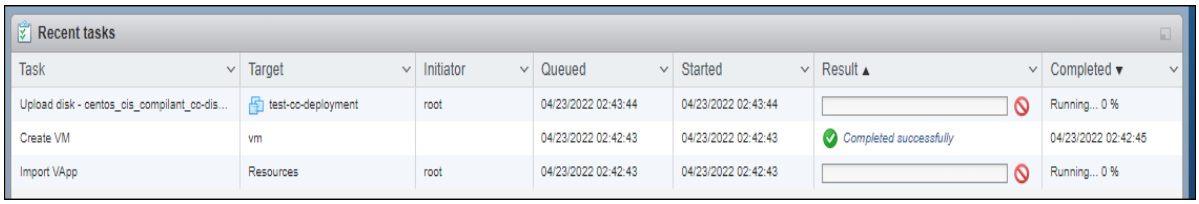
c. Select the **Power on automatically** checkbox.

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

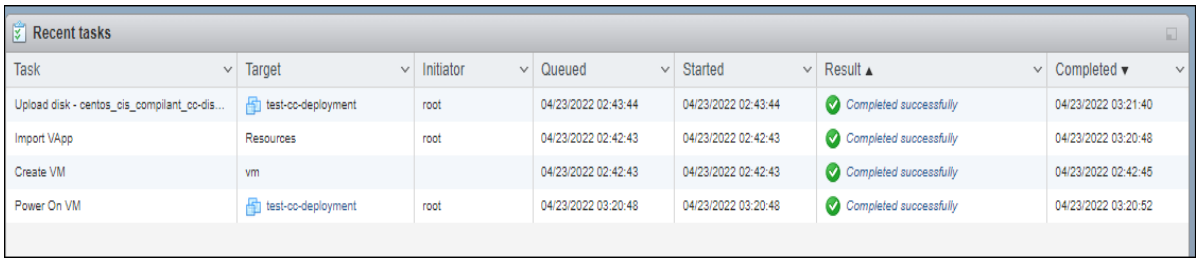


15. Click **Finish**.

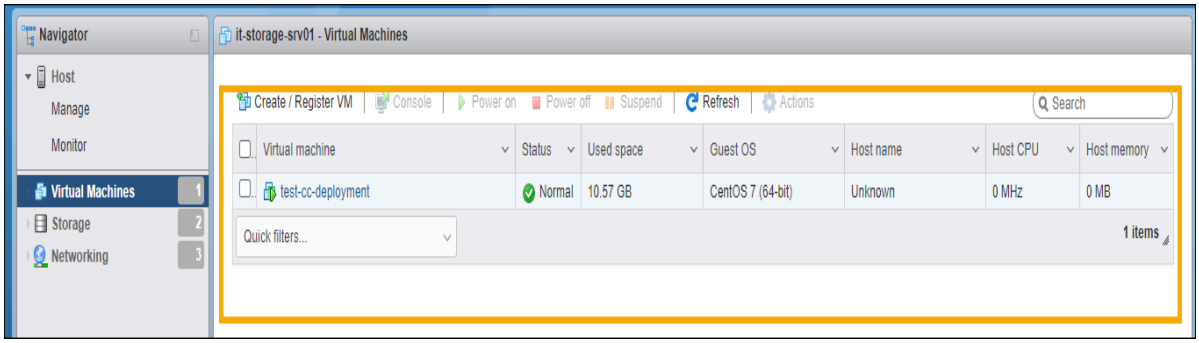
- The progress of the OVA deployment is shown in the **Recent Tasks** section.



Copyright © 2025 AppViewX Inc. All Rights Reserved.



- 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:



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

```
==========================================

IPADDR   = XXX.XXX.XXX.XXX

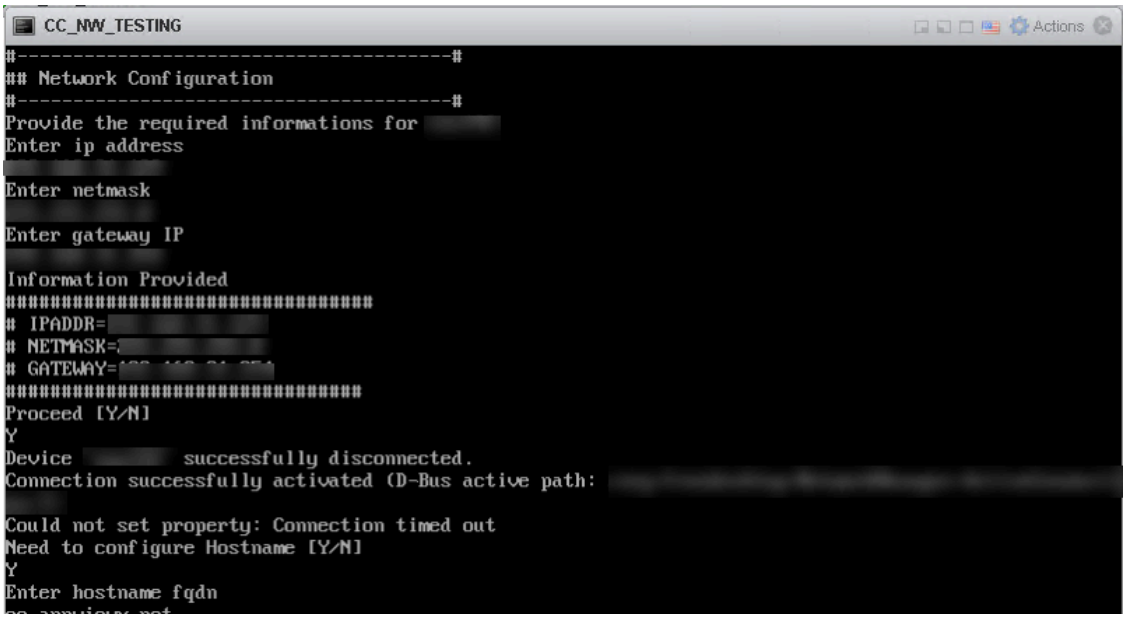
NETMASK  = XXX.XXX.XXX.XXX

GATEWAY  = XXX.XXX.XXX.XXX

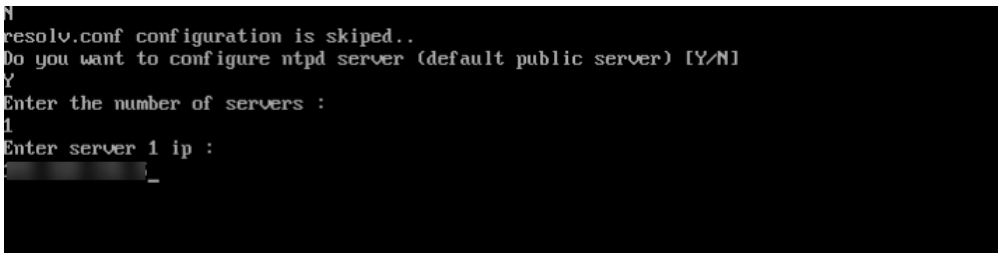
==============================
```

    For example, refer to the sample screenshot below:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

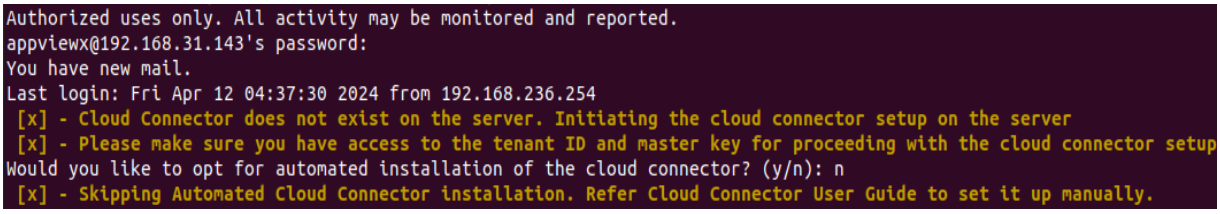


18. To configure the hostname and the DNS, when prompted, press **Y**. If you prefer to configure the hostname and DNS manually, to skip this step, press **N**.
19. To configure an NTP server(s):
    a. When prompted, **Do you want to configure ntpd server (default public server)**press **Y**.



    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**.

    After the execution of this step, if the cloud connector does not exist on this host machine, the following prompt is displayed: **Would you like to opt for automated installation of the cloud connector? (y/n):**.
20. Since these instructions are for setting up the cloud connector via the user interface, enter **n**.



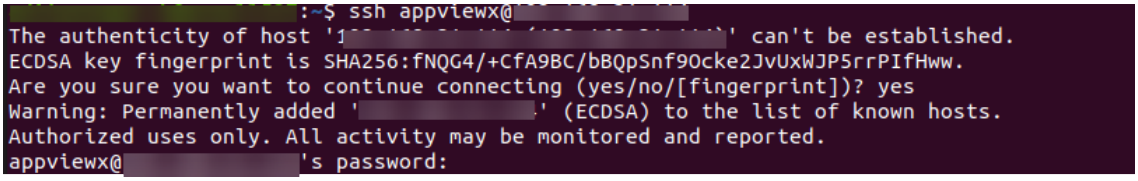
Copyright © 2025 AppViewX Inc. All Rights Reserved.

21. When prompted, login to the VM as the appviewx user, using the credentials shared by AppViewX's customer support team.

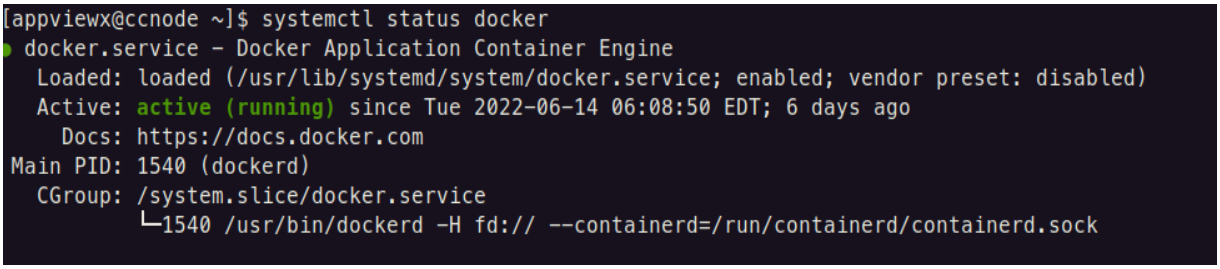
> **Note:** Root user access is required for maintaining the OS configuration and for patching security updates. Since direct root access is not provided, you can:
>
> a. Login as the **appviewx** user.
>
> b. Switch to the root user by executing the command `sudo -i`.

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

22. Since these instructions are for setting up the cloud connector via the user interface, when prompted **Would you like to opt for automated installation of the cloud connector? (y/n):**, enter **y**.
23. When prompted **Are you sure you want to continue connecting**, press **y** and then press **Enter**.



24. 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.



> **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:

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

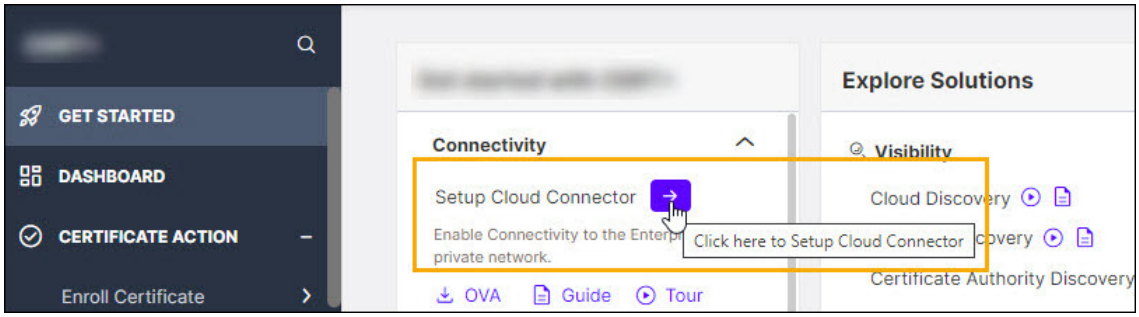
The AppViewX login page is displayed.

26. Login to AppViewX.
27. Navigate to the cloud connector's setup interface.

    There are three ways you can access the interface for setting up the AppViewX Cloud Connector:

    • From the product landing page (that you will see as soon as you have logged in)

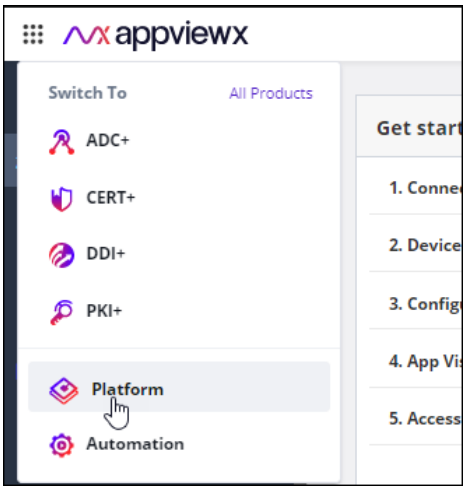
    • Expand the **Connectivity** section and click  .



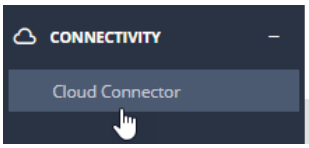
    You will be redirected to the **Settings :: Cloud Connector** page.

    • From the new navigation menu (displayed by default starting product version 2022.1.0 FP3 onwards):

    a. From the menu in the top-right corner of the page, select **Platform**.



    b. From the **Platform** menu, under **Connectivity**, click **Cloud Connector**.



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

    • From the old navigation menu:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

> **Note:** For instructions on switching between the new and the old navigation menus, click here.

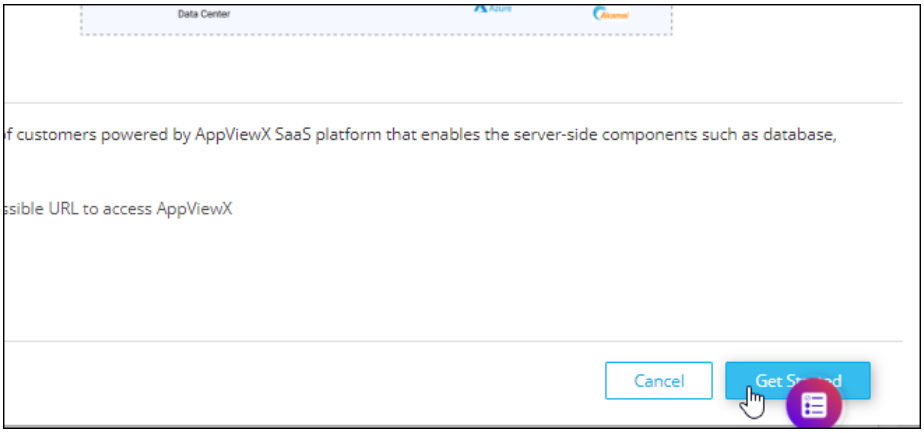
a. From the top right corner of the landing page, click the menu icon.
b. From the menu displayed, navigate to **Settings** > **Cloud Connector**.

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

28. On the **Setting :: Cloud Connector** page, click **Add Cloud Connector**.
The **Cloud Connector Setup** screen is displayed.

The landing page gives you a quick introduction to the AppViewX Cloud Connector, with a graphical representation of how the infrastructure is deployed and works.

29. To start with the process of adding the cloud connector, from the bottom-right corner of the screen, click **Get Started**.



You will be redirected to the **Basic Information** page.

30. On the **Basic Information** page, configure the basic cloud connector settings.

a. To install the cloud connector via the virtual image, from **Installation Type**, select **Virtual Image**.

> **Note:** Click here to read how a virtual image-based installation is different from a native OS installation.

b. In the **Cloud Connector Name (FQDN)** field, enter the hostname of the machine on which the AppViewX Cloud Connector will be installed.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

> **Tip:** To retrieve the hostname, from the command line terminal of the host machine, execute the following command: `hostname -f`:

> **Note:** The hostname entered here is added to the license file that will be generated and downloaded as part of the installer. Therefore, the license file can be used to install the cloud connector only on the machine with the entered hostname and no other.

> **Tip:** The **Setup Cloud Connector** section to the right of the **Basic Information** screen lists hyperlinks to the prerequisites required for setting up the AppViewX Cloud Connector. To read more about what the AppViewX Cloud Connector offers, click **Learn More**.

c. Click **Next**.

31. [Optional] Execute a prerequisite check script.

> **Note:** This is an **optional** step. The prerequisite check script is executed automatically at the time of installing the AppViewX Cloud Connector and the results are shown as a part of the installation logs.

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.

a. On the **Basic Information** screen, under **Setup Cloud Connector**, you will see a list of the installation prerequisites.

   From this list, for **Executing the Prerequisites Check Script**, to download the script, click  .
   The **pre-requisite-check.sh** script file is downloaded.

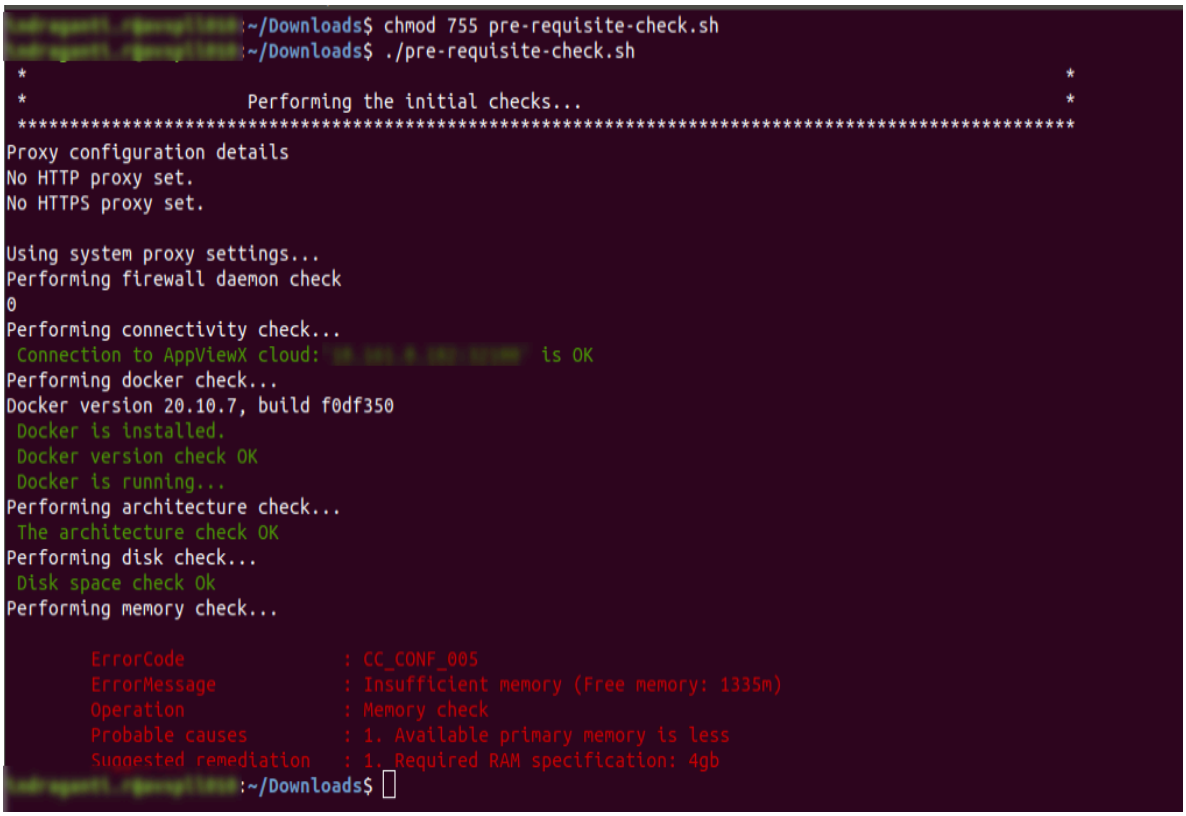
b. Securely copy the **pre-requisite-check.sh** via SCP/SFTP to the host machine where the AppViewX Cloud Connector is to be installed

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

d. Execute the **.sh** prerequisite check script file: `./pre-requisite-check.sh`
   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.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

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.

32. Click **Next**.
    You will be navigated to the **AssignData Center** screen, where, for deploying the AppViewX Cloud Connector, you can either select an existing data center or add a new one.
33. To use an existing data center, select one from the options displayed on the **Assign Data Center** screen.

    Tip: Alternatively, you can use the **Search…** field on this screen to search for an existing data center.

    To add a new data center:

    a. Click **Add Data Center.**

    b. In the **Add Data Center** dialog box, enter a name for the new data center.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

c. Click **Save**.
The new data center will now be displayed on the **Assign Data Center** screen along with the other existing data centers.

d. Select the required data center.

> **Tip:** The **Data Center based routing** section to the right of the **Assign Data Center** screen explains the concept of data center-based routing and how you can achieve high availability. To read more on this, click **Learn More** from the top-right corner of this screen.

34. Click **Next**.
The **Advanced Configuration** screen is displayed.
35. On the **Advanced Configuration** page, to configure the TLS authentication and proxy server settings for your cloud connector:

a. Enter/Select the advanced configuration settings for the AppViewX Cloud Connector.

> **Note:** The **Data center** field is auto-populated based on your selection on the **Assigning a Data Center** screen.

Field	Description
TLS Authentication	**Tip:** The section on the right of the screen gives you a brief context of what is **TLS Authent** • To auto-generate a TLS certificate, select **Auto-generate** (default selection). By default, 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** c • To enter details of a custom TLS certificate, select **Custom**. The **TLS Certificate Password** and **Custom TLS Certificate** fields are displayed. The instructio

Copyright © 2025 AppViewX Inc. All Rights Reserved.

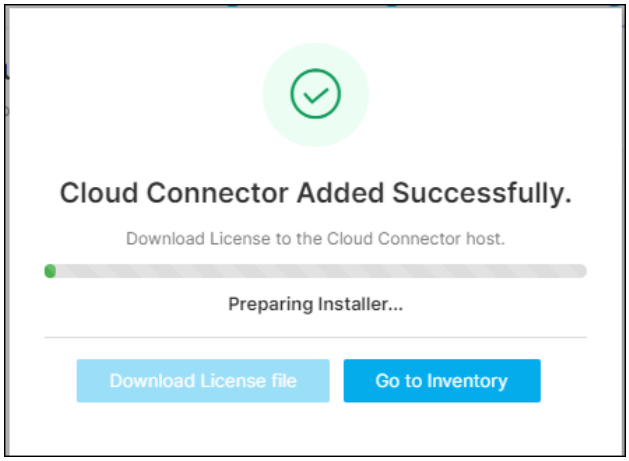
Field	Description
TLS Certificate Password*	**Note:** This field is displayed only if you have selected to enter details of a Custom TLS certi Password of the TLS certificate (that will be uploaded in the next step) **Note:** This is a mandatory field if a Custom TLS certificate is uploaded. AppViewX supports
TLS Certificate	**Note:** This field is displayed only if you have selected to enter details of a Custom TLS certi To upload a custom TLS certificate: i. To navigate to the location of the custom TLS certificate, click within the field. ii. Select the certificate file. iii. Click **Open**. iv. To upload the custom TLS certificate selected, click **Upload**. **Note:** AppViewX supports only password-protected Custom TLS Certificates.
Use proxy	**Tip:** The section on the right of the screen gives you a brief context of what is **Proxy based** A proxy server is required if the AppViewX Cloud Connector is unable to connect to your endpoints To use a proxy server for the deployment: i. Select the **Use proxy** checkbox. ii. To select a preconfigured proxy (for the selected data center), from the **Select Proxy** dropdown **OR** To create a new proxy server setting: i. Use the Click here option shown below the **Select Proxy** dropdown list. The **Add Proxy** pop-up screen is displayed.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Field	Description
	ii. Enter/Select the details required to add a proxy. **Field descriptions for the Add Proxy details** **Field** |
	***Proxy Name** | Na
	***Server IP** | IP
	***Port** | Po
	URL | Fro
	Authentication | To
	***Username** |  Ent
	***Password** |  Ent

b. Click **Finish**.

A confirmation message is displayed. AppViewX begins preparing the installer and the license file. Once the license file is ready, you can download it and proceed with the installation of the AppViewX Cloud Connector.

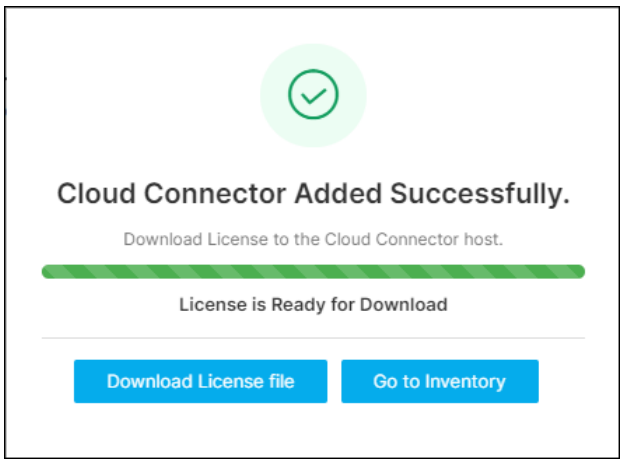


Copyright © 2025 AppViewX Inc. All Rights Reserved.

36. Download the license file.

> **Note:** The installer is prepackaged with the OVA, so, for a virtual image-based installation, you only need to download the license file.

a. On the **Cloud Connector Added Successfully** dialog box, when the **License is Ready for Download**, click **Download License file**.



> **Tip:** At this point, if the installer has been deleted or is not usable, and you wish to revert to a native installation, click **Go to Inventory**. It will take you back to the cloud connector inventory, from where you can download the license file and installer for the native OS download.

> **Tip:** You can also choose to download the license file and the installer package individually. To do this:
>
> i. Click the **Cloud Connector Name**.
>
> The selected Cloud Connector's details are shown in a pane to your right.
>
> ii. To download the AppViewX Cloud Connector installer package, click **Download Cloud Connector**. This is useful in the event that the installer has been deleted or is no longer usable.
>
> To download the license file, click **Download License**.

> **Note:** A installer download is made available even for a virtual-image based deployment, to help you with reconfiguration in case the existing OVA configuration is deleted.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

b. Save the license file on the OVA node.

On the **Settings :: Cloud Connector** page, details of this AppViewX Cloud Connector are added in the inventory table, which is explained here.

37. Install the AppViewX Cloud Connector Agent.

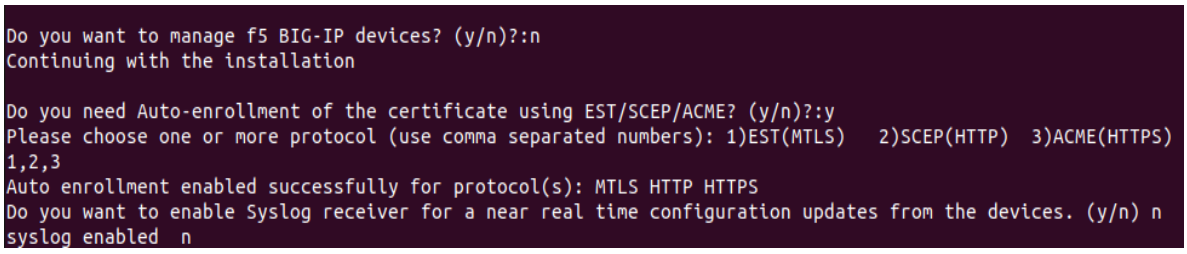
> **Note:** The following steps assume that:
> • All system prerequisites are fulfilled by the host machine.
> • The AppViewX Cloud Connector installer (downloaded in the above step) is securely copied via SCP/SFTP to the host machine where the AppViewX Cloud Connector is to be installed.

a. 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 check if the installation prerequisites for the AppViewX Cloud Connector have been fulfilled.

> **Note:** Ensure that the license file is placed in the same location as the **install.sh** script. If the license file is placed in another location, run the install.sh script using the following command:
>
> ```
> ./install.sh <complete path of the license file with the filename>
> ```

On successful verification of the prerequisites, you will be prompted to specify if you want to manage f5 BIG-IP devices and if you need auto-enrollment of the certificates.



b. When prompted, enter the required input value(s):

> **Important:** If you choose to **not enable** any of the following features, to enable them later, you will have to reinstall the AppViewX Cloud Connector.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

i. If you want manage f5 BIG-IP devices, enter **y**/**n** for yes/no, respectively.

ii. When prompted to enable auto-enrollment of the certificate using one of the following supported auto-enrollment protocols, enter **y** only if the cloud connector is being installed in a demilitarized zone (DMZ) or devices in a restricted environment (that disables them from connecting to the **<tenant>-aep** directly.

• If you choose **y** (yes) here, enter the required protocol(s) name.

> **Note:** By default, the AppViewX certificate is enabled for auto-enrollment. To enable custom certificate for auto-enrollment:
>
> 1. Execute the command `./avxctl upgrade gateway-cert`.
> 2. When prompted, enter the location of the custom certificate.

> **Note:** If you are a KUBE+ customer, the auto-enrollment gateway should be enabled as part of the installation for your KUBE+ use cases to work via the cloud connector.

iii. If you want to enable Syslog receiver for a near-real time configuration updates from the devices, enter **y/n** for yes/no, respectively. For **configuring Syslog reception**, refer to the Platform User guide section, Syslog Reception.

In case you have an older version of AppViewX on cloud and want to make use of Syslog capabilities for ADC, you must manually activate the Syslog flag by setting **SYSLOG_ENABLED**=**true** in the path **ccpath/deps/properties**.

c. Enter the sudo password.

After the relevant details have been entered, the installation proceeds. Installation logs, according to the outcome of the installation, are displayed.

Given below are sample installation logs:

```
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
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
[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)...
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
[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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

 . 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.

38. To approve the cloud connector installation:

    a. Go to  (**Menu**) > **Platform** > **Connectivity** > **Cloud Connector**.
    The **Settings :: Cloud Connector** inventory page is displayed.

    b. For the cloud connector instance installed, from the **Actions** column, click **Approve**.

> **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 using a Virtual Image on AWS

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

Prerequisites

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

  When the AppViewX AMI is successfully shared with your customer account, AppViewX will notify you of this development via email.
- From your AppViewX Onboarding Engineer, get the default password for the **appviewx** user.

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

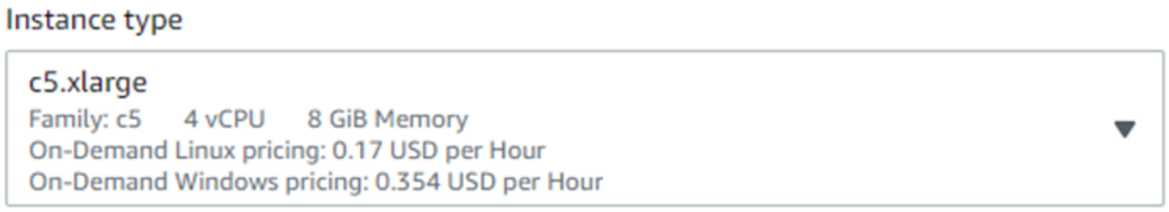
Copyright © 2025 AppViewX Inc. All Rights Reserved.

To create an EC2 instance using the AppViewX AMI:

1. Login to the AWS Management Console and go to **EC2** > **Images** > **AMIs**.
   The **Amazon Machine Images (AMIs)** page is displayed.
2. On the **Amazon Machine Images (AMIs)** page, from the **Owned by me** dropdown list, select **Private images**.
   All AMIs with visibility set to private are listed. This list will also have the AMI that is created and shared by AppViewX for your requirements.
3. From this list, select the checkbox for the AMI shared by AppViewX.
4. Click **Launch instance from AMI**.
   The **EC2 > Instances > Launch an instance** page is displayed.
5. Enter the **Name and tags** to be associated with this EC2 instance.

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

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



7. To securely connect to the EC2 instance, in the **Key pair (login)** section:
   a. To use an existing key pair, from the **Key pair name** dropdown list, select the key pair you want to use.

   OR

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

13. From the list of instances, select the AWS instance just created.
14. To login to this AWS instance using the key pair .pem file:

    a. Execute one of the following commands:
    - For Ubuntu:

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

    - For Amazon Linux 2:

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

    b. To switch to the **sudo** user, execute the following command: `sudo -i`
    After the execution of this step, if the cloud connector does not exist on this host machine, the following prompt is displayed: **Would you like to opt for automated installation of the cloud connector? (y/n):**.

    c. Since these instructions are for setting up the cloud connector via the user interface, enter **n**.

    d. To add an entry for the nameserver in the **resolv.conf** file, execute the following command: `echo "nameserver <IP of nameserver>" > /etc/resolv.conf`

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

    ```
    vi /etc/hosts


    hostnamectl set-hostname “hostname-of-the-vm”
    ```

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

    ```
    hostname -i

    hostname -f

    hostname
    ```

    g. To switch to the **appviewx** user, execute the following command: `sudo su – appviewx`

    h. {Optional, required only for password authentication) In order to successfully execute the installation, AppViewX needs to run a script for which authentication via the **.pem** file needs to be bypassed. To do this, execute the following commands:

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

    sudo systemctl restart sshd
    ```

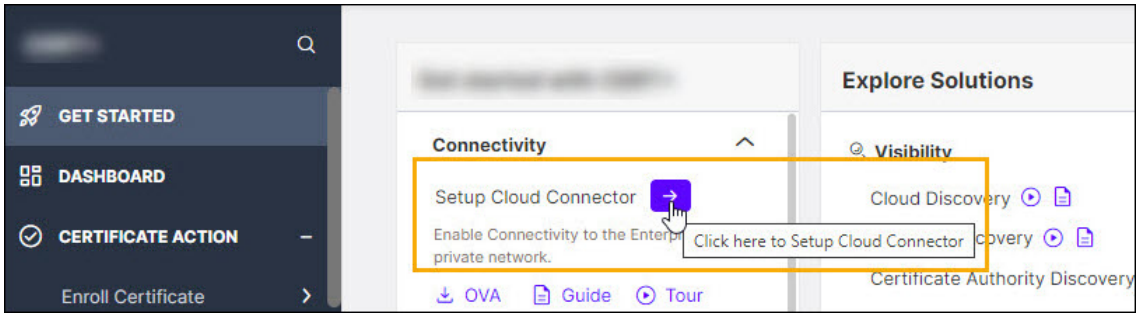
Copyright © 2025 AppViewX Inc. All Rights Reserved.

After the execution of this step, a script is executed to validate if the cloud connector exists on this host machine and the following prompt is displayed: **Would you like to opt for automated installation of the cloud connector? (y/n):**.

15. Since these instructions are for setting up the cloud connector via the user interface, enter **n**.

```
Authorized uses only. All activity may be monitored and reported.
appviewx@192.168.31.143's password:
You have new mail.
Last login: Fri Apr 12 04:37:30 2024 from 192.168.236.254
 [x] - Cloud Connector does not exist on the server. Initiating the cloud connector setup on the server
 [x] - Please make sure you have access to the tenant ID and master key for proceeding with the cloud connector setup
Would you like to opt for automated installation of the cloud connector? (y/n): n
 [x] - Skipping Automated Cloud Connector installation. Refer Cloud Connector User Guide to set it up manually.
```

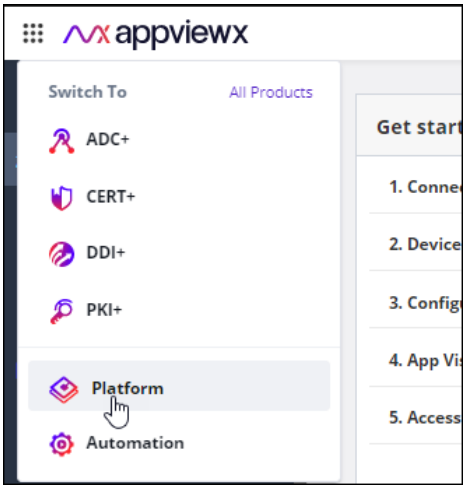
16. Enter your account URL (for example, https://tenant-name.appvx.com/appviewx/login) in the address bar of your browser.
    The AppViewX login page is displayed.
17. Login to AppViewX.
18. Navigate to the cloud connector's setup interface.
    There are three ways you can access the interface for setting up the AppViewX Cloud Connector:
    - From the product landing page (that you will see as soon as you have logged in)
      - Expand the **Connectivity** section and click  .



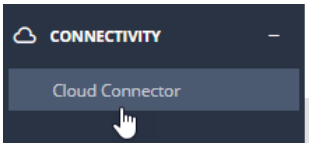
      You will be redirected to the **Settings :: Cloud Connector** page.
    - From the new navigation menu (displayed by default starting product version 2022.1.0 FP3 onwards):

Copyright © 2025 AppViewX Inc. All Rights Reserved.

a. From the menu in the top-right corner of the page, select **Platform**.



b. From the **Platform** menu, under **Connectivity**, click **Cloud Connector**.



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

• From the old navigation menu:

> **Note:** For instructions on switching between the new and the old navigation menus, click here.

a. From the top right corner of the landing page, click the menu icon.
b. From the menu displayed, navigate to **Settings** > **Cloud Connector**.

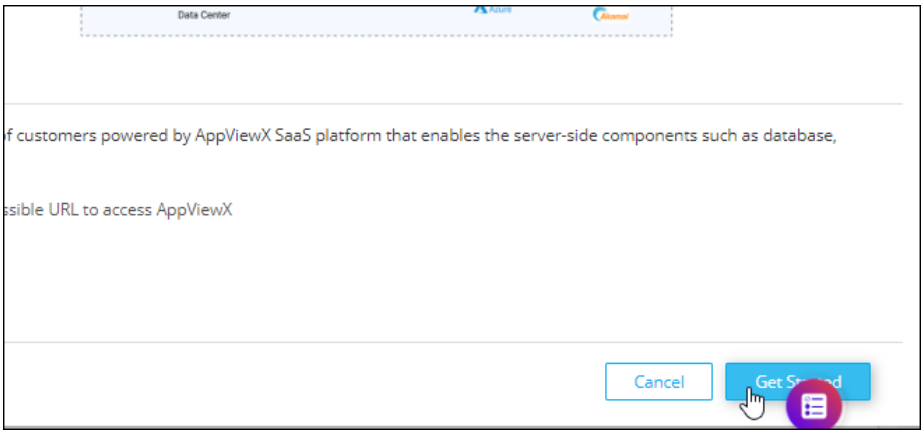
The **Settings :: Cloud Connector** page is displayed.

19. On the **Setting :: Cloud Connector** page, click **Add Cloud Connector**.
    The **Cloud Connector Setup** screen is displayed.

    The landing page gives you a quick introduction to the AppViewX Cloud Connector, with a graphical representation of how the infrastructure is deployed and works.

20. To start with the process of adding the cloud connector, from the bottom-right corner of the screen, click **Get Started**.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



You will be redirected to the **Basic Information** page.

21. On the **Basic Information** page, configure the basic cloud connector settings.

    a. To install the cloud connector via the virtual image, from **Installation Type**, select **Virtual Image**.

    > **Note:** Click here to read how a virtual image-based installation is different from a native OS installation.

    b. In the **Cloud Connector Name (FQDN)** field, enter the hostname of the machine on which the AppViewX Cloud Connector will be installed.

    > **Tip:** To retrieve the hostname, from the command line terminal of the host machine, execute the following command: `hostname -f`:

    > **Note:** The hostname entered here is added to the license file that will be generated and downloaded as part of the installer. Therefore, the license file can be used to install the cloud connector only on the machine with the entered hostname and no other.

    > **Tip:** The **Setup Cloud Connector** section to the right of the **Basic Information** screen lists hyperlinks to the prerequisites required for setting up the AppViewX Cloud Connector. To read more about what the AppViewX Cloud Connector offers, click **Learn More**.

    c. Click **Next**.

22. [Optional] Execute a prerequisite check script.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Note: This is an **optional** step. The prerequisite check script is executed automatically at the time of installing the AppViewX Cloud Connector and the results are shown as a part of the installation logs.

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.

a. On the **Basic Information** screen, under **Setup Cloud Connector**, you will see a list of the installation prerequisites.

   From this list, for **Executing the Prerequisites Check Script**, to download the script, click . The **pre-requisite-check.sh** script file is downloaded.

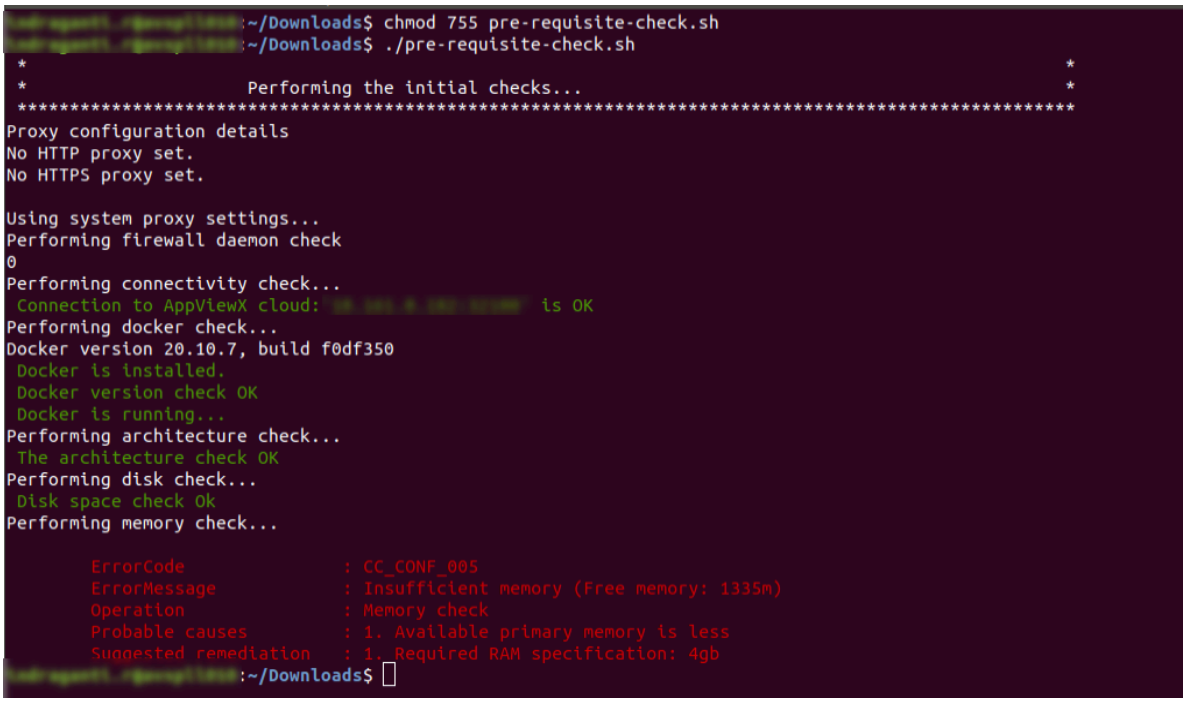
b. Securely copy the **pre-requisite-check.sh** via SCP/SFTP to the host machine where the AppViewX Cloud Connector is to be installed

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

d. Execute the **.sh** prerequisite check script file: `./pre-requisite-check.sh`
   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.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



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

23. Click **Next**.
    You will be navigated to the **AssignData Center** screen, where, for deploying the AppViewX Cloud Connector, you can either select an existing data center or add a new one.
24. To use an existing data center, select one from the options displayed on the **Assign Data Center** screen.

> **Tip:** Alternatively, you can use the **Search…** field on this screen to search for an existing data center.

To add a new data center:

a. Click **Add Data Center.**

b. In the **Add Data Center** dialog box, enter a name for the new data center.

c. Click **Save**.
The new data center will now be displayed on the **Assign Data Center** screen along with the other existing data centers.

d. Select the required data center.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

> **Tip:** The **Data Center based routing** section to the right of the **Assign Data Center** screen explains the concept of data center-based routing and how you can achieve high availability. To read more on this, click **Learn More** from the top-right corner of this screen.

25. Click **Next**.
    The **Advanced Configuration** screen is displayed.
26. On the **Advanced Configuration** page, to configure the TLS authentication and proxy server settings for your cloud connector:

    a. Enter/Select the advanced configuration settings for the AppViewX Cloud Connector.

> **Note:** The **Data center** field is auto-populated based on your selection on the **Assigning a Data Center** screen.

Field	Description
TLS Authentication	**Tip:** The section on the right of the screen gives you a brief context of what is **TLS Authent** • To auto-generate a TLS certificate, select **Auto-generate** (default selection). By default, 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** c • To enter details of a custom TLS certificate, select **Custom**. The **TLS Certificate Password** and **Custom TLS Certificate** fields are displayed. The instructio
TLS Certificate Password*	**Note:** This field is displayed only if you have selected to enter details of a Custom TLS certi 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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

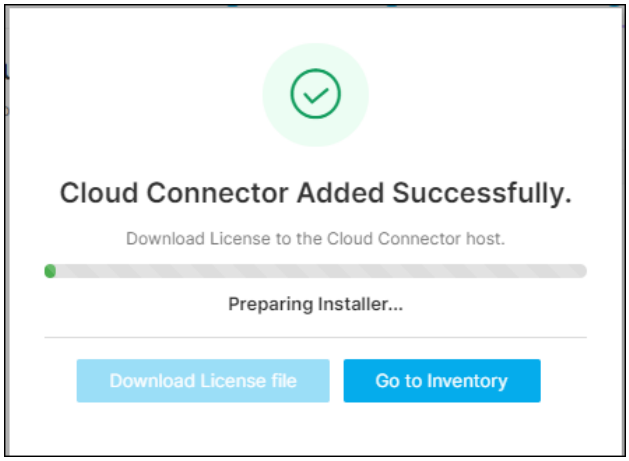
Field	Description
TLS Certificate	**Note:** This field is displayed only if you have selected to enter details of a Custom TLS certi To upload a custom TLS certificate: i. To navigate to the location of the custom TLS certificate, click within the field. ii. Select the certificate file. iii. Click **Open**. iv. To upload the custom TLS certificate selected, click **Upload**. **Note:** AppViewX supports only password-protected Custom TLS Certificates.
Use proxy	**Tip:** The section on the right of the screen gives you a brief context of what is **Proxy based** A proxy server is required if the AppViewX Cloud Connector is unable to connect to your endpoints To use a proxy server for the deployment: i. Select the **Use proxy** checkbox. ii. To select a preconfigured proxy (for the selected data center), from the **Select Proxy** dropdown **OR** To create a new proxy server setting: i. Use the Click here option shown below the **Select Proxy** dropdown list. The **Add Proxy** pop-up screen is displayed. ii. Enter/Select the details required to add a proxy. **Field descriptions for the Add Proxy details** **Field** | ***Proxy Name** | Na ***Server IP** | IP

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Field	Description
	Field
	***Port**
	URL
	Authentication
	***Username**
	***Password**

b. Click **Finish**.

A confirmation message is displayed. AppViewX begins preparing the installer and the license file. Once the license file is ready, you can download it and proceed with the installation of the AppViewX Cloud Connector.

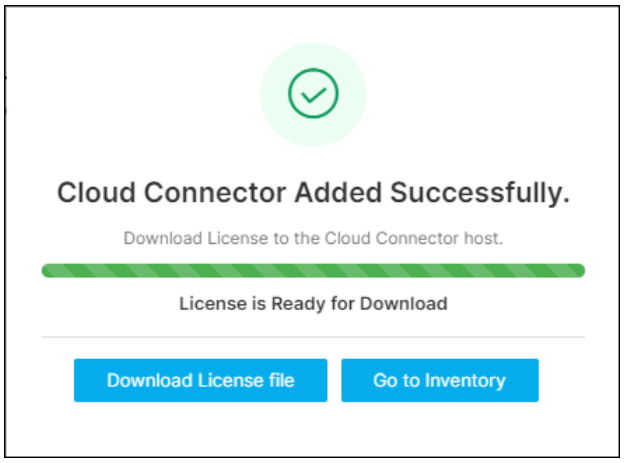


27. Download the license file.

Note: The installer is prepackaged with the OVA, so, for a virtual image-based installation, you only need to download the license file.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

a. On the **Cloud Connector Added Successfully** dialog box, when the **License is Ready for Download**, click **Download License file**.



Tip: At this point, if the installer has been deleted or is not usable, and you wish to revert to a native installation, click **Go to Inventory**. It will take you back to the cloud connector inventory, from where you can download the license file and installer for the native OS download.

Tip: You can also choose to download the license file and the installer package individually. To do this:

i. Click the **Cloud Connector Name**.

The selected Cloud Connector's details are shown in a pane to your right.

ii. To download the AppViewX Cloud Connector installer package, click **Download Cloud Connector**. This is useful in the event that the installer has been deleted or is no longer usable.

To download the license file, click **Download License**.

Note: A installer download is made available even for a virtual-image based deployment, to help you with reconfiguration in case the existing OVA configuration is deleted.

b. Save the license file on the OVA node.

On the **Settings :: Cloud Connector** page, details of this AppViewX Cloud Connector are added in the inventory table, which is explained here.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

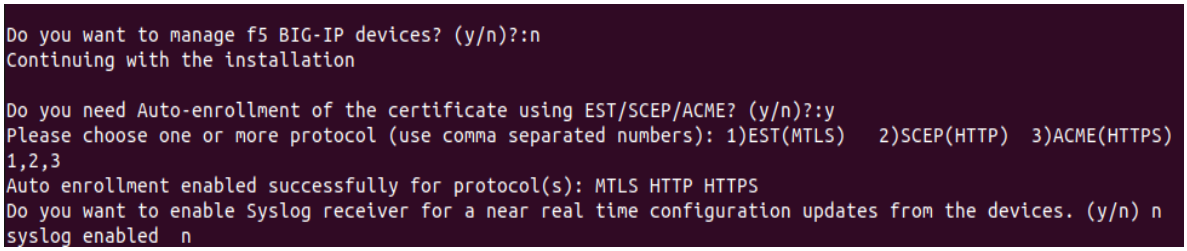
28. Install the AppViewX Cloud Connector Agent.

> **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.

a. 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 check if the installation prerequisites for the AppViewX Cloud Connector have been fulfilled.

> **Note:** Ensure that the license file is placed in the same location as the **install.sh** script.
> If the license file is placed in another location, run the install.sh script using the following command:
>
> ```
> ./install.sh <complete path of the license file with the filename>
> ```

On successful verification of the prerequisites, you will be prompted to specify if you want to manage f5 BIG-IP devices and if you need auto-enrollment of the certificates.



b. When prompted, enter the required input value(s):

> **Important:** If you choose to **not enable** any of the following features, to enable them later, you will have to reinstall the AppViewX Cloud Connector.

i. If you want manage f5 BIG-IP devices, enter **y**/**n** for yes/no, respectively.
ii. When prompted to enable auto-enrollment of the certificate using one of the following supported auto-enrollment protocols, enter **y** only if the cloud connector is being installed in a demilitarized zone (DMZ) or devices in a restricted environment (that disables them from connecting to the **<tenant>-aep** directly.
• If you choose **y** (yes) here, enter the required protocol(s) name.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

> **Note:** By default, the AppViewX certificate is enabled for auto-enrollment. To enable custom certificate for auto-enrollment:
>
> 1. Execute the command `./avxctl upgrade gateway-cert`.
> 2. When prompted, enter the location of the custom certificate.

> **Note:** If you are a KUBE+ customer, the auto-enrollment gateway should be enabled as part of the installation for your KUBE+ use cases to work via the cloud connector.

iii. If you want to enable Syslog receiver for a near-real time configuration updates from the devices, enter **y/n** for yes/no, respectively. For **configuring Syslog reception**, refer to the Platform User guide section, Syslog Reception.

In case you have an older version of AppViewX on cloud and want to make use of Syslog capabilities for ADC, you must manually activate the Syslog flag by setting **SYSLOG_ENABLED**=**true** in the path **ccpath/deps/properties**.

c. Enter the sudo password.

After the relevant details have been entered, the installation proceeds. Installation logs, according to the outcome of the installation, are displayed.

Given below are sample installation logs:

```
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
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
[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)...
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
[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.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

29. To approve the cloud connector installation:

    a. Go to (**Menu**) > **Platform** > **Connectivity** > **Cloud Connector**.
    The **Settings :: Cloud Connector** inventory page is displayed.

    b. For the cloud connector instance installed, from the **Actions** column, click **Approve**.

> **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 using a Virtual Image on Azure

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

To deploy the AppViewX virtual machine for Azure:

1. Go to https://release.appviewx.com/Login and, from **Overview**, navigate to release Example: **20XX.0.0**.
2. Scroll down to Production Images and download the latest artifact of Azure CC VHD, **AppViewX-20XX.X.X-FPX-CC-Ubuntu-Azure-ddmmmyyy-vhd.tar.gz**.
3. Scroll down to Production Images and download the latest artifact of Azure CC VHD, **AppViewX-20XX.X.X.10-CC-Ubuntu-Azure-ddmmmyyy-vhd.tar.gz**.
4. Untar the downloaded artifact.
   ```
   tar -xvf AppViewX-20XX.X.X.10-CC-Ubuntu-Azure-ddmmmyyy-vhd.tar.gz
   ```
5. Download the **Azure Storage Explorer** from here.
   The **Azure Storage Explorer** is a desktop application that provides you with a GUI for easily managing your Azure resources.

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

6. Using the Azure Storage Explorer, login to the Azure account for which the VM has to be created.
7. On successful login, go to the **disks** section and select the resource group.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

The resource group page is displayed.

8. Click **Upload**.
9. In the pop-up window displayed, enter/select the resource details.

Descriptions for the resources and their corresponding values

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

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

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

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

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

```
vi /etc/hosts

hostnamectl set-hostname “hostname-of-the-vm”
```

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
hostname -i

hostname -f

hostname
```

After the execution of this step, if the cloud connector does not exist on this host machine, the following prompt is displayed: **Would you like to opt for automated installation of the cloud connector? (y/n):**.

16. Since these instructions are for setting up the cloud connector via the user interface, enter **n**.

```
Authorized uses only. All activity may be monitored and reported.
appviewx@192.168.31.143's password:
You have new mail.
Last login: Fri Apr 12 04:37:30 2024 from 192.168.236.254
 [x] - Cloud Connector does not exist on the server. Initiating the cloud connector setup on the server
 [x] - Please make sure you have access to the tenant ID and master key for proceeding with the cloud connector setup
Would you like to opt for automated installation of the cloud connector? (y/n): n
 [x] - Skipping Automated Cloud Connector installation. Refer Cloud Connector User Guide to set it up manually.
```

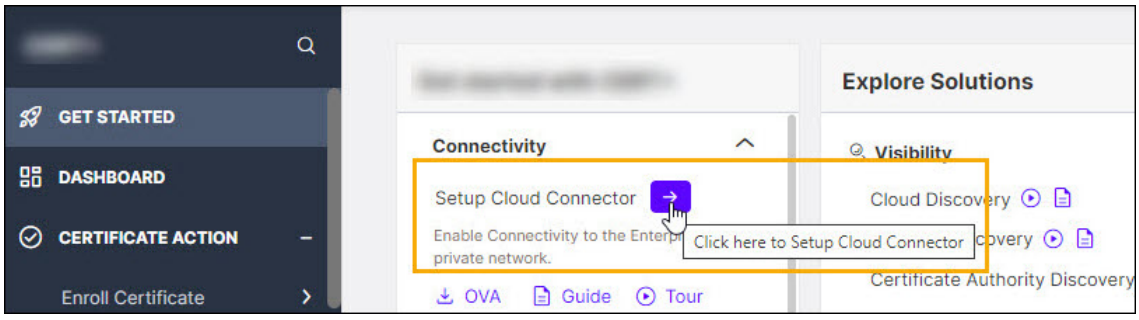
17. Enter your account URL (for example, https://tenant-name.appvx.com/appviewx/login) in the address bar of your browser.

    The AppViewX login page is displayed.

18. Login to AppViewX.
19. Navigate to the cloud connector's setup interface.

    There are three ways you can access the interface for setting up the AppViewX Cloud Connector:

    - From the product landing page (that you will see as soon as you have logged in)
      - Expand the **Connectivity** section and click  .

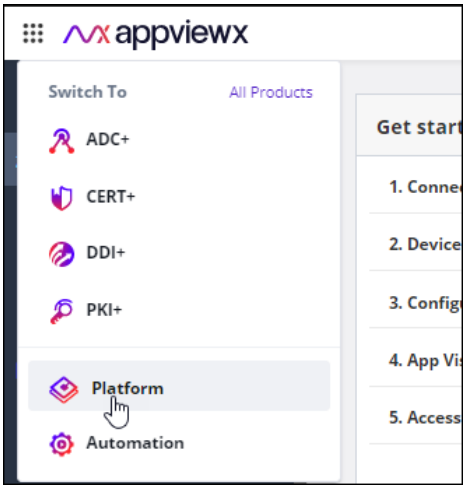


      You will be redirected to the **Settings :: Cloud Connector** page.

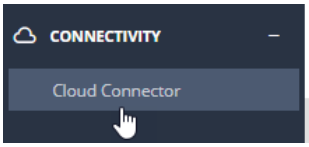
    - From the new navigation menu (displayed by default starting product version 2022.1.0 FP3 onwards):

Copyright © 2025 AppViewX Inc. All Rights Reserved.

a. From the menu in the top-right corner of the page, select **Platform**.



b. From the **Platform** menu, under **Connectivity**, click **Cloud Connector**.



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

• From the old navigation menu:

Note: For instructions on switching between the new and the old navigation menus, click here.

a. From the top right corner of the landing page, click the menu icon.

b. From the menu displayed, navigate to **Settings** > **Cloud Connector**.

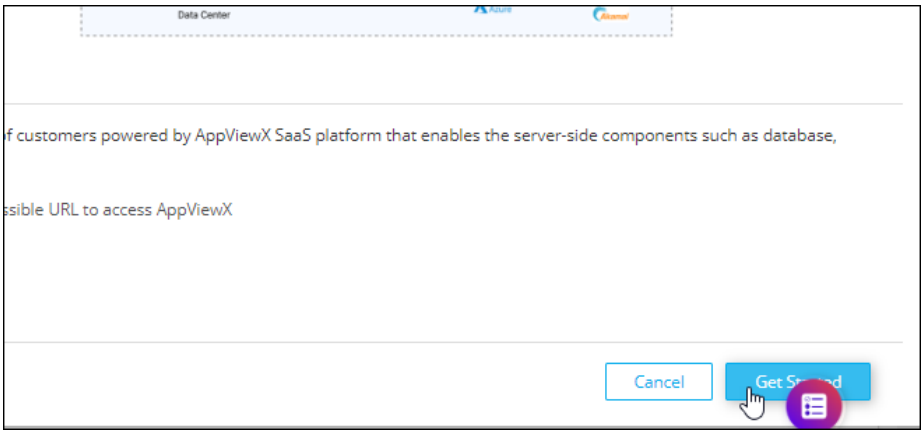
The **Settings :: Cloud Connector** page is displayed.

20. On the **Setting :: Cloud Connector** page, click **Add Cloud Connector**.
    The **Cloud Connector Setup** screen is displayed.

    The landing page gives you a quick introduction to the AppViewX Cloud Connector, with a graphical representation of how the infrastructure is deployed and works.

21. To start with the process of adding the cloud connector, from the bottom-right corner of the screen, click **Get Started**.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



You will be redirected to the **Basic Information** page.

22. On the **Basic Information** page, configure the basic cloud connector settings.

    a. To install the cloud connector via the virtual image, from **Installation Type**, select **Virtual Image**.

    > **Note:** Click here to read how a virtual image-based installation is different from a native OS installation.

    b. In the **Cloud Connector Name (FQDN)** field, enter the hostname of the machine on which the AppViewX Cloud Connector will be installed.

    > **Tip:** To retrieve the hostname, from the command line terminal of the host machine, execute the following command: `hostname -f`:

    > **Note:** The hostname entered here is added to the license file that will be generated and downloaded as part of the installer. Therefore, the license file can be used to install the cloud connector only on the machine with the entered hostname and no other.

    > **Tip:** The **Setup Cloud Connector** section to the right of the **Basic Information** screen lists hyperlinks to the prerequisites required for setting up the AppViewX Cloud Connector. To read more about what the AppViewX Cloud Connector offers, click **Learn More**.

    c. Click **Next**.

23. [Optional] Execute a prerequisite check script.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Note: This is an **optional** step. The prerequisite check script is executed automatically at the time of installing the AppViewX Cloud Connector and the results are shown as a part of the installation logs.

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.

a. On the **Basic Information** screen, under **Setup Cloud Connector**, you will see a list of the installation prerequisites.

   From this list, for **Executing the Prerequisites Check Script**, to download the script, click . The **pre-requisite-check.sh** script file is downloaded.

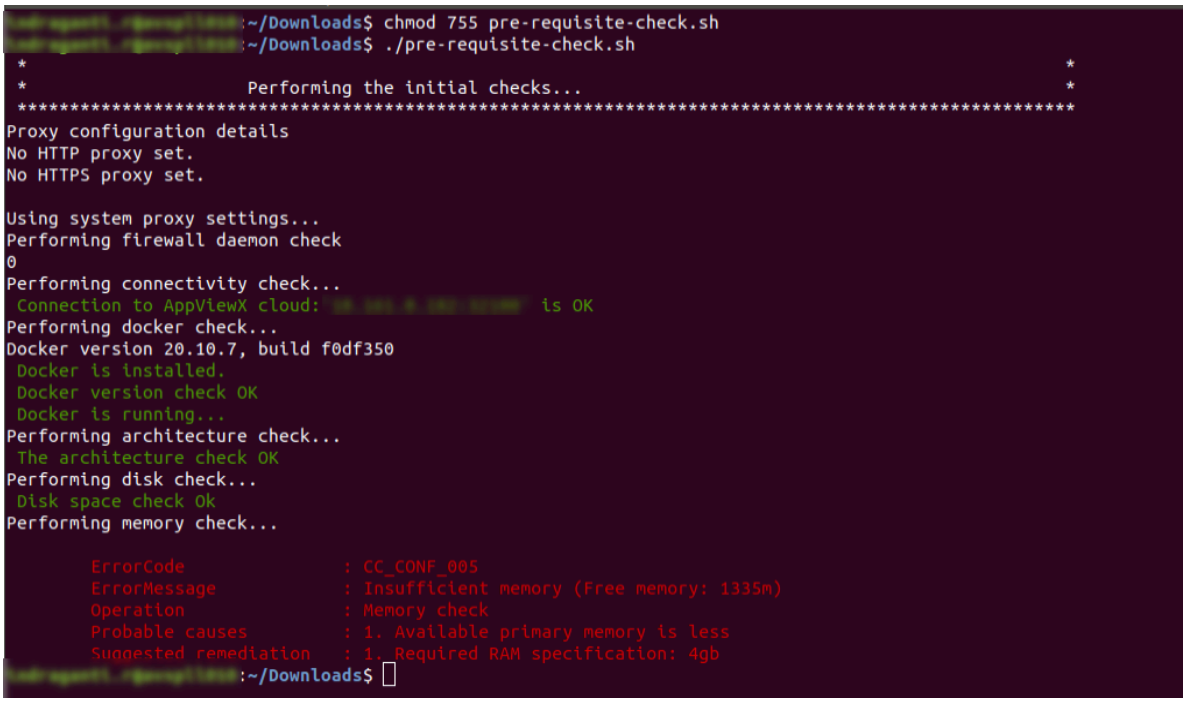
b. Securely copy the **pre-requisite-check.sh** via SCP/SFTP to the host machine where the AppViewX Cloud Connector is to be installed

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

d. Execute the **.sh** prerequisite check script file: `./pre-requisite-check.sh`
   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.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



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

24. Click **Next**.
    You will be navigated to the **AssignData Center** screen, where, for deploying the AppViewX Cloud Connector, you can either select an existing data center or add a new one.
25. To use an existing data center, select one from the options displayed on the **Assign Data Center** screen.

> **Tip:** Alternatively, you can use the **Search…** field on this screen to search for an existing data center.

To add a new data center:

a. Click **Add Data Center.**

b. In the **Add Data Center** dialog box, enter a name for the new data center.

c. Click **Save**.
   The new data center will now be displayed on the **Assign Data Center** screen along with the other existing data centers.

d. Select the required data center.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

> **Tip:** The **Data Center based routing** section to the right of the **Assign Data Center** screen explains the concept of data center-based routing and how you can achieve high availability. To read more on this, click **Learn More** from the top-right corner of this screen.

26. Click **Next**.
    The **Advanced Configuration** screen is displayed.
27. On the **Advanced Configuration** page, to configure the TLS authentication and proxy server settings for your cloud connector:

    a. Enter/Select the advanced configuration settings for the AppViewX Cloud Connector.

> **Note:** The **Data center** field is auto-populated based on your selection on the **Assigning a Data Center** screen.

Field	Description
TLS Authentication	**Tip:** The section on the right of the screen gives you a brief context of what is **TLS Authent** • To auto-generate a TLS certificate, select **Auto-generate** (default selection). By default, 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** c • To enter details of a custom TLS certificate, select **Custom**. The **TLS Certificate Password** and **Custom TLS Certificate** fields are displayed. The instructio
TLS Certificate Password*	**Note:** This field is displayed only if you have selected to enter details of a Custom TLS certi 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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Field	Description
TLS Certificate	**Note:** This field is displayed only if you have selected to enter details of a Custom TLS certi To upload a custom TLS certificate: i. To navigate to the location of the custom TLS certificate, click within the field. ii. Select the certificate file. iii. Click **Open**. iv. To upload the custom TLS certificate selected, click **Upload**. **Note:** AppViewX supports only password-protected Custom TLS Certificates.
Use proxy	**Tip:** The section on the right of the screen gives you a brief context of what is **Proxy based** A proxy server is required if the AppViewX Cloud Connector is unable to connect to your endpoints To use a proxy server for the deployment: i. Select the **Use proxy** checkbox. ii. To select a preconfigured proxy (for the selected data center), from the **Select Proxy** dropdown **OR** To create a new proxy server setting: i. Use the Click here option shown below the **Select Proxy** dropdown list. The **Add Proxy** pop-up screen is displayed. ii. Enter/Select the details required to add a proxy. **Field descriptions for the Add Proxy details**

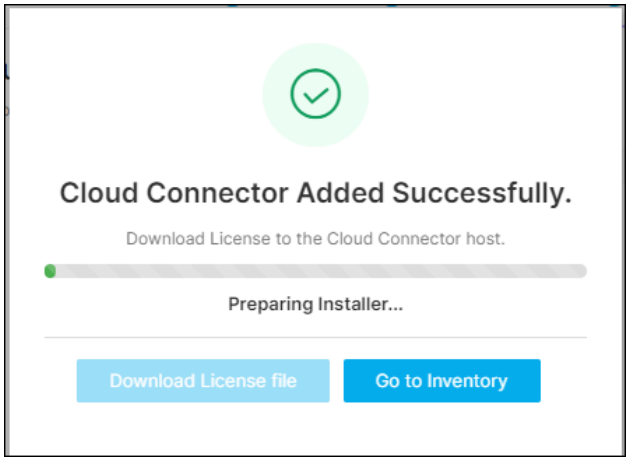
Field	
***Proxy Name**	Na
***Server IP**	IP a

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Field	Description
	Field
	***Port**
	URL
	Authentication
	***Username**
	***Password**

b. Click **Finish**.

A confirmation message is displayed. AppViewX begins preparing the installer and the license file. Once the license file is ready, you can download it and proceed with the installation of the AppViewX Cloud Connector.

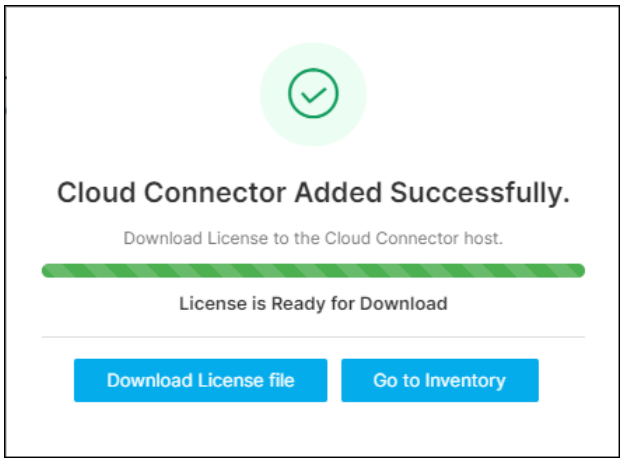


28. Download the license file.

Note: The installer is prepackaged with the OVA, so, for a virtual image-based installation, you only need to download the license file.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

a. On the **Cloud Connector Added Successfully** dialog box, when the **License is Ready for Download**, click **Download License file**.



Tip: At this point, if the installer has been deleted or is not usable, and you wish to revert to a native installation, click **Go to Inventory**. It will take you back to the cloud connector inventory, from where you can download the license file and installer for the native OS download.

Tip: You can also choose to download the license file and the installer package individually. To do this:

i. Click the **Cloud Connector Name**.

The selected Cloud Connector's details are shown in a pane to your right.

ii. To download the AppViewX Cloud Connector installer package, click **Download Cloud Connector**. This is useful in the event that the installer has been deleted or is no longer usable.

To download the license file, click **Download License**.

Note: A installer download is made available even for a virtual-image based deployment, to help you with reconfiguration in case the existing OVA configuration is deleted.

b. Save the license file on the OVA node.

On the **Settings :: Cloud Connector** page, details of this AppViewX Cloud Connector are added in the inventory table, which is explained here.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

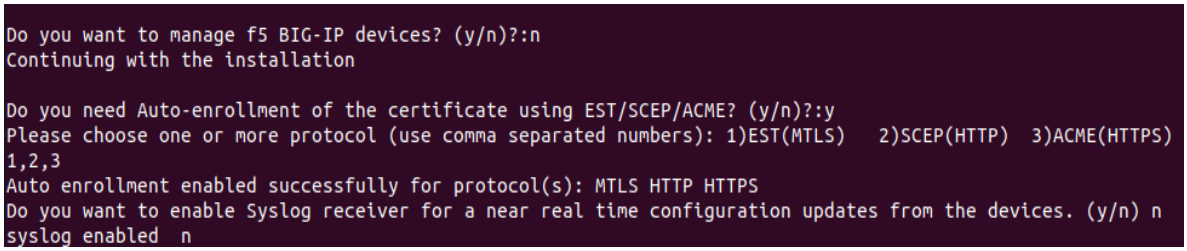
29. Install the AppViewX Cloud Connector Agent.

> **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.

a. 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 check if the installation prerequisites for the AppViewX Cloud Connector have been fulfilled.

> **Note:** Ensure that the license file is placed in the same location as the **install.sh** script.
> If the license file is placed in another location, run the install.sh script using the following command:
>
> ```
> ./install.sh <complete path of the license file with the filename>
> ```

On successful verification of the prerequisites, you will be prompted to specify if you want to manage f5 BIG-IP devices and if you need auto-enrollment of the certificates.



b. When prompted, enter the required input value(s):

> **Important:** If you choose to **not enable** any of the following features, to enable them later, you will have to reinstall the AppViewX Cloud Connector.

i. If you want manage f5 BIG-IP devices, enter **y**/**n** for yes/no, respectively.
ii. When prompted to enable auto-enrollment of the certificate using one of the following supported auto-enrollment protocols, enter **y** only if the cloud connector is being installed in a demilitarized zone (DMZ) or devices in a restricted environment (that disables them from connecting to the **<tenant>-aep** directly.
• If you choose **y** (yes) here, enter the required protocol(s) name.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

> **Note:** By default, the AppViewX certificate is enabled for auto-enrollment. To enable custom certificate for auto-enrollment:
>
> 1. Execute the command `./avxctl upgrade gateway-cert`.
> 2. When prompted, enter the location of the custom certificate.

> **Note:** If you are a KUBE+ customer, the auto-enrollment gateway should be enabled as part of the installation for your KUBE+ use cases to work via the cloud connector.

iii. If you want to enable Syslog receiver for a near-real time configuration updates from the devices, enter **y/n** for yes/no, respectively. For **configuring Syslog reception**, refer to the Platform User guide section, Syslog Reception.

In case you have an older version of AppViewX on cloud and want to make use of Syslog capabilities for ADC, you must manually activate the Syslog flag by setting **SYSLOG_ENABLED**=**true** in the path **ccpath/deps/properties**.

c. Enter the sudo password.

After the relevant details have been entered, the installation proceeds. Installation logs, according to the outcome of the installation, are displayed.

Given below are sample installation logs:

```
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
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
[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)...
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
[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.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

30. To approve the cloud connector installation:

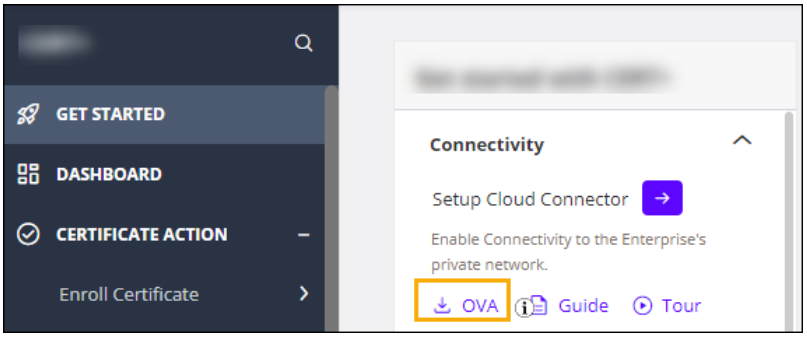
    a. Go to (**Menu**) > **Platform** > **Connectivity** > **Cloud Connector**.
    The **Settings :: Cloud Connector** inventory page is displayed.

    b. For the cloud connector instance installed, from the **Actions** column, click **Approve**.

> **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 using a Virtual Image for GCP

1. Log in to the GCP console and start a **Cloud Shell** instance.
2. From the command line terminal, create a bucket with the default settings.
   `gsutil mb gs://my-virtual-appliances-bucket`
3. Download the GCP OVA from the AppViewX release portal.
   Alternatively, you can Download the release package in the OVA format, from the respective AppViewX's product line landing page, under **GET STARTED** menu > **Connectivity** section, click OVA.



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

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

5. Create a virtual instance from the OVA.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

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

6. Login to the GCP node using AppViewX credentials.

   `ssh appviewx@<node IP>`

   After the execution of this step, a script is executed to validate if the cloud connector exists on this host machine and the following prompt is displayed: **Would you like to opt for automated installation of the cloud connector? (y/n):**.
7. Since these instructions are for setting up the cloud connector via the user interface, enter **n**.

```
Authorized uses only. All activity may be monitored and reported.
appviewx@192.168.31.143's password:
You have new mail.
Last login: Fri Apr 12 04:37:30 2024 from 192.168.236.254
 [x] - Cloud Connector does not exist on the server. Initiating the cloud connector setup on the server
 [x] - Please make sure you have access to the tenant ID and master key for proceeding with the cloud connector setup
Would you like to opt for automated installation of the cloud connector? (y/n): n
 [x] - Skipping Automated Cloud Connector installation. Refer Cloud Connector User Guide to set it up manually.
```

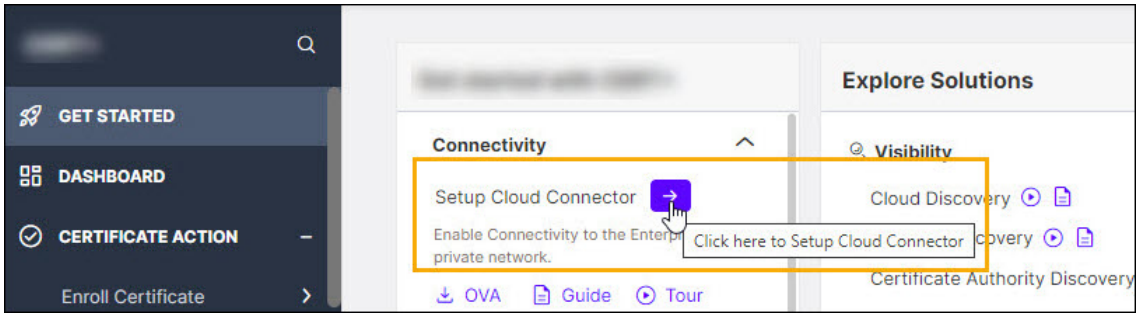
8. Enter your account URL (for example, https://tenant-name.appvx.com/appviewx/login) in the address bar of your browser.

   The AppViewX login page is displayed.
9. Login to AppViewX.
10. Navigate to the cloud connector's setup interface.

    There are three ways you can access the interface for setting up the AppViewX Cloud Connector:

    • From the product landing page (that you will see as soon as you have logged in)

      • Expand the **Connectivity** section and click  .

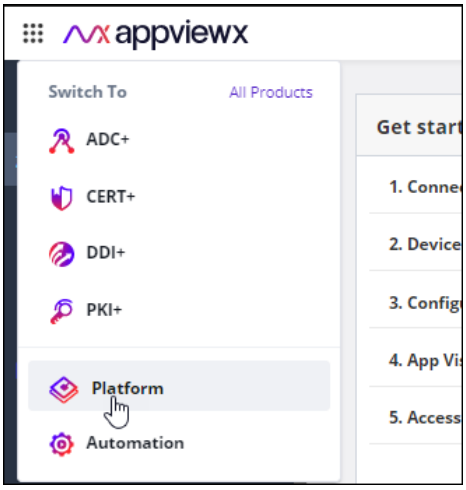


      You will be redirected to the **Settings :: Cloud Connector** page.

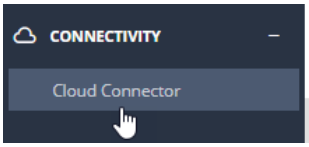
    • From the new navigation menu (displayed by default starting product version 2022.1.0 FP3 onwards):

Copyright © 2025 AppViewX Inc. All Rights Reserved.

a. From the menu in the top-right corner of the page, select **Platform**.



b. From the **Platform** menu, under **Connectivity**, click **Cloud Connector**.



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

• From the old navigation menu:

> **Note:** For instructions on switching between the new and the old navigation menus, click here.

a. From the top right corner of the landing page, click the menu icon.

b. From the menu displayed, navigate to **Settings** > **Cloud Connector**.

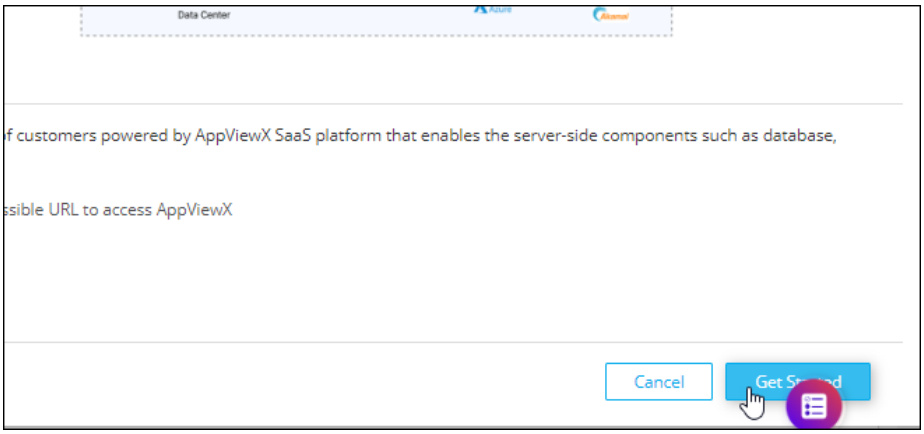
The **Settings :: Cloud Connector** page is displayed.

11. On the **Setting :: Cloud Connector** page, click **Add Cloud Connector**.
    The **Cloud Connector Setup** screen is displayed.

    The landing page gives you a quick introduction to the AppViewX Cloud Connector, with a graphical representation of how the infrastructure is deployed and works.

12. To start with the process of adding the cloud connector, from the bottom-right corner of the screen, click **Get Started**.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



You will be redirected to the **Basic Information** page.

13. On the **Basic Information** page, configure the basic cloud connector settings.

    a. To install the cloud connector via the virtual image, from **Installation Type**, select **Virtual Image**.

    > **Note:** Click here to read how a virtual image-based installation is different from a native OS installation.

    b. In the **Cloud Connector Name (FQDN)** field, enter the hostname of the machine on which the AppViewX Cloud Connector will be installed.

    > **Tip:** To retrieve the hostname, from the command line terminal of the host machine, execute the following command: `hostname -f`:

    > **Note:** The hostname entered here is added to the license file that will be generated and downloaded as part of the installer. Therefore, the license file can be used to install the cloud connector only on the machine with the entered hostname and no other.

    > **Tip:** The **Setup Cloud Connector** section to the right of the **Basic Information** screen lists hyperlinks to the prerequisites required for setting up the AppViewX Cloud Connector. To read more about what the AppViewX Cloud Connector offers, click **Learn More**.

    c. Click **Next**.

14. [Optional] Execute a prerequisite check script.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Note: This is an **optional** step. The prerequisite check script is executed automatically at the time of installing the AppViewX Cloud Connector and the results are shown as a part of the installation logs.

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.

a. On the **Basic Information** screen, under **Setup Cloud Connector**, you will see a list of the installation prerequisites.

   From this list, for **Executing the Prerequisites Check Script**, to download the script, click . The **pre-requisite-check.sh** script file is downloaded.

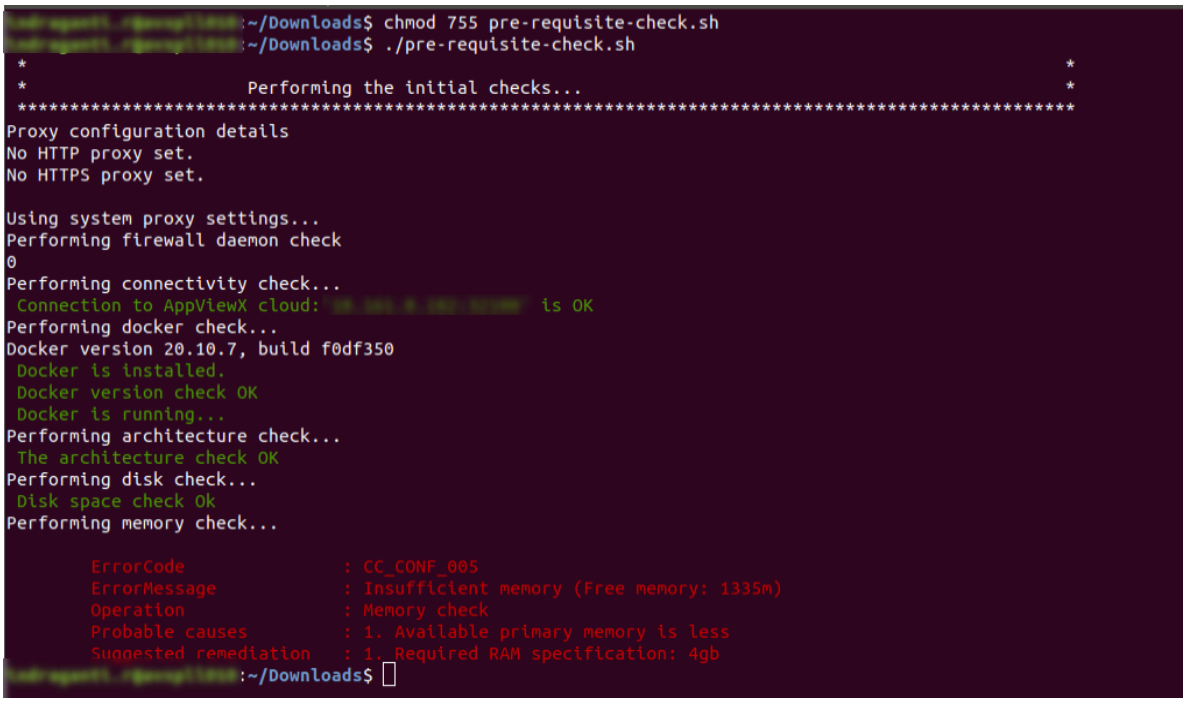
b. Securely copy the **pre-requisite-check.sh** via SCP/SFTP to the host machine where the AppViewX Cloud Connector is to be installed

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

d. Execute the **.sh** prerequisite check script file: `./pre-requisite-check.sh`
   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.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



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

15. Click **Next**.
    You will be navigated to the **AssignData Center** screen, where, for deploying the AppViewX Cloud Connector, you can either select an existing data center or add a new one.
16. To use an existing data center, select one from the options displayed on the **Assign Data Center** screen.

> **Tip:** Alternatively, you can use the **Search…** field on this screen to search for an existing data center.

To add a new data center:

a. Click **Add Data Center.**

b. In the **Add Data Center** dialog box, enter a name for the new data center.

c. Click **Save**.
The new data center will now be displayed on the **Assign Data Center** screen along with the other existing data centers.

d. Select the required data center.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

> **Tip:** The **Data Center based routing** section to the right of the **Assign Data Center** screen explains the concept of data center-based routing and how you can achieve high availability. To read more on this, click **Learn More** from the top-right corner of this screen.

17. Click **Next**.
    The **Advanced Configuration** screen is displayed.
18. On the **Advanced Configuration** page, to configure the TLS authentication and proxy server settings for your cloud connector:

    a. Enter/Select the advanced configuration settings for the AppViewX Cloud Connector.

> **Note:** The **Data center** field is auto-populated based on your selection on the **Assigning a Data Center** screen.

Field	Description
TLS Authentication	**Tip:** The section on the right of the screen gives you a brief context of what is **TLS Authent** • To auto-generate a TLS certificate, select **Auto-generate** (default selection). By default, 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** c • To enter details of a custom TLS certificate, select **Custom**. The **TLS Certificate Password** and **Custom TLS Certificate** fields are displayed. The instructio
TLS Certificate Password*	**Note:** This field is displayed only if you have selected to enter details of a Custom TLS certi Password of the TLS certificate (that will be uploaded in the next step)

Copyright © 2025 AppViewX Inc. All Rights Reserved.

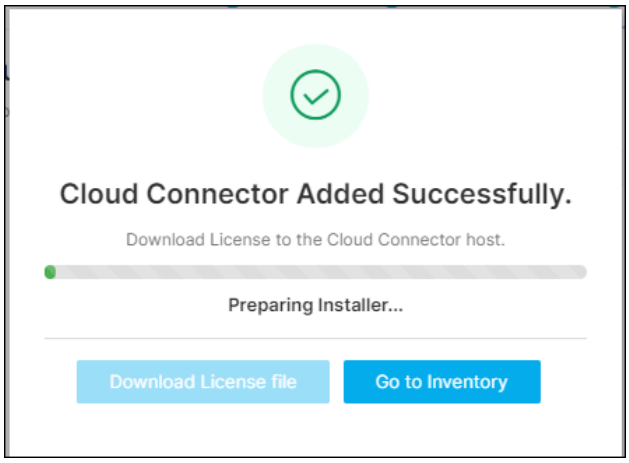
Field	Description
	Note: This is a mandatory field if a Custom TLS certificate is uploaded. AppViewX supports
TLS Certificate	**Note:** This field is displayed only if you have selected to enter details of a Custom TLS certi To upload a custom TLS certificate: i. To navigate to the location of the custom TLS certificate, click within the field. ii. Select the certificate file. iii. Click **Open**. iv. To upload the custom TLS certificate selected, click **Upload**. **Note:** AppViewX supports only password-protected Custom TLS Certificates.
Use proxy	**Tip:** The section on the right of the screen gives you a brief context of what is **Proxy based** A proxy server is required if the AppViewX Cloud Connector is unable to connect to your endpoints To use a proxy server for the deployment: i. Select the **Use proxy** checkbox. ii. To select a preconfigured proxy (for the selected data center), from the **Select Proxy** dropdown **OR** To create a new proxy server setting: i. Use the Click here option shown below the **Select Proxy** dropdown list. The **Add Proxy** pop-up screen is displayed. ii. Enter/Select the details required to add a proxy. **Field descriptions for the Add Proxy details** **Field** | ***Proxy Name** | Na

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Field	Description
	Field
	***Server IP**
	***Port**
	URL
	Authentication
	***Username**
	***Password**

b. Click **Finish**.

A confirmation message is displayed. AppViewX begins preparing the installer and the license file. Once the license file is ready, you can download it and proceed with the installation of the AppViewX Cloud Connector.

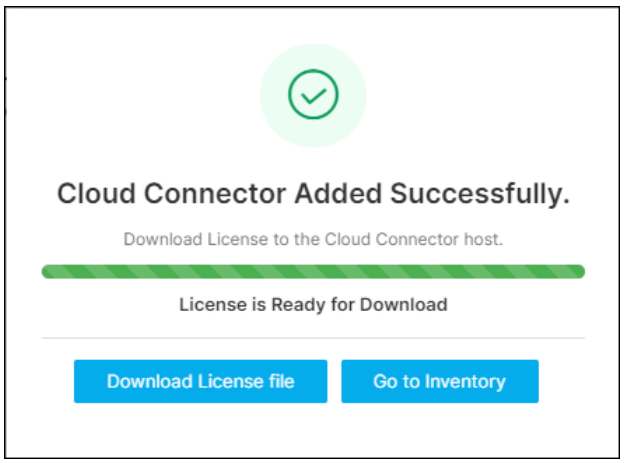


19. Download the license file.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Note: The installer is prepackaged with the OVA, so, for a virtual image-based installation, you only need to download the license file.

a. On the **Cloud Connector Added Successfully** dialog box, when the **License is Ready for Download**, click **Download License file**.



Tip: At this point, if the installer has been deleted or is not usable, and you wish to revert to a native installation, click **Go to Inventory**. It will take you back to the cloud connector inventory, from where you can download the license file and installer for the native OS download.

Tip: You can also choose to download the license file and the installer package individually. To do this:

i. Click the **Cloud Connector Name**.

The selected Cloud Connector's details are shown in a pane to your right.

ii. To download the AppViewX Cloud Connector installer package, click **Download Cloud Connector**. This is useful in the event that the installer has been deleted or is no longer usable.

To download the license file, click **Download License**.

Note: A installer download is made available even for a virtual-image based deployment, to help you with reconfiguration in case the existing OVA configuration is deleted.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

b. Save the license file on the OVA node.

On the **Settings :: Cloud Connector** page, details of this AppViewX Cloud Connector are added in the inventory table, which is explained here.

20. Install 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.

a. 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 check if the installation prerequisites for the AppViewX Cloud Connector have been fulfilled.

> **Note:** Ensure that the license file is placed in the same location as the **install.sh** script. If the license file is placed in another location, run the install.sh script using the following command:
>
> ```
> ./install.sh <complete path of the license file with the filename>
> ```

On successful verification of the prerequisites, you will be prompted to specify if you want to manage f5 BIG-IP devices and if you need auto-enrollment of the certificates.

```
Do you want to manage f5 BIG-IP devices? (y/n)?:n
Continuing with the installation

Do you need Auto-enrollment of the certificate using EST/SCEP/ACME? (y/n)?:y
Please choose one or more protocol (use comma separated numbers): 1)EST(MTLS)   2)SCEP(HTTP)  3)ACME(HTTPS)
1,2,3
Auto enrollment enabled successfully for protocol(s): MTLS HTTP HTTPS
Do you want to enable Syslog receiver for a near real time configuration updates from the devices. (y/n) n
syslog enabled  n
```

b. When prompted, enter the required input value(s):

> **Important:** If you choose to **not enable** any of the following features, to enable them later, you will have to reinstall the AppViewX Cloud Connector.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

i. If you want manage f5 BIG-IP devices, enter **y**/**n** for yes/no, respectively.

ii. When prompted to enable auto-enrollment of the certificate using one of the following supported auto-enrollment protocols, enter **y** only if the cloud connector is being installed in a demilitarized zone (DMZ) or devices in a restricted environment (that disables them from connecting to the **<tenant>-aep** directly.

• If you choose **y** (yes) here, enter the required protocol(s) name.

> **Note:** By default, the AppViewX certificate is enabled for auto-enrollment. To enable custom certificate for auto-enrollment:
>
> 1. Execute the command `./avxctl upgrade gateway-cert`.
> 2. When prompted, enter the location of the custom certificate.

> **Note:** If you are a KUBE+ customer, the auto-enrollment gateway should be enabled as part of the installation for your KUBE+ use cases to work via the cloud connector.

iii. If you want to enable Syslog receiver for a near-real time configuration updates from the devices, enter **y/n** for yes/no, respectively. For **configuring Syslog reception**, refer to the Platform User guide section, Syslog Reception.

In case you have an older version of AppViewX on cloud and want to make use of Syslog capabilities for ADC, you must manually activate the Syslog flag by setting **SYSLOG_ENABLED**=**true** in the path **ccpath/deps/properties**.

c. Enter the sudo password.

After the relevant details have been entered, the installation proceeds. Installation logs, according to the outcome of the installation, are displayed.

Given below are sample installation logs:

```
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
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
[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)...
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
[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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

 . 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.

21. To approve the cloud connector installation:

    a. Go to  (**Menu**) > **Platform** > **Connectivity** > **Cloud Connector**.
    The **Settings :: Cloud Connector** inventory page is displayed.

    b. For the cloud connector instance installed, from the **Actions** column, click **Approve**.

> **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 by Quick Installation using CLI

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

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

To install the AppViewX Cloud Connector using the automated installation script, from the cloud connector details banner, you can either download the installer script or copy the `curl` command.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

1. Configure the native OS for the cloud connector installation.

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

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

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

b. Uninstall the previous version of Docker.

- **For RHEL, CentOS, and Amazon Linux 2**:

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

- **For Ubuntu**

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

c. Configure the hostname and the nameserver.

i. Configure the hostname: `sudo hostnamectl set-hostname "hostname"`

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

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

iii. Replace the default nameserver IP address.

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

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

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

d. Add Docker repo and install Docker.

- For **RHEL** and **CentOS**

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

- For **Amazon Linux 2**

```
sudo yum install -y docker containerd
sudo systemctl start docker
sudo systemctl enable docker
sudo systemctl status docker
```

- For **Ubuntu**

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

To install Docker for Ubuntu

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

e. Add the appviewx user in the Docker group.

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

f. Install NTP.

- For **RHEL** and **CentOS**

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

- For **Amazon Linux 2**

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

- For **Ubuntu**

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

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

- For **RHEL**, **CentOS**, and **Amazon Linux 2**

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

- For **Ubuntu**

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

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

```
sudo vi /etc/security/limit.conf
appviewx soft nproc 65536
appviewx hard nproc 65536
appviewx soft nofile 65536
appviewx hard nofile 65536
```

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
sudo vi /etc/sudoers

appviewx ALL=(ALL) ALL
```

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

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

k. If access to the internet is restricted, whitelist the AppViewX repository to perform OS patching.

- For **RHEL** and **CentOS**

```
mv /etc/yum.repos.d /etc/yum.repos.d_backup

mkdir -p /etc/yum.repos.d

mv appviewx.repo /etc/yum.repos.d/appviewx.repo

yum clean all

yum update
```

- For **Ubuntu**

i. Create a file, **appviewx_repo.list**, and open it for editing.

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

ii. Add the following to the **appviewx_repo.list** file:

```
#appviewx apt repo

deb https://repos.appviewx.com/repository/ubuntu focal universe

deb https://repos.appviewx.com/repository/ubuntu focal-updates universe

deb https://repos.appviewx.com/repository/ubuntu focal multiverse

deb https://repos.appviewx.com/repository/ubuntu focal-updates multiverse

deb https://repos.appviewx.com/repository/ubuntu focal-backports main restricted universe multiverse

deb https://repos.appviewx.com/repository/ubuntu focal-security main restricted

deb https://repos.appviewx.com/repository/ubuntu focal-security universe

deb https://repos.appviewx.com/repository/ubuntu focal-security multiverse
```

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

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

iv. Add the following to the **auth.conf** file:

```
machine repos.appviewx.com

login xxxxx
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
password xxxxx
```

Note: Contact the AppViewX TAC/CSM team for the login and password.

l. Create a new directory at the location **/home/appviewx** and name it **cc-installer**.

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

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

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

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

2. Access the AppViewX user 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.

   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.

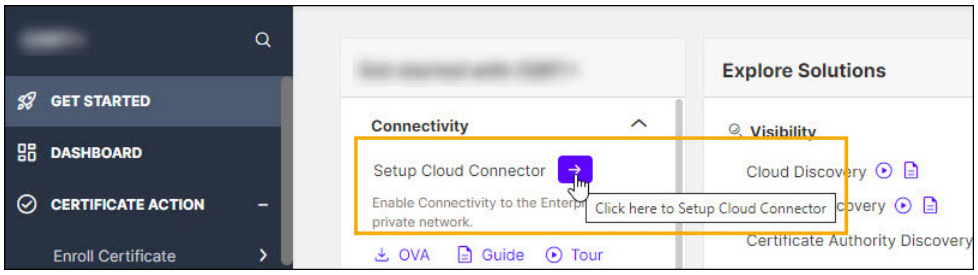
   a. Enter your account URL (for example, https://tenant-name.appvx.com/appviewx/login) in the address bar of your browser.
      The AppViewX login page is displayed.

   b. Login to AppViewX.

   c. Navigate to the cloud connector's setup interface.
      There are three ways you can access the interface for setting up the AppViewX Cloud Connector:

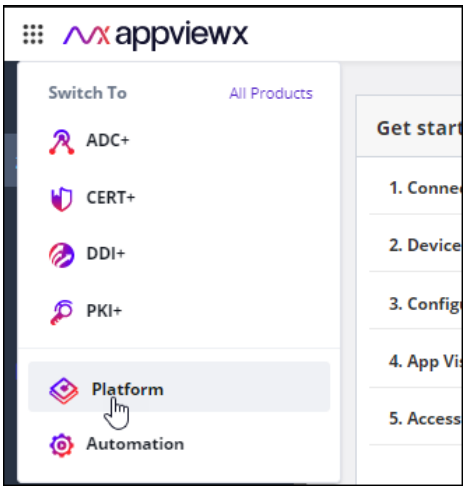
Copyright © 2025 AppViewX Inc. All Rights Reserved.

- From the product landing page (that you will see as soon as you have logged in)
  - Expand the **Connectivity** section and click  .

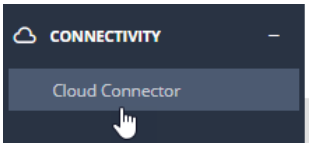


You will be redirected to the **Settings :: Cloud Connector** page.

- From the new navigation menu (displayed by default starting product version 2022.1.0 FP3 onwards):
  i. From the menu in the top-right corner of the page, select **Platform**.



  ii. From the **Platform** menu, under **Connectivity**, click **Cloud Connector**.



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

- From the old navigation menu:

> **Note:** For instructions on switching between the new and the old navigation menus, click here.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

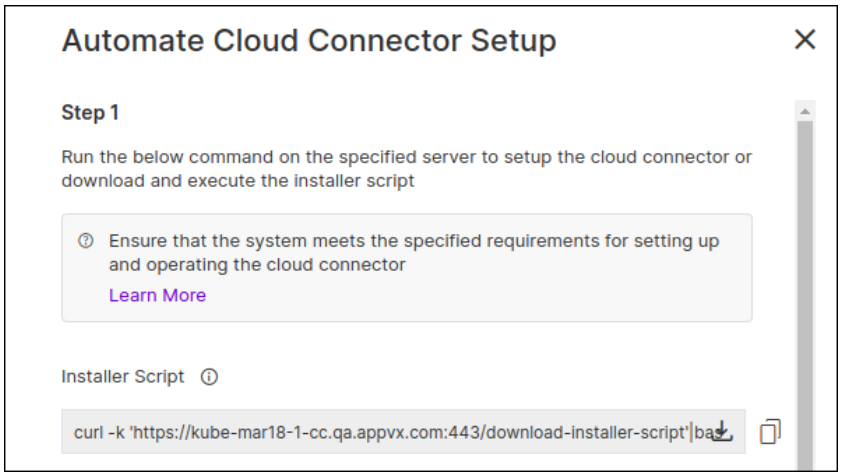
i. From the top right corner of the landing page, click the menu icon.
ii. From the menu displayed, navigate to **Settings** > **Cloud Connector**.

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

3. Install the AppViewX Cloud Connector using the automated script.
To install the AppViewX Cloud Connector using the automated installation script, from the cloud connector details banner, you can either download the installer script or copy the `curl` command.

a. From the cloud connector details banner, click **Download Installer Script**.
OR

Copy the `curl` command to run the installer script.



b. On the host machine:
If you have downloaded the installer script, execute the following commands:

```
chmod +x installer.sh

./installer.sh
```

OR

If you have copied the `curl` command, to execute the command, paste the command on the command line terminal and press **Enter**.

Internet connectivity on the host machine is validated. The installation proceeds only if the host machine has internet access.

Note: For instructions on proceeding with the installation despite not having internet access, click here.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

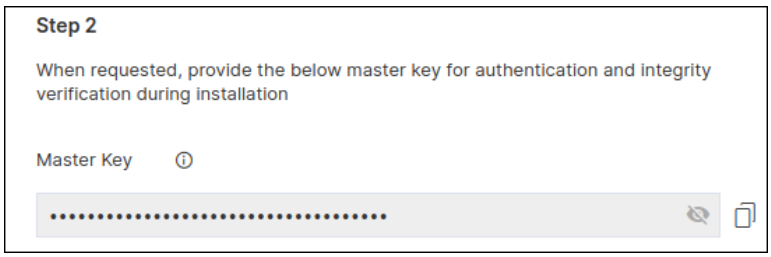
c. When prompted to **Please provide the master key**, enter the required details.
To retrieve the master key:

i. Go to (**Menu**) > **Platform** > **Connectivity** > **Cloud Connector**.

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

ii. From the cloud connector details banner, under **Automate Cloud Connector Setup**, click **Steps to Automate Setup**.

iii. From the **Automated Cloud Connector Setup** window, under **Step 2**, copy the **Master Key**.



iv. Paste the master key in the terminal window and press **Enter**.

The cloud connector installation script will check for the prerequisites and trigger the cloud connector installation.

When the cloud connector instance is successfully installed, a corresponding entry will be listed in the cloud connector inventory.

d. When prompted, **Please enter Datacenter**, enter the name of the data center on which this cloud connector will be deployed.

e. **Note:** Enabling auto-enrollment protocols is recommended only if the cloud connector is being installed in a demilitarized zone (DMZ) or devices in a restricted environment (that disables them from connecting to the <tenant>-aep directly.

When prompted to enable auto-enrollment protocols, enter **y** and enter the protocol name(s) you want to enable. For instructions on enabling auto-enrollment protocols, click here.

Note: By default, only the AppViewX certificate is enabled for auto-enrollment. To enable custom certificate for auto-enrollment:

i. Execute the command `./avxctl upgrade gateway-cert`.

ii. When prompted, enter the location of the custom certificate.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

On successful completion of the setup, a corresponding instance of this cloud connector is displayed in the inventory.

Note: The cloud connector installation on a OVA-based host machine will not prompt you to select if you want to manage F5 Big-IP devices. However, after the cloud connector has been installed you can copy the iControl jar in the deps/external_libs folder (click here for instructions) and restart the starter and platform pods (click here for instructions), to enable this feature.

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

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

sudo systemctl restart sshd
```

4. To approve the cloud connector installation:

   a. Go to (**Menu**) > **Platform** > **Connectivity** > **Cloud Connector**.
   The **Settings :: Cloud Connector** inventory page is displayed.

   b. For the cloud connector instance installed, from the **Actions** column, click **Approve**.

Setting up the AppViewX Cloud Connector using a Packaged Installer

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

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

1. Configure the native OS for the cloud connector installation.

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

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

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

   b. Uninstall the previous version of Docker.

   - **For RHEL, CentOS, and Amazon Linux 2**:

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

   - **For Ubuntu**

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

   c. Configure the hostname and the nameserver.

   i. Configure the hostname: `sudo hostnamectl set-hostname "hostname"`

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

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

   iii. Replace the default nameserver IP address.

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

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

```
sudo vi /etc/resolv.conf

nameserver <nameserver ip>
```

d. Add Docker repo and install Docker.

- For **RHEL** and **CentOS**

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

sudo systemctl start docker

sudo systemctl enable docker

sudo systemctl status docker
```

- For **Amazon Linux 2**

```
sudo yum install -y docker containerd

sudo systemctl start docker

sudo systemctl enable docker

sudo systemctl status docker
```

- For **Ubuntu**

```
sudo apt-get update

sudo apt-get install ca-certificates curl gnupg

#Add Docker’s official GPG key for Ubuntu:

sudo install -m 0755 -d /etc/apt/keyrings

curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

sudo chmod a+r /etc/apt/keyrings/docker.gpg

#Use the following command to set up the repository:

echo \

 "deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \  "$(. /etc/os-release &&
 echo "$VERSION_CODENAME")" stable" | \

 sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update
```

To install Docker for Ubuntu

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

sudo systemctl enable docker

sudo systemctl status docker
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

e. Add the appviewx user in the Docker group.

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

f. Install NTP.

- For **RHEL** and **CentOS**

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

- For **Amazon Linux 2**

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

- For **Ubuntu**

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

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

- For **RHEL**, **CentOS**, and **Amazon Linux 2**

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

- For **Ubuntu**

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

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

```
sudo vi /etc/security/limit.conf
appviewx soft nproc 65536
appviewx hard nproc 65536
appviewx soft nofile 65536
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
appviewx hard nofile 65536
```

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

```
sudo vi /etc/sudoers

appviewx ALL=(ALL) ALL
```

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

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

k. If access to the internet is restricted, whitelist the AppViewX repository to perform OS patching.

- For **RHEL** and **CentOS**

```
mv /etc/yum.repos.d /etc/yum.repos.d_backup

mkdir -p /etc/yum.repos.d

mv appviewx.repo /etc/yum.repos.d/appviewx.repo

yum clean all

yum update
```

- For **Ubuntu**

i. Create a file, **appviewx_repo.list**, and open it for editing.

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

ii. Add the following to the **appviewx_repo.list** file:

```
#appviewx apt repo

deb https://repos.appviewx.com/repository/ubuntu focal universe

deb https://repos.appviewx.com/repository/ubuntu focal-updates universe

deb https://repos.appviewx.com/repository/ubuntu focal multiverse

deb https://repos.appviewx.com/repository/ubuntu focal-updates multiverse

deb https://repos.appviewx.com/repository/ubuntu focal-backports main restricted universe multiverse

deb https://repos.appviewx.com/repository/ubuntu focal-security main restricted

deb https://repos.appviewx.com/repository/ubuntu focal-security universe

deb https://repos.appviewx.com/repository/ubuntu focal-security multiverse
```

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

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

iv. Add the following to the **auth.conf** file:

```
machine repos.appviewx.com

login xxxxx
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
password xxxxx
```

Note: Contact the AppViewX TAC/CSM team for the login and password.

l. Create a new directory at the location **/home/appviewx** and name it **cc-installer**.

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

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

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

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

2. Access the AppViewX user 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.

   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.

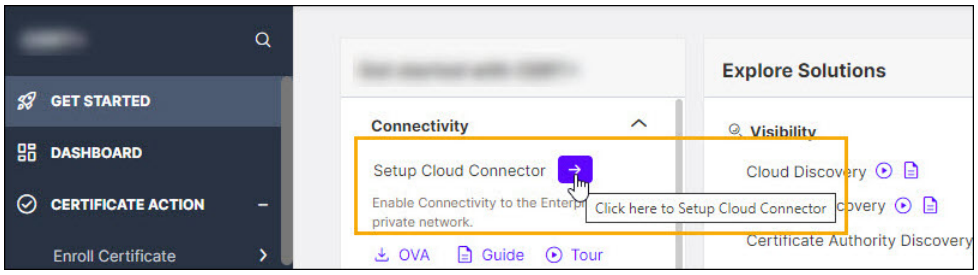
   a. Enter your account URL (for example, https://tenant-name.appvx.com/appviewx/login) in the address bar of your browser.
      The AppViewX login page is displayed.

   b. Login to AppViewX.

   c. Navigate to the cloud connector's setup interface.
      There are three ways you can access the interface for setting up the AppViewX Cloud Connector:

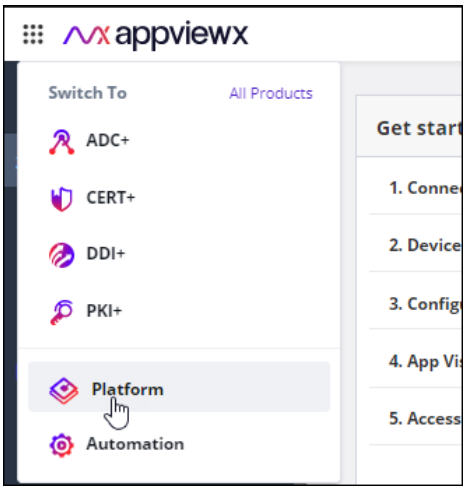
Copyright © 2025 AppViewX Inc. All Rights Reserved.

- From the product landing page (that you will see as soon as you have logged in)
  - Expand the **Connectivity** section and click  .

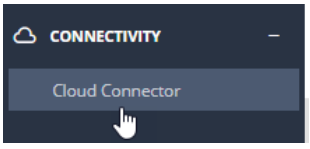


You will be redirected to the **Settings :: Cloud Connector** page.

- From the new navigation menu (displayed by default starting product version 2022.1.0 FP3 onwards):
  i. From the menu in the top-right corner of the page, select **Platform**.



  ii. From the **Platform** menu, under **Connectivity**, click **Cloud Connector**.



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

- From the old navigation menu:

> **Note:** For instructions on switching between the new and the old navigation menus, click here.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

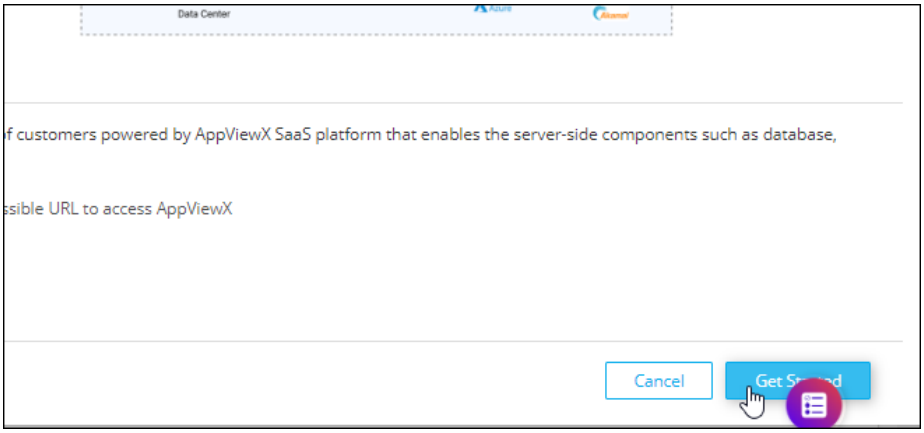
i. From the top right corner of the landing page, click the menu icon.
ii. From the menu displayed, navigate to **Settings** > **Cloud Connector**.

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

3. On the **Setting :: Cloud Connector** page, click **Add Cloud Connector**.
The **Cloud Connector Setup** screen is displayed.

The landing page gives you a quick introduction to the AppViewX Cloud Connector, with a graphical representation of how the infrastructure is deployed and works.

4. To start with the process of adding the cloud connector, from the bottom-right corner of the screen, click **Get Started**.



You will be redirected to the **Basic Information** page.

5. On the **Basic Information** page, configure the basic cloud connector settings.

a. From **Installation Type**, select **Native OS**.

b. In the **Cloud Connector Name (FQDN)** field, enter the hostname of the machine on which the AppViewX Cloud Connector will be installed.

> **Tip:** To retrieve the hostname, from the command line terminal of the host machine, execute the following command: `hostname -f`

> **Note:** The hostname entered here is added to the license file that will be generated and downloaded as part of the installer. Therefore, the license file can be used to install the cloud connector only on the machine with the entered hostname and no other.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

> **Tip:** The **Setup Cloud Connector** section to the right of the **Basic Information** screen lists hyperlinks to the prerequisites required for setting up the AppViewX Cloud Connector. To read more about what the AppViewX Cloud Connector offers, click **Learn More**.

c. Click **Next**.

6. [Optional] Execute a 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.

> **Note:** This is an **optional** step. The prerequisite check script is executed automatically at the time of installing the AppViewX Cloud Connector and the results are shown as a part of the installation logs.

   a. On the **Basic Information** screen, under **Setup Cloud Connector**, you will see a list of the installation prerequisites.

      From this list, for **Executing the Prerequisites Check Script**, to download the script, click .
      The **pre-requisite-check.sh** script file is downloaded.

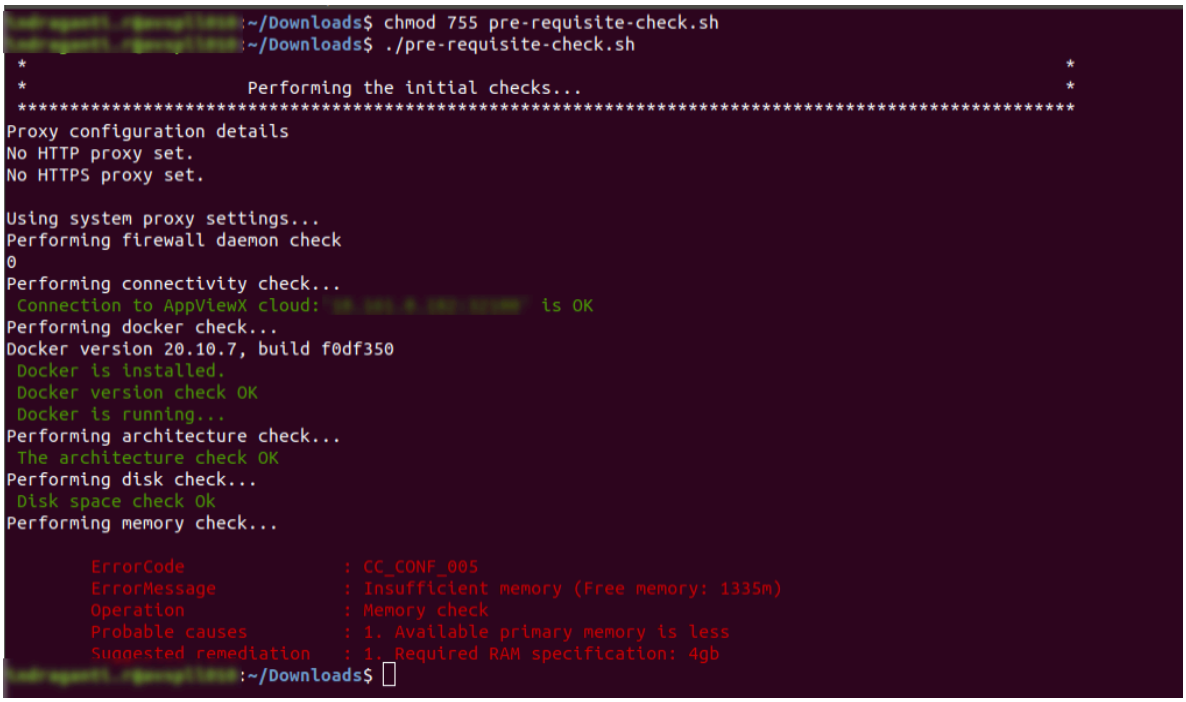
   b. Securely copy the **pre-requisite-check.sh** via SCP/SFTP to the host machine where the AppViewX Cloud Connector is to be installed

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

   d. Execute the **.sh** prerequisite check script file: `./pre-requisite-check.sh`
      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.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



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

7. Click **Next**.
   You will be navigated to the **AssignData Center** screen, where, for deploying the AppViewX Cloud Connector, you can either select an existing data center or add a new one.
8. To use an existing data center, select one from the options displayed on the **Assign Data Center** screen.

> **Tip:** Alternatively, you can use the **Search…** field on this screen to search for an existing data center.

To add a new data center:

a. Click **Add Data Center.**

b. In the **Add Data Center** dialog box, enter a name for the new data center.

c. Click **Save**.
   The new data center will now be displayed on the **Assign Data Center** screen along with the other existing data centers.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

d. Select the required data center.

> **Tip:** The **Data Center based routing** section to the right of the **Assign Data Center** screen explains the concept of data center-based routing and how you can achieve high availability. To read more on this, click **Learn More** from the top-right corner of this screen.

9. Click **Next**.
   The **Advanced Configuration** screen is displayed.
10. On the **Advanced Configuration** page, to configure the TLS authentication and proxy server settings for your cloud connector:

    a. Enter/Select the advanced configuration settings for the AppViewX Cloud Connector.

> **Note:** The **Data center** field is auto-populated based on your selection on the **Assigning a Data Center** screen.

Field	Description
TLS Authentication	**Tip:** The section on the right of the screen gives you a brief context of what is **TLS Authent** • To auto-generate a TLS certificate, select **Auto-generate** (default selection). By default, 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** c • To enter details of a custom TLS certificate, select **Custom**. The **TLS Certificate Password** and **Custom TLS Certificate** fields are displayed. The instructio
TLS Certificate Password*	**Note:** This field is displayed only if you have selected to enter details of a Custom TLS certi Password of the TLS certificate (that will be uploaded in the next step)

Copyright © 2025 AppViewX Inc. All Rights Reserved.

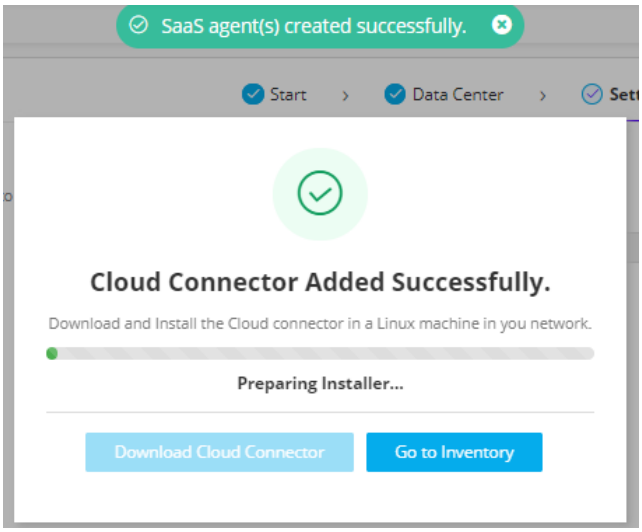
Field	Description
	Note: This is a mandatory field if a Custom TLS certificate is uploaded. AppViewX supports
TLS Certificate	**Note:** This field is displayed only if you have selected to enter details of a Custom TLS certi To upload a custom TLS certificate: i. To navigate to the location of the custom TLS certificate, click within the field. ii. Select the certificate file. iii. Click **Open**. iv. To upload the custom TLS certificate selected, click **Upload**. **Note:** AppViewX supports only password-protected Custom TLS Certificates.
Use proxy	**Tip:** The section on the right of the screen gives you a brief context of what is **Proxy based** A proxy server is required if the AppViewX Cloud Connector is unable to connect to your endpoints To use a proxy server for the deployment: i. Select the **Use proxy** checkbox. ii. To select a preconfigured proxy (for the selected data center), from the **Select Proxy** dropdown **OR** To create a new proxy server setting: i. Use the Click here option shown below the **Select Proxy** dropdown list. The **Add Proxy** pop-up screen is displayed. ii. Enter/Select the details required to add a proxy. **Field descriptions for the Add Proxy details** **Field** | ***Proxy Name** | Na

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Field	Description
	Field
	***Server IP**
	***Port**
	URL
	Authentication
	***Username**
	***Password**

b. Click **Finish**.

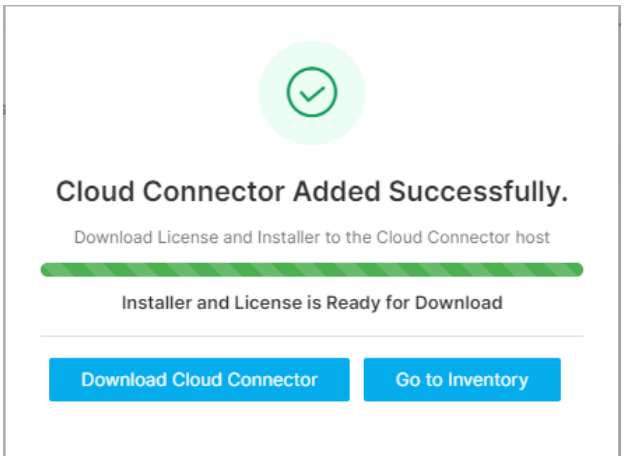
A confirmation message is displayed. AppViewX begins preparing the installer that you can download and proceed with the installation of the AppViewX Cloud Connector.



11. Download the cloud connector installer and license file.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

a. On the **Cloud Connector Added Successfully** dialog box, when the **Installer and License is Ready for Download**, click **Download Cloud Connector**.



> **Tip:** Alternatively, you can click **Go to Inventory** and download the installer from the AppViewX Cloud Connector inventory.

> **Tip:** You can also choose to download the license file and the installer package individually. To do this:
>
> i. From the cloud connector inventory, click the **Cloud Connector Name** (for which you want to download the license file and the installer package).
>
> The selected cloud connector's details are shown in a pane to your right.
>
> ii. To download the AppViewX Cloud Connector installer package, click **Download Cloud Connector**. This is especially useful in the event that the installer has been deleted or is no longer usable.
>
> To download the license file, click **Download License**.

b. Save the installer and the license file on the host machine.

On the **Settings :: Cloud Connector** page, details of this AppViewX Cloud Connector are added in the inventory table, which is explained here.

12. Install the AppViewX Cloud Connector Agent.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Note: The following steps assume that:

- All system prerequisites are fulfilled by the host machine.
- The AppViewX Cloud Connector installer and license file are securely copied via SCP/SFTP to the host machine where the cloud connector is to be installed.
- For installation on RHEL8+, the user must have **sudo** access with **read/write/execute** permissions for the following directories at the least:
  - **/var/lib**
  - **/etc**
  - **/run**
  - **/usr/local/bin**
  - **/tmp**

a. 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`

b. 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 check if the installation prerequisites for the AppViewX Cloud Connector have been fulfilled.

Note:
Ensure that the license file is placed in the same location as the **install.sh** script. If the license file is placed in another location, run the install.sh script using the following command:

```
./install.sh <complete path of the license file with the filename>
```

On successful verification of the prerequisites, you will be prompted to specify if you want to manage f5 BIG-IP devices and if you need auto-enrollment of the certificates.

```
Do you want to manage f5 BIG-IP devices? (y/n)?:n
Continuing with the installation

Do you need Auto-enrollment of the certificate using EST/SCEP/ACME? (y/n)?:y
Please choose one or more protocol (use comma separated numbers): 1)EST(MTLS)   2)SCEP(HTTP)  3)ACME(HTTPS)
1,2,3
Auto enrollment enabled successfully for protocol(s): MTLS HTTP HTTPS
Do you want to enable Syslog receiver for a near real time configuration updates from the devices. (y/n) n
syslog enabled  n
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

c. When prompted, enter the required input value(s):

> **Important:** If you choose to **not enable** any of the following features, to enable them later, you will have to reinstall the AppViewX Cloud Connector.

i. If you want manage f5 BIG-IP devices, enter **y**/**n** for yes/no, respectively.

ii. When prompted to enable auto-enrollment of the certificate using one of the following supported auto-enrollment protocols, enter **y** only if the cloud connector is being installed in a demilitarized zone (DMZ) or devices in a restricted environment (that disables them from connecting to the **<tenant>-aep** directly.

iii. If you choose **y** (yes) here, enter the required protocol(s) name.

> **Note:** By default, the AppViewX certificate is enabled for auto-enrollment. To enable custom certificate for auto-enrollment:
>
> i. Execute the command `./avxctl upgrade gateway-cert`.
>
> ii. When prompted, enter the location of the custom certificate.

> **Note:** If you are a KUBE+ customer, the auto-enrollment gateway should be enabled as part of the installation for your KUBE+ usecases to work via the cloud connector.

d. If you want to enable Syslog receiver for a near-real time configuration updates from the devices, enter **y**/**n** for yes/no, respectively. For configuring Syslog reception, refer to Platform User guide section, Syslog Reception.
In case you have an older version of AppViewX on cloud and want to make use of Syslog capabilities for ADC, you must manually activate the Syslog flag by setting SYSLOG_ENABLED=`true` in the path ccpath/deps/properties.

e. Enter the sudo password.
After the relevant details have been entered, the installation proceeds. Installation logs, according to the outcome of the installation, are displayed.

Given below are the sample installation logs:

For installation on RHEL8+:

```
Moving images and k3s binary as required for k3s installation...


Triggering k3s install.....
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
[INFO]  Skipping k3s download and verify
[INFO]  Skipping installation of SELinux RPM
[WARN]  Failed to find the k3s-selinux policy, please install:
    dnf install -y container-selinux
    dnf install -y https://rpm.rancher.io/k3s/stable/common/centos/8/noarch/k3s-selinux-0.4-1.el8.noarch.rpm

[INFO]  Creating /usr/local/bin/kubectl symlink to k3s
[INFO]  Creating /usr/local/bin/crictl symlink to k3s
[INFO]  Creating /usr/local/bin/ctr symlink to k3s
[INFO]  Creating killall script /usr/local/bin/k3s-killall.sh
[INFO]  Creating uninstall script /usr/local/bin/k3s-uninstall.sh
[INFO]  env: Creating environment file /etc/systemd/system/k3s.service.env
[INFO]  systemd: Creating service file /etc/systemd/system/k3s.service
[INFO]  systemd: Enabling k3s unit
Created symlink /etc/systemd/system/multi-user.target.wants/k3s.service → /etc/systemd/system/k3s.service.
[INFO]  systemd: Starting k3s



k3s install success. Backing up the kubeconfig...
Importing CC base image...
unpacking docker.io/library/avx-mid-server-base:22.1.0.0 (sha256:5e1948b797dd19382f50faf06f921645337d53a1736016db81cd680c0afd7317)...done
************************************************
adding nameservers in coredns configmap

configmap/coredns configured
************************************************
Deploying the Cloud Connector...
NAME: avx-mid-server-starter
LAST DEPLOYED: Wed May 10 11:02:29 2023
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
************************************************************************
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
*    Congratulations!!! The installation completed successfully.   *
*    Please wait till the Cloud Connector is up and running.      *
**********************************************************************
(1%) Cloud Connector status: Running
 Cloud Connector is up and running.
```

Note: If selinux is enabled on the node and is set to **enforcing**, the warning **Failed to find the k3s-selinux policy...** will show up in the logs. This warning can be ignored.

For installation on an OS other than RHEL8+:

```
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)...
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
[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.
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
 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.

13. To approve the cloud connector installation:

    a. Go to (**Menu**) > **Platform** > **Connectivity** > **Cloud Connector**.
    The **Settings :: Cloud Connector** inventory page is displayed.

    b. For the cloud connector instance installed, from the **Actions** column, click **Approve**.

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.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Setting up the AppViewX Cloud Connector using Helm model in a Kubernetes Cluster

Cloud Connectors (CC) can be deployed via Helm in customers' Kubernetes infrastructures, such as EKS, AKS, GKE, and OpenShift. Customers can utilize their existing infrastructure and workforce to manage the cluster efficiently while leveraging their existing monitoring tools and CI/CD pipelines within that infrastructure.

CC can now be provisioned in the managed Kubernetes clusters using Helm, eliminating the need for the CC tarball.

- Appendix: Prerequisites Configuration in the Managed Cluster
- Troubleshooting

Prerequisites

The prerequisites for installing the cloud connector using the Helm model in a Managed Kubernetes cluster are as follows:

- **Kubernetes Cluster:** A running Kubernetes cluster with adequate privileges to deploy applications. The minimum version should be Kubernetes greater than or equal to 1.30.
- **kubectl:** The Kubernetes command-line tool (kubectl) should be installed and configured on the machine where the installation will be triggered to communicate with the Kubernetes cluster. The minimum kubectl version should be v1.30.1
- **Helm:** The helm package manager for Kubernetes, should be installed on the machine.
- **Cluster Access and Permissions:** Sufficient permissions on the Kubernetes cluster are required to create the necessary resources for installing the Cloud Connector. This includes the ability to create resources such as namespaces, pods, secrets, deployments, ReplicaSets, services, service accounts, ConfigMaps, RoleBindings, roles, and persistent volumes.

  Note: The **avx-mid-server-platform** pods must be permitted to run in privileged mode to support network scan functionalities. This requires enabling **allowPrivilegeEscalation**.

- **Helm Chart Repository:** Access to the helm chart repository (AppViewX or private) that contains the chart for AppViewX cloud connector. Make sure the necessary credentials are available to access the repository

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- **Docker image repository:** Access to the docker image repository (AppViewX or private) that contains the images for AppViewX cloud connector. Make sure the necessary credentials are available to access the repository.
- **Network Access:** Network access to pull the Helm chart and associated Docker images from remote repositories (AppViewX or private).

> **Note:** Additionally, network access must be configured to allow communication from the managed Kubernetes cluster to the Cloud Connector instance/deployment. This enables interactions with devices for operations such as discovery and CLM activities. If CA-related operations are involved, internet access must be permitted to allow Cloud Connectors to reach the relevant Certificate Authorities (CAs).

- **Volume Mounting:** For Certificate Lifecycle Management (CLM) in F5 devices and other ADC use cases, volume mounting is required to host the iControl jar. AppViewX can assist in modifying the Helm chart and mounting the volume. If volume mounting is not feasible, support can be offered to manually configure the database to work with the REST API instead of the iControl jar. However, this manual configuration will only enable CLM operations, excluding ADC use cases. Additionally, if the REST API is used, the "Overwrite the Existing Certificate" functionality will not work during the certificate push operation.
- **Load balancer (LB) provisioning:** LB provisioning in Customer's Kubernetes cluster will be required to manage certificates with auto-enrollment protocols such as SCEP, ACME, and EST.
- **Organizational policies or restrictions within the tenants organization:** Any restrictions or policies within the tenant's organization that block the cloud connector's functionality, such as firewall rules or DNS configurations, should be adjusted to support the intended use cases.
- **AppViewX SaaS as Managed Kube deployments or MSPs:** If the AppViewX Server is deployed on-premises in a managed Kubernetes platform, the credentials for accessing Helm charts and images cannot be created through the AppViewX UI. You will need to contact the AppViewX support team to obtain these credentials.
- **Permission to deploy pods:** The cluster must permit deploying pods with `allowPrivilegeEscalation` enabled for the specified name.
- **IoT use case support:** If external access is required, the cluster administrator must configure an additional load balancer to route requests to the service endpoint exposed by the Cloud Connector, as illustrated below.

Service	Type	Port
avx-mid-server-gateway-http	NodePort	30022

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Service	Type	Port
avx-mid-server-gateway-https	NodePort	30020
avx-mid-server-gateway-mtls	NodePort	30021

The services can be accessed within the cluster using format <*service-name*>.<*namespace*>.*svc.cluster.local:*<*port*> for example, avx-mid-server-gateway-http.cc.svc.cluster.local:30022

Remember: In the case of IoT, if the custom certificate needs to be configured instead of the default gateway certificate, it should be configured at the load balancer level that will be provisioned to access the services.

Method 1: Automated Process

In the automated process, a generic Helm command is generated for installing the CC.

1. Go to (**Menu**) > **Platform** > **Cloud Connector** > **Add Cloud Connector** > **Automated** > **Helm**.
2. If you haven't already, generate the credentials to access the AppViewX Helm and Image repositories, then copy the credentials.
3. If you wish to use the custom repository, then select the **Use Internal Repository** radiobutton and enter the details in the **Helm Repository** and **Image Repository** fields and click **Save**. Else, if you choose the default radiobutton **Direct Pull from Internal Repository**, the field details are default populated.
4. Copy the Helm command and execute it in the Kubernetes cluster. When prompted, provide the **Cluster Type** (*EKS* or *OCP*), **Cloud Connector name**, **Datacenter name**, and ***Repository Credentials***.
5. Once the installation is complete in the Kubernetes Cluster, return to (**Menu**) > **Platform** > **Cloud Connector** and search for the cloud connector by the name used during installation. **Approve** it to finalize the installation.

Method 2: Manual Process

In the manual process, the CC details have to be provided to generate a generic Helm command for installing the CC.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

1. Go to (**Menu**) > **Platform** > **Cloud Connector** > **Add Cloud Connector** > **Manual** > **Helm**.
2. Enter the **Cloud Connector Name**, and select the **Data Center**.
3. If you haven't already, generate the credentials to access the AppViewX Helm and Image repositories, then copy the credentials.
4. If you wish to use the custom repository, then select the **Use Internal Repository** radiobutton and enter the details in the **Helm Repository** and **Image Repository** fields and click **Save**. Else, if you choose the default radiobutton **Direct Pull from Internal Repository**, the field details are default populated.
5. Copy the Helm command and execute it in the Kubernetes cluster. When prompted, provide the **Cluster Type** (*EKS* or *OCP*), **Cloud Connector name**, **Datacenter name**, and ***Repository Credentials***.
6. Once the installation is complete in the Kubernetes Cluster, return to (**Menu**) > **Platform** > **Cloud Connector** and search for the cloud connector by the name used during installation. **Approve** it to finalize the installation.

Upgrading Cloud Connectors

When upgrading the Cloud Connector, download the new set of files from the AppViewX server and upload them to the Customer's repository to update the Cloud Connectors.

Note: The **Upgrade** option in the AppViewX server GUI will not work for upgrading Cloud Connectors deployed in customer's Kubernetes infrastructure. Customer must upgrade the Cloud Connector through the CI/CD pipeline.

Use Case Limitations

When deploying the Cloud Connector in the Managed Kubernetes infrastructure, please note the following limitations:

- HSM vendors that require binary installation on the node will not be supported. Thales GPN, Thales TCT, Thales DPoD, and Entrust HSM’s will not work. Utimaco and Fortanix HSM’s will continue to work.
- If the Customer chooses to use the REST API with an F5 device instead of the iControl jar, the "Overwrite the Existing Certificate" functionality will not work during the certificate push operation.
- Device parsing in F5 which requires iControl jar, will not be supported unless the Customer allows mounting external volumes for hosting iControl Jar.
- A10 Functionalities: Device Backup/Restore and Syslog functionalities will not work since they require reverse SSH to the AppViewX Cloud Connector.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- Firewall Functionalities: Backup/Restore in Panorama and Device Parsing in Checkpoint MDS will not work.since they require reverse SSH to the AppViewX Cloud Connector.
- Syslog support and SSH-server are not supported.
- Using client certificate for CC-to-mothership communication is currently not supported through the Helm model.

Appendix: Prerequisites Configuration in the Managed Cluster

Prerequisites Configuration in the OCP Cluster

This section outlines the prerequisites configuration in the OCP cluster before the Helm based CC installation.

In order to perform nmap based discovery, the platform pod in cloud connector will require elevated permissions. Please follow the below steps to modify the security context configuration. A new security context configuration named **scc-appviewx-cc-scc**is to be created with the below permissions before installing the cloud connector.

```
apiVersion: security.openshift.io/v1
kind: SecurityContextConstraints
metadata:
  name: scc-appviewx-cc-scc
allowHostDirVolumePlugin: true
allowHostIPC: false
allowHostNetwork: true
allowHostPID: false
allowHostPorts: true
allowPrivilegeEscalation: true
allowPrivilegedContainer: false
allowedCapabilities: null
defaultAddCapabilities:
  - NET_RAW
fsGroup:
  type: RunAsAny
groups: []
priority: null
readOnlyRootFilesystem: false
requiredDropCapabilities: null
runAsUser:
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
  type: MustRunAsNonRoot
  uidRangeMin: 1
  uidRangeMax: 4294967295
seLinuxContext:
  type: MustRunAs
seccompProfiles:
  - runtime/default
supplementalGroups:
  type: RunAsAny
users: []
volumes:
- configMap
- csi
- downwardAPI
- emptyDir
- ephemeral
- persistentVolumeClaim
- projected
- secret
```

The steps are as follows:

1. Create a namespace cc.

   ```
   kubectl create namespace cc
   ```

2. Create a service account for platform pod with the name **scc-appviewx-cc-sa** in the **cc** namespace.

   ```
   oc create sa scc-appviewx-cc-sa -n cc
   ```

3. Create the below RBAC rule using the Role configuration as shown below.

   ```
   oc apply -f rbac.yaml -n cc
   ```

   ```
   ---
   kind: Role
   apiVersion: rbac.authorization.k8s.io/v1
   metadata:
    name: use-scc-appviewx-cc-scc
   rules:
    - apiGroups: ["security.openshift.io"]
      resources: ["securitycontextconstraints"]
   ```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
  resourceNames: ["scc-appviewx-cc-scc"]
  verbs: ["use"]
---
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
 name: use-scc-appviewx-cc-scc
subjects:
 - kind: ServiceAccount
   name: scc-appviewx-cc-sa
roleRef:
 kind: Role
 name: use-scc-appviewx-cc-scc
 apiGroup: rbac.authorization.k8s.io
```

4. Once the cloud connector is installed, verify whether the **scc** is assigned properly.

```
oc get pod $(kubectl get pods -n cc |grep platform|cut -d ' ' -f 1|grep Running) -n cc -o yaml | grep openshift.io/scc
```

   The above command should print `scc-appviewx-cc-sa` as the value.

5. In order to clean up the security context, perform the below operations (post CC uninstall).

```
oc delete scc/scc-appviewx-cc-scc
```

```
oc delete sa/scc-appviewx-cc-sa -n cc
```

```
oc delete role/use-scc-appviewx-cc-scc -n cc
```

```
oc delete rolebindings/use-scc-appviewx-cc-scc -n cc
```

Required Permissions for Managed Kubernetes Clusters (EKS, GKE, AKS)

For managed Kubernetes clusters such as **EKS, GKE, and AKS**, the following permissions are required:

- A **ServiceAccount** will be used to grant **RBAC permissions** without the need to create a separate user.
- **ServiceAccounts** are built-in Kubernetes resources specifically designed for assigning permissions to workloads (e.g., Pods) rather than individual users or external entities.

Role

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
metadata:
  name: appviewx-cc-role
rules:
- apiGroups: ["rbac.authorization.k8s.io"]
  resources:
  - pods
  - deployments
  - replicasets
  - services
  - serviceaccounts
  - configmaps
  - cronjobs
  - label
  verbs:
  - get
  - list
  - watch
```

Role binding

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: appviewx-cc-rolebinding
subjects:
- kind: ServiceAccount
  name: appviewx-cc-sa
roleRef:
  kind: Role
  name: appviewx-cc-role
  apiGroup: rbac.authorization.k8s.io
```

Service Account

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: appviewx-cc-sa
  #namespace: my-namespace
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Enabling LoadBalancer for IoT Support

It is advisable for the cluster administrator to configure a dedicated load balancer to support IoT protocols.

Alternatively, the IoT port can be enabled via a LoadBalancer. To do this, update the installer script by changing `serviceType=NodePort` to `serviceType=LoadBalancer` before installing the Helm module.

> ⚠ **Warning:** Enabling the load balancer by changing the `serviceType` will expose 3 publicly accessible Load Balancers for each of the IoT protocols.

Uninstallation and Cleanup for Helm Model

To uninstall perform the following:

```
namespace=cc
```

```
helm uninstall avx-mid-server -n $namespace;kubectl delete namespace $namespace;helm repo remove appviewx-helm-repo
```

Troubleshooting

Here are some useful commands with the assumption that the namespace used is **cc**.

Removing Namespace cc Stuck as Terminating

Execute the following command:

```
NS=`kubectl get ns | grep cc| grep Terminating | awk 'NR==1 {print $1}'` && kubectl get namespace "$NS" -o json   | tr -d "\n" | sed "s/\"finalizers\":
\[[^]]\+\]/\"finalizers\": []/"   | kubectl replace --raw /api/v1/namespaces/$NS/finalize -f -
```

Verifying Logs of all the Platform Pods

Execute the following commands:

```
kubectl logs -f --selector app.kubernetes.io/name=platform -n cc
```

```
kubectl logs -f --prefix -l app.kubernetes.io/name=platform -n cc
```

Performing a Rollout Restart

Execute the following command:

```
kubectl rollout restart deployment -n cc
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Listing Helm Repo Charts

Execute the following command:

```
helm search repo -l
```

Managing ADC Devices

The following ADC vendors can be managed within the AppViewX cloud environment:

- F5
- AVI
- NginxPlus
- A10
- Citrix
- HAProxy
- BigIQ
- Array
- F5XC

To manage F5 BIG-IP and A10 devices and to handle certificates through the AppViewX Cloud Connector, ensure the following prerequisites are met:

- Prerequisites for F5 BIG-IP Devices
- Prerequisites for A10 Devices
- Enabling the Device Backup Bot Pod

Prerequisites for F5 BIG-IP Devices

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

1. Ensure you have a Licensed version of the iControl jar for F5 BIG-IP devices (Refer: https://devcentral.f5.com/s/articles/iControl-Library-For-Java-With-Source).
2. Download the **axis.jar** from the axis library.
3. Download the **javax.xml.soap-api-1.4.0.jar** from the file library.
4. In the AppViewX Cloud Connector installation package, copy the **iControl jar**, **axis jar**, and the **javax.xml.soap-api-1.4.0.jar** to the folder **/deps/external_libs**.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Note: If the pre-requisite libraries are placed manually to the installed/upgraded Cloud Connector, then restart the pods by executing the commands below.

```
cd <cc_installation_directory>/deps/tools
```

```
./k3s kubectl delete pods -A --all --force
```

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

Prerequisites for A10 Devices

For certificate management and device backup on A10 devices, navigate to the **Authentication Settings** and configure the **Node Password** to match the password of the node where the AppViewX Cloud Connector instance is deployed.

For detailed instructions on how to configure the Authentication Settings, click here.

Enabling the Device Backup Bot Pod

To improve system efficiency and offload backup operations from core subsystems, the `avx_device_backup_bot` pod can be enabled to handle all ADC device backup requests.

- Enable the avx_device_backup_bot Pod
- Enable the ROUTE_ADC_BACKUP Flag
- Disable the ROUTE_ADC_BACKUP Flag

Enable the avx_device_backup_bot Pod

1. Navigate to the installer scripts directory:
   `cd <InstallerLocation>/appviewx_kubernetes/scripts`
2. Open the configuration file:
   `vi appviewx.conf`
3. Update the `ENABLED_PLUGINS` section to include:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
appviewx_dependencies

avx_platform_gateway

avx_device_backup_bot
```

4. Specify the datacenter for the `avx_device_backup_bot`:

   Example: `avx_device_backup_bot=absecon`

5. Save and exit the configuration file:

   `:wq`

6. Install the pod:

   `./plugins_install.sh`

Enable the ROUTE_ADC_BACKUP Flag

1. Login to the MongoDB interactive shell:

   `kubectl exec -it mongodb-0 -n avx -- mongo --authenticationDatabase admin -u admin -p `grep "mongo" ~/appviewx/.appviewx_configuration | cut -d "=" -f 2` appviewx`

2. Enable the routing flag:

```
db.getCollection('avx_app_metadata').updateOne(
   { "_id": "ROUTE_ADC_BACKUP" },
   {
      $set: {
         "description": "Route ADC Backup adc-device-backup-bot pod",
         "subsystem": "adc",
         "value": "true"
      }
   },
   { upsert: true }
)
```



Note: Backup requests will be routed to the avx_device_backup_bot pod only if the `ROUTE_ADC_BACKUP` flag is set to `true`.

Disable the ROUTE_ADC_BACKUP Flag

Copyright © 2025 AppViewX Inc. All Rights Reserved.

1. Login to the MongoDB shell:

   `kubectl exec -it mongodb-0 -n avx -- mongo --authenticationDatabase admin -u admin -p `grep "mongo" ~/appviewx/.appviewx_configuration | cut -d "=" -f 2` appviewx`

2. Disable the routing flag:

```
db.getCollection('avx_app_metadata').updateOne(
   { "_id": "ROUTE_ADC_BACKUP" },
   {
      $set: {
         "description": "Route ADC Backup adc-device-backup-bot pod",
         "subsystem": "adc",
         "value": "false"
      }
   },
   { upsert: true }
)
```

Installing the AppViewX Windows Gateway

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

- To download the AppViewX Windows Gateway, from the AppViewX CERT+ landing page, under **Get started with CERT+** > **Connectivity**, click Windows Gateway .
- To install and manage the AppViewX Windows Gateway, refer to the AppViewX Windows Gateway Setup Guide.

Troubleshooting the AppViewX Cloud Connector

- Managing Certificates on F5 BIG-IP Devices
- AppViewX Cloud Connector Health
- Connectivity Checks
- Installation Errors
- Log Analysis

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- Steps to check pod status
- Steps to restart pods

Managing Certificates on F5 BIG-IP Devices

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

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

AppViewX Cloud Connector Health

Amber/red health indicator

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

CC heath check using script

In cases where patching is frequently performed on the Linux machine or the customer frequently reboots the system, AppViewX has developed a script to ensure that the CC operates without issues. This script monitors the CC health status as needed, ensuring that patch updates run without interruption. The script performs the following checks:

- **Resources** – Primarily monitors memory and CPU, including docker storage, disk space, docker status, and memory usage. Use the following command:

  ```
  ./avxctl status resources
  ```

  The list of check include:
  - docker storage Driver check
  - disk checks
  - /home/appviewx/cloud-connector disk space check

Copyright © 2025 AppViewX Inc. All Rights Reserved.

  - docker root directory disk space check
  - memory check
- **Services** – Since the CC runs in a Kubernetes cluster, certain Linux system services, such as Firewall, nftables, pupppet, etc. may interfere with communication between the CC and the mother-ship. The script verifies these services to ensure seamless connectivity. Use the following command:

```
./avxctl status services
```

  - firewall daemon check
  - nftables check
  - pupppet services check
  - docker check
  - IPv6 check
  - IP forwarding check
  - SELinux check
  - Firewall status check
- **CC Communication** - The utility exposes a service in the CC, this service call checks if any successful communication has happened in the last 5 minutes. A successful communication indicates that CC is healthy. Use the following command:

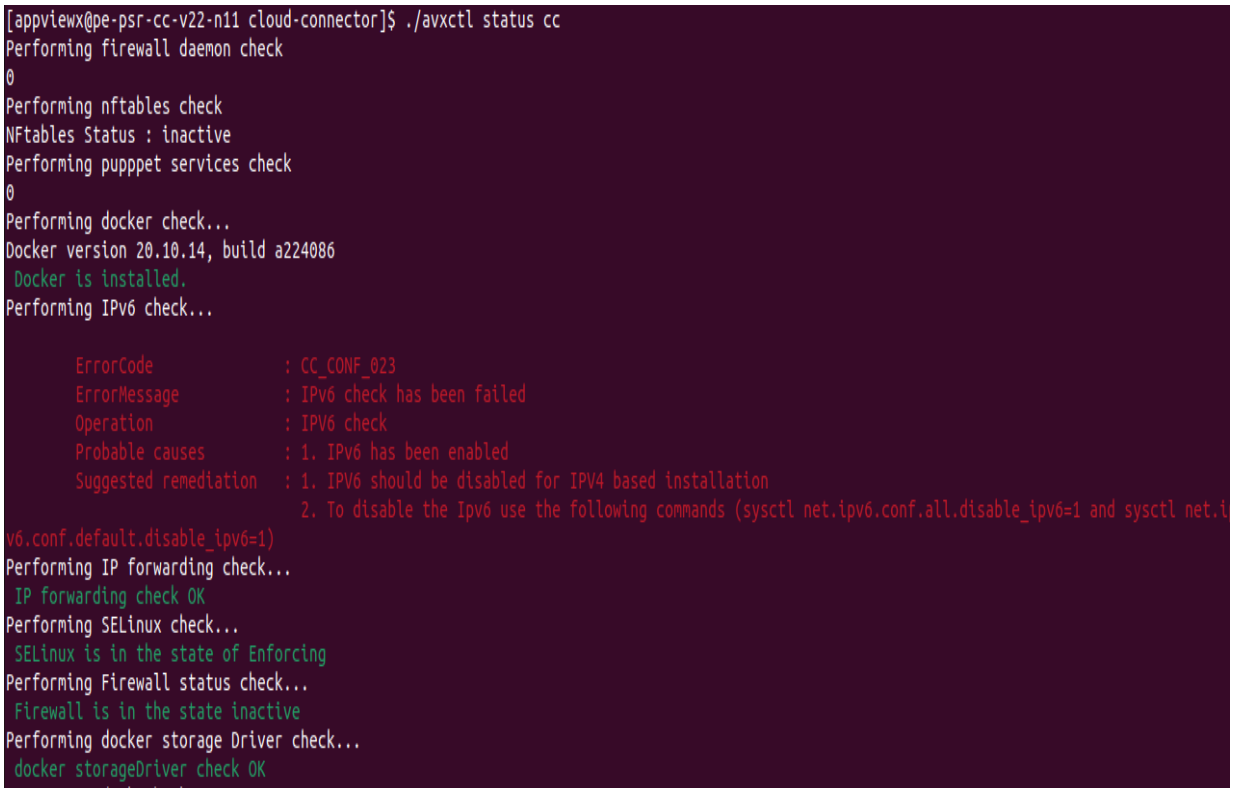
```
./avxctl status cc
```

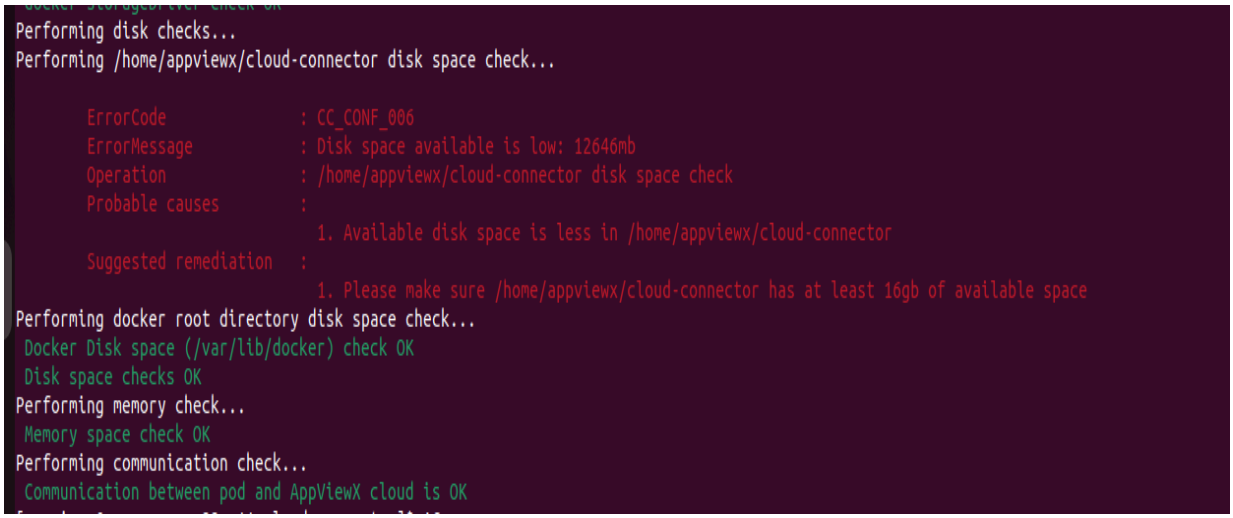


Note: The above command performs the resources and services checks along with the CC communication.

  - firewall daemon check
  - nftables check
  - pupppet services check
  - docker check
  - IPv6 check
  - IP forwarding check
  - SELinux check
  - Firewall status check
  - docker storage Driver check
  - disk checks
  - /home/appviewx/cloud-connector disk space check
  - docker root directory disk space check
  - memory check
  - communication check

Copyright © 2025 AppViewX Inc. All Rights Reserved.





Connectivity Checks

- Scenario 1: At the time of installation
  - Check if the AppViewX Cloud Connector is able to reach the AppViewX cloud.
- Scenario 2: After the package has been successfully installed

Copyright © 2025 AppViewX Inc. All Rights Reserved.

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

Installation Errors

Prerequisite check failure

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

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

> **Note:** Since RHEL8+ excludes Docker support, Docker prerequisites are not applicable when the AppViewX Cloud Connector is being installed on a RHEL 8+ node.

SHA256 checksum failure

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

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

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

Installation Errors

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Full list of installation error codes and their resolutions

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Full list of installation error codes and their resolutions (continued)

Error Code	Error Message	Resolution
CC_CONF_006	Disk space available is low: <available disk space in MB>	Minimum available disk space required: 16GB
CC_CONF_007	Docker not running or not accessible for non sudoers	1. To check the Docker status, execute **one** of the following commands: • `service docker status` • `systemctl status docker` 2. To start the Docker, execute **one** of the following commands: • `service docker start` • `systemctl start docker` 3. Ensure that the Docker is accessible to non sudoers. • For post-installation steps for Linux: https://docs.docker.com/engine/install/linux-postinstall/ • In the event of a VM reboot, the Docker needs to be restarted. To configure the Docker to restart on boot, follow the instructions given here.
CC_CONF_008	Cluster already exists.	1. Uninstall the AppViewX Cloud Connector. 2. Reinstall the AppViewX Cloud Connector in the same/different node.
CC_CONF_009	firewalld is running	• Execute the following script to open the port in firewalld that requires sudo access: `./deps/utils/open-ips-ports-firewalld.sh`

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Full list of installation error codes and their resolutions (continued)

Error Code	Error Message	Resolution

		OR
		
		• Execute the following commands:
		
		`sudo firewall-cmd --permanent --add-port=22/tcp`
		`sudo firewall-cmd --permanent --add-source=10.42.0.0/16`
		`sudo firewall-cmd --permanent --add-source=10.43.0.0/16`
		`sudo firewall-cmd --direct --permanent --add-rule ipv4 filter FORWARD 1 -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT`
		`sudo firewall-cmd --permanent --add-forward-port=port=30020:proto=tcp:toport=30020:toaddr=`
		`sudo firewall-cmd --permanent --add-forward-port=port=30021:proto=tcp:toport=30021:toaddr=`
		`sudo firewall-cmd --permanent --add-forward-port=port=30022:proto=tcp:toport=30022:toaddr=`
		`sudo firewall-cmd --reload`

CC_CONF_010	Not met cpu requirement:- No of available processors(vCPU): <number>	The required number of processors (vCPU) is 4.
CC_CONF_011	Docker running with Incompatible storage Driver	Update the storage driver to **overlay2**. For setup instructions, click here.
CC_CONF_012	A default route is not available in the tenant premises or the tenant is not connected to a network.	• Add a default route with an IP address. • Ensure that the network connection is up.
CC_CONF_013	Cluster already exists	Uninstall the AppViewX Cloud Connector.
CC_CONF_015	Low available disk space	Minimum available (free) disk space required for **/var/lib**: **5 GB**

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Full list of installation error codes and their resolutions (continued)

Error Code	Error Message	Resolution
		In case of restrictions in meeting this requirement, it is recommended to change the data root directory from **/var/lib** to another dedicated directory. For instructions on changing the data root directory, click here.
CC_CONF_016	Low available disk space	Minimum available (free) disk space required for **/run**: **3 GB** **Note:** Currently, in the event that this requirement is not met, a workaround is not available.
CC_CONF_017	`systemctl` command not found	Ensure that your operating system has `systemctl` and `systemd` installed.
CC_CONF_021	Local IP is configured as Nameserver	1. On the command line terminal, execute the following command: `cat /etc/resolv.conf` Contents of the resolv.conf file are displayed, which include the nameserver IP addresses. 2. Check the nameserver IP address(es) in the resolv.conf file. **Scenario 1**: If the nameserver IP address is other than 127.0.0.*: a. From the resolv.conf file, copy one (or more) IP address(es). b. On the command line terminal of the node where the cloud connector is installed, navigate to the location: **(Cloud_Connector_Installed_Folder)/deps/tools**. c. To edit the config map of the coredns pod, execute the following command: `./k3s kubectl edit cm coredns -n kube-system`

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Full list of installation error codes and their resolutions (continued)

Error Code	Error Message	Resolution
		d. Replace the forward IP address in the coredns pod config map with the IP address(es) copied from the resolv.conf file. e. Restart the coredns pod: ./k3s kubectl delete pod <COREDNS_POD_NAME> -n kube-system. **Scenario 2**: Nameserver IP address in the resolv.conf file is 127.0.0.*: a. Execute the following command: `/run/systemd/resolve/resolv.conf`. b. Now, check the nameserver IP address(es). **Scenario 2.1**: If the nameserver IP address is other than 127.0.0.*: i. Copy the IP address(es). ii. On the command line terminal of the node where the cloud connector is installed, navigate to the location: **(Cloud_Connector_Installed_Folder)/deps/tools**. iii. To edit the config map of the coredns pod, execute the following command: `./k3s kubectl edit cm coredns -n kube-system` iv. Replace the forward IP address in the coredns pod config map with the IP address(es) copied from the resolv.conf file. v. Restart the coredns pod: `./k3s kubectl delete pod <COREDNS_POD_NAME> -n kube-system`. **Scenario 2.2**: If the /run/systemd/resolve directory is not created: i. On the command line terminal of the node where the cloud connector is installed, navigate to the location:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Full list of installation error codes and their resolutions (continued)

Error Code	Error Message	Resolution
		(Cloud_Connector_Installed_Folder)/deps/tools. ii. To edit the config map of the coredns pod, execute the following command: `./k3s kubectl edit cm coredns -n kube-system` iii. Replace the forward IP address in the coredns pod config map with the location: /etc/resolv.conf. iv. Restart the coredns pod: `./k3s kubectl delete pod <COREDNS_POD_NAME> -n kube-system`.
CC_INSTALL_001	Moving images and k3s binary failed	• Ensure that the user has **write** permission to the **/var/lib** and the **/usr/local/bin** directories. • Ensure that the sha256sum of the downloaded installer package matches with that of the corresponding cloud connector in the AppViewX Cloud Connector inventory. In case of a mismatch, redownload the package.
CC_INSTALL_002	Triggering k3s install failed	• Ensure that the sha256sum of the downloaded installer package matches with that of the corresponding cloud connector in the AppViewX Cloud Connector inventory. In case of a mismatch, redownload the package

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Full list of installation error codes and their resolutions (continued)

Error Code	Error Message	Resolution
CC_INSTALL_003	Setting Kube config failed	Uninstall and reinstall the AppViewX Cloud Connector.
CC_INSTALL_004	Importing CC base image failed	• Uninstall and reinstall the AppViewX Cloud Connector. • Ensure that the sha256sum of the downloaded installer package matches with that of the corresponding cloud connector in the AppViewX Cloud Connector inventory. In case of a mismatch, redownload the package
CC_PLATFORM_001	Failed to untar the upgrade dependencies	For SRE: 1. Retry the upgrade operation once. 2. If the upgrade operation fails, reinstall the Cloud Connector.
CC_PLATFORM_002	Upgrade operation failed	For SRE: Check user logs for cause of operation failure (insufficient disk and/or memory). • If the cause of failure is insufficient disk and/or memory space, direct the tenant to free disk and/or memory. • If not, reinstall the Cloud Connector.

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Log Analysis

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

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

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

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

Syntax:

`<kubectl-command-parameters>`

- To list all the pods and their status, use the following syntax:`./deps/tools/k3s kubectl get pods -A`
- To delete a pod and then restart it, use the following syntax:`./deps/tools/k3s kubectl delete pods <pod-id> -n <name-space>`
- To describe the pods, use the following syntax: `./deps/tools/k3s kubectl describe pods <pod-id> -n <name-space>`
- To check the logs via `kubectl` command for k3s related pods, use the following syntax:`./deps/tools/k3s kubectl logs <pod-id> -n kube-system`



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



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

Steps to check pod status

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

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

Expected result:

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

Pod Description:

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

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

Steps to restart pods

Restart specific pod:

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

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

Restart both starter and platform pods:

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

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

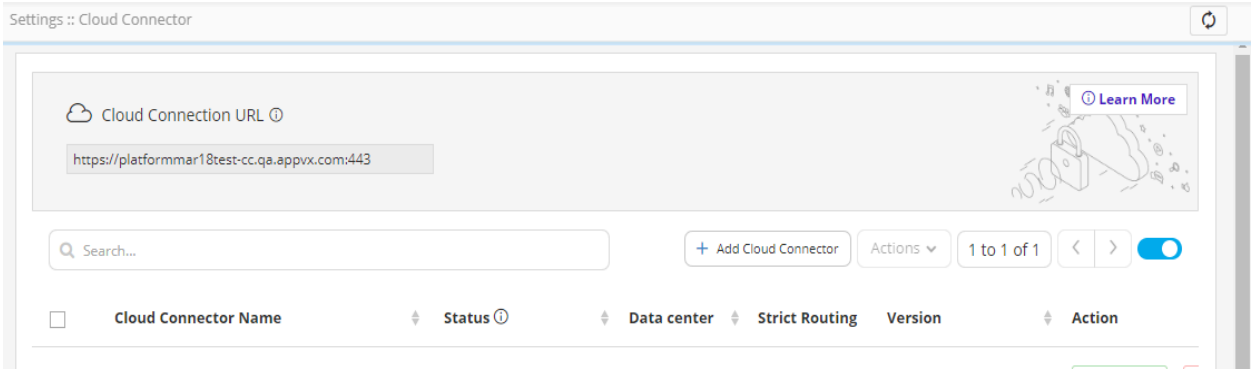
Managing the AppViewX Cloud Connector

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

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

Understanding the AppViewX Cloud Connector Inventory

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



Elements and Fields in the AppViewX Cloud Connector Inventory

Element/Field	Description
Cloud Connection URL	AppViewX cloud URL of the server (internal) that hosts the AppViewX instance of the AppViewX Cloud Connector
Learn More (in the banner)	Displays a quick introduction to the AppViewX Cloud Connector using a graphical representation of how the infrastructure

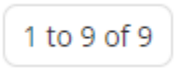
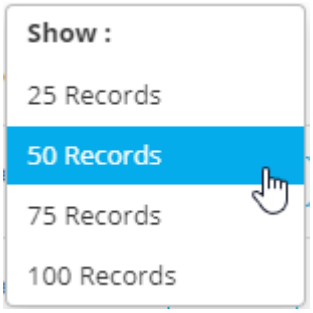
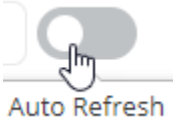
Copyright © 2025 AppViewX Inc. All Rights Reserved.

Elements and Fields in the AppViewX Cloud Connector Inventory (continued)

Element/Field	Description
	is deployed and works. The diagram is followed by a brief description of the key terms related to the cloud connector setup.
Search	 To search for a AppViewX Cloud Connector entry: 1. In the **Search** field, enter the value you want to filter the records for. 2. Press **Enter**. The **Settings :: Cloud Connector** page is updated to show details of only those records that match the search criteria.
Add Cloud Connector	To add a new AppViewX Cloud Connector, click  . **Note:** Refer to steps on adding a cloud connector.
Actions	**Note:** This button is enabled only when one or multiple AppViewX Cloud Connectors are selected. AppViewX lets you perform the following actions on a AppViewX Cloud Connector: • Start • Pause • Upgrade • Update config • Delete To perform these actions, click  .

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Elements and Fields in the AppViewX Cloud Connector Inventory (continued)

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Elements and Fields in the AppViewX Cloud Connector Inventory (continued)

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

key.

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

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

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

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

/ 

buttons from the Action field.

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Elements and Fields in the AppViewX Cloud Connector Inventory (continued)

<table>
<tr><th>Element/Field</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Paused</td><td>The AppViewX Cloud Connector has been approved by the admin but is paused.

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

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

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

This field displays an action that can be performed for the AppViewX Cloud Connector, depending on the current status of the AppViewX Cloud Connector.</td></tr>
</table>

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Elements and Fields in the AppViewX Cloud Connector Inventory (continued)

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

AppViewX Cloud Connector Actions

Key actions that can be performed include:

- Starting the AppViewX Cloud Connector
- Pausing the AppViewX Cloud Connector
- Upgrading the AppViewX Cloud Connector Version
- Updating the Certificate Configuration

Copyright © 2025 AppViewX Inc. All Rights Reserved.

• Deleting an AppViewX Cloud Connector Instance
• Uninstalling an AppViewX Cloud Connector Instance

Starting the AppViewX Cloud Connector

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

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

1. Navigate to the AppViewX Cloud Connector inventory.
2. Select the checkbox for the AppViewX Cloud Connector you want to start.
3. Click 

.
4. From the menu displayed, select **Start**.
   The AppViewX Cloud Connector Status is set to **Running**.

Pausing the AppViewX Cloud Connector

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

To pause the AppViewX Cloud Connector:

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

.
4. From the menu displayed, select **Pause**.
   The AppViewX Cloud Connector Status is set to **Paused**.

Upgrading the AppViewX Cloud Connector Version

AppViewX provides a seamless CI/CD pipeline to capture the AppViewX Cloud Connector versioning and upgrades on the release portal. If a new version of the AppViewX Cloud Connector component is

Copyright © 2025 AppViewX Inc. All Rights Reserved.

available, you will receive a notification with the name of the cloud connector that is ready for upgrade and the Upgrade button will be displayed for that AppViewX Cloud Connector.

1. Navigate to the AppViewX Cloud Connector inventory.
2. Select the checkbox for the AppViewX Cloud Connector you want to upgrade.
3. Click Actions.
4. From the menu displayed, select **Upgrade**.
   The **Confirmation message** dialog box is displayed. It shows the current version as well as the upgraded version number, and the checksum of the binaries that will be uploaded.
5. (Optional) To view the checksum of the binaries that will be uploaded as part of the cloud connector's version upgrade, click **Checksum of the binaries uploaded**.
6. (Optional) Validate the binaries.
7. Click**Upgrade**.
   - At the start of the upgrade, a prerequisite check will be triggered. For issues detected as a result of these checks, refer to the section Appendix: Internal Prerequisite Checks for CC Upgrade for further details.
   - After the prerequiste checks run successfully, the **Upgrade in progress** status is displayed and a notification informing that the upgrade is in progress is received.
   - If the AppViewX Cloud Connector version is upgraded successfully, the symbol is prefixed to the version number. You will receive an upgrade successful notification.
   - If the upgrade fails, you will receive an upgrade failure notification. The **Upgrade** button is displayed again.
     - When an upgrade fails, internally, AppViewX attempts to rollback to the previous successful version of the AppViewX Cloud Connector. When the rollback is successful, the **Upgrade** button is displayed again so that you can retry the version upgrade.

       OR

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

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Appendix: Internal Prerequisite Checks for CC Upgrade

As part of the Cloud Connector upgrade enhancement, additional pre-checks have been added to the process to ensure system readiness before the upgrade:

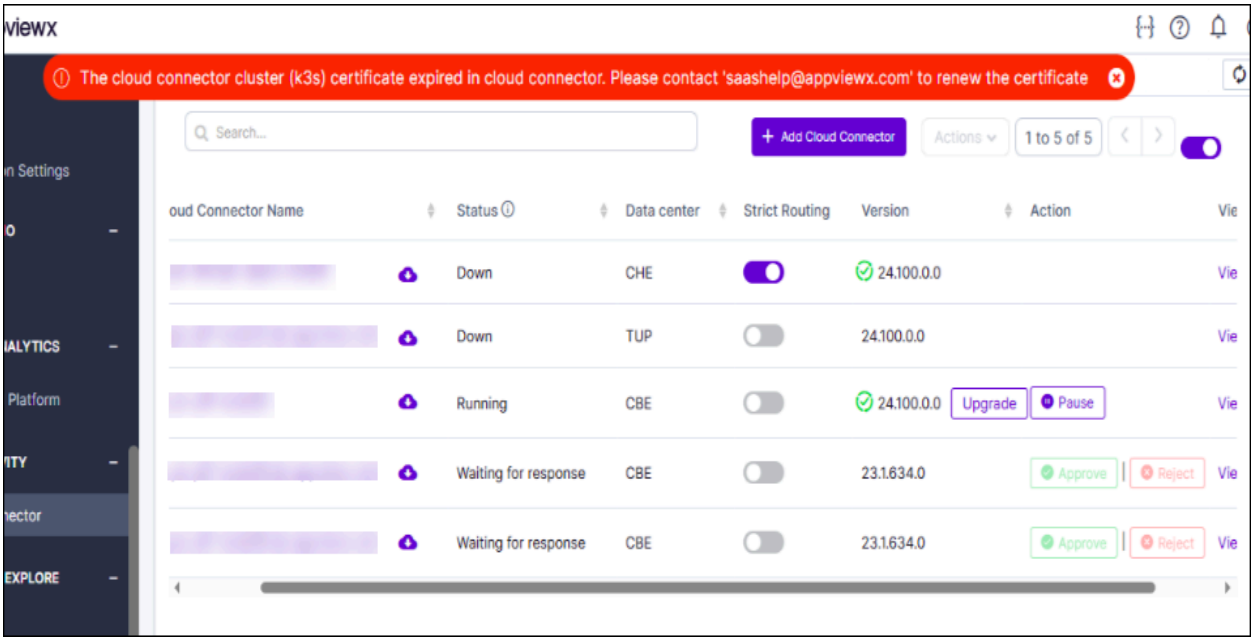
- Kubernetes certificate expiry check
- Free memory check
- Disk space check

Kubernetes Certificate Expiry Check and Renewal

During the Cloud Connector (CC) upgrade process, an expired K3s Kubernetes certificate can lead to unauthorized errors when executing K3s commands, resulting in upgrade failures. The pre-check mechanism validates the K3s certificate for expiry before initiating the upgrade thus preventing upgrade failures caused by expired certificates.

If the K3s certificate is found to be expired during the pre-check, the system will:

- Display a clear error message
- Provide a reference to the **SRE Destro**, where customers can reach out for support to renew the certificate manually



Automated Certificate Renewal (For v2024.3.0.0 and above)

Copyright © 2025 AppViewX Inc. All Rights Reserved.

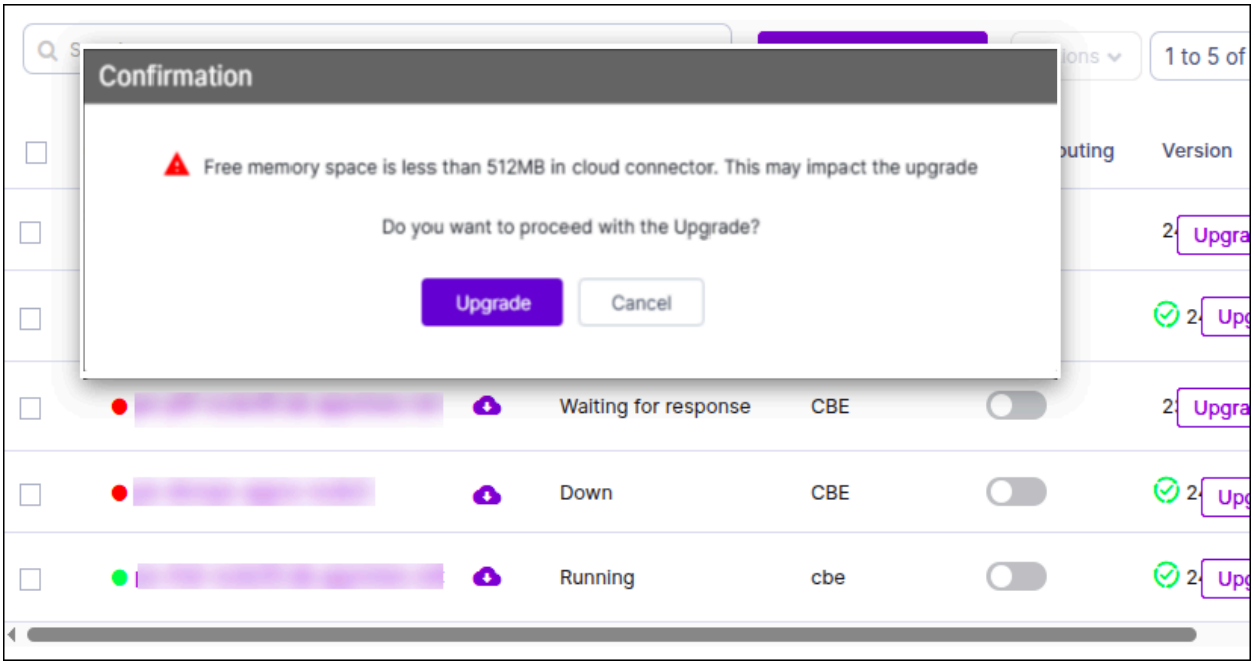
Beginning with v2024.3.0.0, customers can automatically renew an expired K3s Kubernetes certificate using the **avxctl** utility. The command is as follows:

```
./avxctl rotate k3s-cert
```

- This command must be executed on the Cloud Connector (CC) machine within the CC installation folder.
- The **avxctl** utility is available starting from v2024.3.0.0. Hence, based on the upgrade scenario:
  - **Upgrading to a version lower than v2024.3.0.0:** If the K3s certificate is already expired, it must be renewed manually before the upgrade.
  - **Upgrading from v2024.3.0.0 or above:** The **avxctl** utility will already be available, allowing customers to use it directly for certificate renewal if required.

Free Memory Check

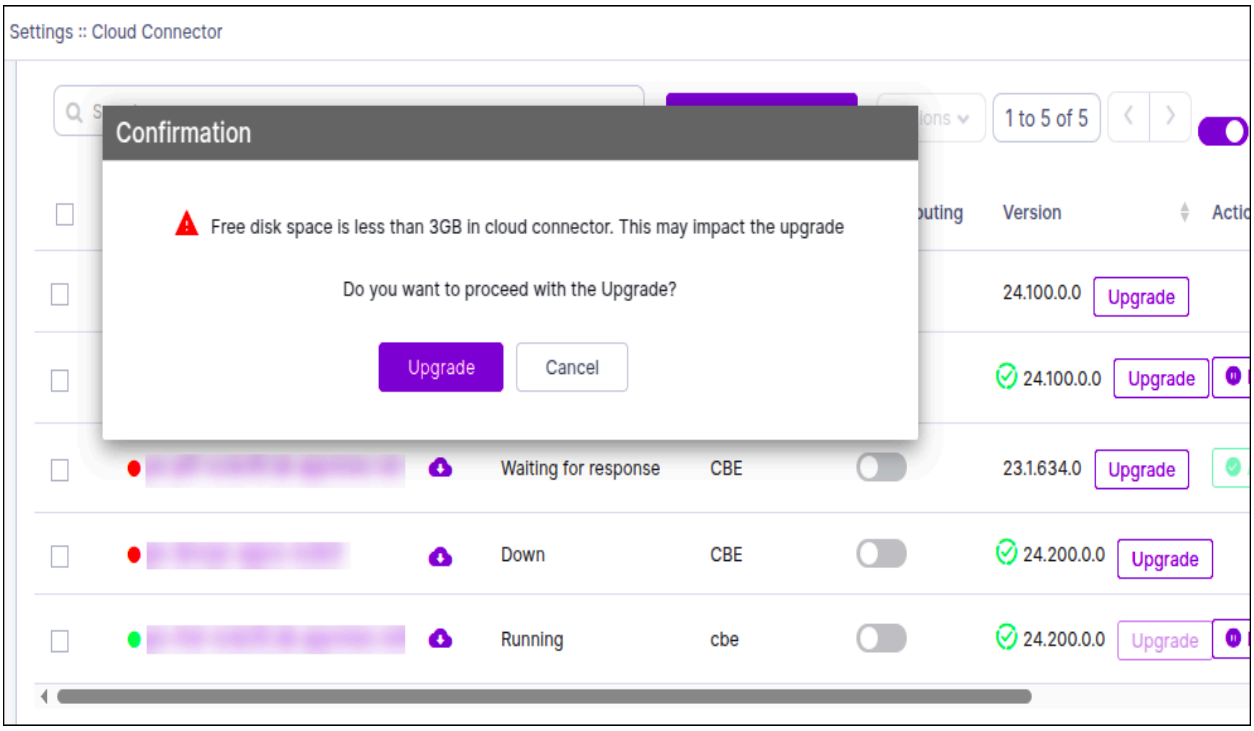
If the free memory is less than **512MB**, the pop-up with message is displayed as shown below. Users can however proceed to upgrade even if the checks fail.



Disk Space Check

If the disk space is less than **3GB**, the pop-up with message is displayed as shown below. Users can however proceed to upgrade even if the checks fail.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



Updating the Certificate Configuration

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

1. Select the checkbox for the required AppViewX Cloud Connector.
2. Click Actions.
3. From the menu displayed, select **Update Config**.
   On successful update of the certificate configuration, the message **Update config triggered successfully** is displayed.

Deleting an AppViewX Cloud Connector Instance

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

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

1. Select the checkbox for the AppViewX Cloud Connector you want to delete.
2. Click Actions.
3. From the menu displayed, select **Delete**.
4. In the **Confirmation message** dialog box, click **Delete**.
   The selected AppViewX Cloud Connector is deleted.

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

Uninstalling an AppViewX Cloud Connector Instance

To uninstall a AppViewX Cloud Connector instance:

1. On the node where the AppViewX Cloud Connector agent is installed, run the **uninstall.sh** script (included in the AppViewX Cloud Connector agent's download package).
2. For uninstallation on RHEL8+, when prompted, enter the sudo password.
   The AppViewX Cloud Connector instance is uninstalled.

Monitoring the Health of the AppViewX Cloud Connector

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

Descriptions for the color-coded health indicators

Color of the health indicator	Description
🟢	The AppViewX Cloud Connector is working as expected.
🟠	Although there are no current problems with routing traffic to and from the AppViewX Cloud Connector, the AppViewX Cloud Connector's health needs to be checked. To resolve, refer to the Troubleshooting section.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Descriptions for the color-coded health indicators (continued)

Color of the health indicator	Description
🔴	The AppViewX Cloud Connector is not receiving traffic. The AppViewX Cloud Connector's health is analyzed for 3 to 5 minutes before it is declared to be down. To resolve, refer to the Troubleshooting section.

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

Frequently Asked Questions

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

Disabling firewalld

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

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

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

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

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

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

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

Docker Prerequisites

Note: Since RHEL8+ does not include Docker support, Docker prerequisites are not applicable when the AppViewX Cloud Connector is being installed on a RHEL8+ node.

- Docker version 20.10.5, 27.1.1 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.

  Note: If **/var/lib** is going to be a separate mount, ensure that it has minimum 5 GB of free space.
  In case of restrictions in meeting this requirement, it is recommended to change the data root directory from **/var/lib** to another dedicated directory. For instructions on changing the data root directory, click here.

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

• Changing the Data Root Directory

Monitoring the Health of the AppViewX Cloud Connector

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

Descriptions for the color-coded health indicators

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

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

Steps to check pod status

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

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Expected result:

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

Pod Description:

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

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

Steps to restart pods

Restart specific pod:

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

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

Restart both starter and platform pods:

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

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

Deploying the AppViewX OVA

Copyright © 2025 AppViewX Inc. All Rights Reserved.

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

Detailed instructions for the OVA deployment are documented are here.

Detailed instructions for updating the AppViewX virtual image from the AppViewX repository are documented here.

Updating the AppViewX Virtual Image from the AppViewX Repository

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

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

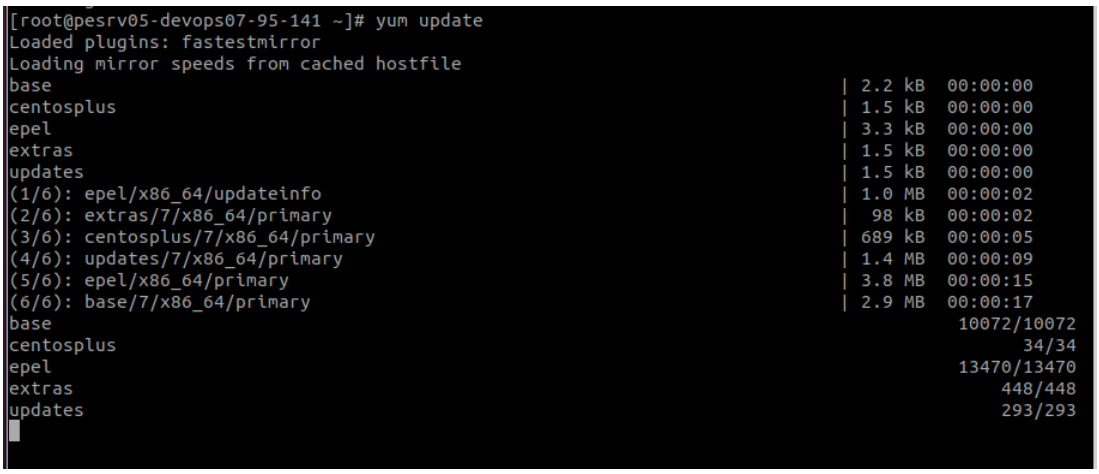
Prerequisites for Updating the Virtual Image from the Repository

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

Updating the Virtual Image from the Repository

Copyright © 2025 AppViewX Inc. All Rights Reserved.

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



Synchronizing the Node Clock with the Network Time

- On the node on which the AppViewX Cloud Connector is installed, ensure that the node's clock is synchronized with the network time using NTP/PTP.

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

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

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

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

    OR

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

Validating the SHA256 Checksum

Copyright © 2025 AppViewX Inc. All Rights Reserved.

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

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

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

Appendix A: Network Scan Recommendations

This section lists the AppViewX-recommended best practices for ensuring that network scans for cloud connectors yield more accurate results, thus facilitating a more secure and compliant network infrastructure and appropriate measures for mitigating potential risks.

Selecting Ports for Scanning

- Commonly, the ports **443** and **8443** are used for scanning. Additional ports identified at the time of your onboarding can be added to the list.
- If a list of specific ports cannot be identified, a standard port scan is the next best recommendation.

> **Note:** Research suggests that 99% of open ports can be identified by scanning the top 3328 ports. For more information, click here. For the complete list of standard ports that are scanned by AppViewX, click here.

- While the all ports scan can also be used, it can be time consuming and add a significant load to the infrastructure. AppViewX recommends a lesser number of ports, so that the scan time is less and the process is more optimally completed.
- When performing larger subnet scans, for example for a /16 subnet, for throttle scanning, split larger subnets into smaller batches. For example, split the /16 subnet into its /24 equivalent.

Configuring the AppViewX Cloud Connectors' Infrastructure

- For production environments, it is recommended to have two cloud connectors per datacenter. This enables high availability within a datacenter as well as across all datacenters in the environment.
- Enable strict routing for the cloud connectors within the same datacenter so traffic can be optimally routed between the cloud connectors.
- For scanning more than a 100 subnets within a span of 24 hours, allocate additional computing resources by provisioning one cloud connector for every 100 subnets (so there'll be a total of 1000 IPs across the subnet).

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Setting Batch Limits for Network Discovery

For network discovery, 10K is the maximum recommended batch limit.

Setting the Scanning Intensity

During a network scan, the AppViewX Network Plugin sends the number of packets to the IP address configured on the network scan. The load on the target network can be controlled by selecting a scanning intensity from the range 1 to 12.

Scanning intensity 1 to 4

Scanning intensities between 1 and 4 are designed to scan **less than 250 ports** or, for larger networks, common SSL ports like **443**, **8443**.

For a larger network, these intensities can take up to several days to complete scanning, especially if a **all ports** scan is triggered.

Scanning intensity 5 to 12

Scanning intensities between 5 and 12 are designed for scanning **standard ports** and **all ports**. For these higher intensities, the number of network connections increases, which then decreases the time required for scanning.

Tip: A scanning intensity in the range **4 to 6** is known to be appropriately reliable and accurate, and consumes very less bandwidth.

Warning: A scan intensity in the range **8 to 12** is known to establish high packet transmission and a higher number of connections per second. For example, **setting intensity = 12** will establish **16K connections/second**. Setting the scanning intensity to a value in this range is not recommended unless your network team confirms that your network infrastructure can handle the number of connections established.

Optimizing the Load Factor

For handling increased loads, it is preferable to adopt horizontal scaling by adding more cloud connectors. The requests are then handled using the round robin allocation method across all the available cloud connectors.

Example: For a load of 140K subnets, it is recommended to add one cloud connector for scanning a set of 17.5K subnets. Since a standard ports scan and a all ports scan yield nearly identical results, and

Copyright © 2025 AppViewX Inc. All Rights Reserved.

a standard port scan is 7x faster, it is proposed to run the scan in two phases: **phase 1** will cover the standard ports and **phase 2** will cover all ports.

Cloud connector distribution across datacenters will be decided based on your IP distribution across those datacenters.

To reduce network latency, more datacenters (and cloud connectors) should be added based on your subnet topology.

Appendix B: Automated Installation without Internet

After you have entered the tenant ID, if the host machine is not connected to the internet, the following error message is displayed: **The AppViewX Cloud URL is unreachable.** followed by the prompt: **Do you want to configure the proxy? (y/n).**

To proceed with the installation:

1. When prompted **Do you want to configure the proxy? (y/n)**:, enter **y**.
2. Enter the proxy details as prompted.

Note: If you enter **n**, the automated installation of the cloud connector is skipped. You can choose to install the cloud connector using the user interface.

Appendix C: CIS Benchmarking for AppViewX Cloud Connector

This report is specific to the v1.27-v1.29 release line of K3s and the v1.8 release of the CIS Kubernetes Benchmark.

Note:

- **PASS** denotes that the controls conform to the standards.
- **WARN/FAIL** denotes that the specific control does not conform to the standard.
- **NA** is not applicable for k3s or cloud connector.

Overall Compliance	65.91%

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Summary

Policy	PASS/NA	WARN/FAIL
Count	87	45

Controls	Status	Comment
Overview		
Testing controls methodology		
1.1 Control Plane Node Configuration Files		
1.1.1 Ensure that the API server pod specification file permissions are set to 600 or more restrictive (Automated)	PASS	
1.1.2 Ensure that the API server pod specification file ownership is set to root (Automated)	NA	Not applicable for k3s.
1.1.3 Ensure that the controller manager pod specification file permissions are set to 600 or more restrictive (Automated)	NA	Not applicable for k3s.
1.1.4 Ensure that the controller manager pod specification file ownership is set to root (Automated)	NA	Not applicable for k3s.
1.1.5 Ensure that the scheduler pod specification file permissions are set to 600 or more restrictive (Automated)	NA	Not applicable for k3s.
1.1.6 Ensure that the scheduler pod specification file ownership is set to root (Automated)	NA	Not applicable for k3s.
1.1.7 Ensure that the etcd pod specification file permissions are set to 600 or more restrictive (Automated)	NA	Not applicable for k3s.
1.1.8 Ensure that the etcd pod specification file ownership is set to root (Automated)	NA	Not applicable for k3s.
1.1.9 Ensure that the Container Network Interface file permissions are set to 600 or more restrictive (Automated)	NA	Not applicable for k3s.
1.1.10 Ensure that the Container Network Interface file ownership is set to root (Manual)	NA	Not applicable for k3s.
1.1.11 Ensure that the etcd data directory permissions are set to 700 or more restrictive (Automated)	PASS	

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Controls	Status	Comment
1.1.12 Ensure that the etcd data directory ownership is set to etcd (Automated)	NA	Not applicable for k3s.
1.1.13 Ensure that the admin.conf file permissions are set to 600 or more restrictive (Automated)	NA	Not applicable for k3s.
1.1.14 Ensure that the admin.conf file ownership is set to root (Automated)	PASS	
1.1.15 Ensure that the scheduler.conf file permissions are set to 600 or more restrictive (Automated)	PASS	
1.1.16 Ensure that the scheduler.conf file ownership is set to root (Automated)	PASS	
1.1.17 Ensure that the controller-manager.conf file permissions are set to 600 or more restrictive (Automated)	PASS	
1.1.18 Ensure that the controller-manager.conf file ownership is set to root (Automated)	PASS	
1.1.19 Ensure that the Kubernetes PKI directory and file ownership is set to root (Automated)	PASS	
1.1.20 Ensure that the Kubernetes PKI certificate file permissions are set to 600 or more restrictive (Manual)	FAIL	Currently permissions are set to 644.
1.1.21 Ensure that the Kubernetes PKI key file permissions are set to 600 (Manual)	PASS	
1.2 API Server	PASS	
1.2.1 Ensure that the --anonymous-auth argument is set to false (Manual)	PASS	
1.2.2 Ensure that the --token-auth-file parameter is not set (Automated)	PASS	
1.2.3 Ensure that the --DenyServiceExternalIPs is not set (Automated)	PASS	
1.2.4 Ensure that the --kubelet-client-certificate and --kubelet-client-key arguments are set as appropriate (Automated)	PASS	
1.2.5 Ensure that the --kubelet-certificate-authority argument is set as appropriate (Automated)	NA	Not applicable for k3s.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Controls	Status	Comment
1.2.6 Ensure that the --authorization-mode argument is not set to AlwaysAllow (Automated)	PASS	
1.2.7 Ensure that the --authorization-mode argument includes Node (Automated)	PASS	
1.2.8 Ensure that the --authorization-mode argument includes RBAC (Automated)	PASS	
1.2.9 Ensure that the admission control plugin EventRateLimit is set (Manual)	FAIL	Not supported.
1.2.10 Ensure that the admission control plugin AlwaysAdmit is not set (Automated)	PASS	
1.2.11 Ensure that the admission control plugin AlwaysPullImages is set (Manual)	FAIL	The cloud connector does not directly import images from external registries; instead, it imports all images into the local registry in the node before consumption.
1.2.12 Ensure that the admission control plugin SecurityContextDeny is set if PodSecurityPolicy is not used (Manual)	NA	Not applicable for k3s.
1.2.13 Ensure that the admission control plugin ServiceAccount is set (Automated)	PASS	
1.2.14 Ensure that the admission control plugin NamespaceLifecycle is set (Automated)	PASS	
1.2.15 Ensure that the admission control plugin NodeRestriction is set (Automated)	PASS	
1.2.16 Ensure that the --profiling argument is set to false (Automated)	PASS	
1.2.17 Ensure that the --audit-log-path argument is set (Automated)	NA	Not applicable for k3s.
1.2.18 Ensure that the --audit-log-maxage argument is set to 30 or as appropriate (Automated)	NA	Not applicable for k3s.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Controls	Status	Comment
1.2.19 Ensure that the --audit-log-maxbackup argument is set to 10 or as appropriate (Automated)	NA	Not applicable for k3s.
1.2.20 Ensure that the --audit-log-maxsize argument is set to 100 or as appropriate (Automated)	NA	Not applicable for .
1.2.21 Ensure that the --request-timeout argument is set as appropriate (Manual)	NA	Not applicable for k3s.
1.2.22 Ensure that the --service-account-lookup argument is set to true (Automated)	PASS	
1.2.23 Ensure that the --service-account-key-file argument is set as appropriate (Automated)	NA	Not applicable for k3s.
1.2.24 Ensure that the --etcd-certfile and --etcd-keyfile arguments are set as appropriate (Automated)	FAIL	The etcd is not used, instead SQLite is used for cloud connector deployment. Not applicable for cloud connector deployment.
1.2.25 Ensure that the --tls-cert-file and --tls-private-key-file arguments are set as appropriate (Automated)	PASS	
1.2.26 Ensure that the --client-ca-file argument is set as appropriate (Automated)	PASS	
1.2.27 Ensure that the --etcd-cafile argument is set as appropriate (Automated)	FAIL	The etcd is not used, instead SQLite is used for cloud connector deployment. Not applicable for cloud connector deployment.
1.2.28 Ensure that the --encryption-provider-config argument is set as appropriate (Manual)	NA	Not applicable for k3s.
1.2.29 Ensure that encryption providers are appropriately configured (Manual)	NA	Not applicable for k3s.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Controls	Status	Comment
1.2.30 Ensure that the API Server only makes use of Strong Cryptographic Ciphers (Manual)	PASS	
1.3 Controller Manager		
1.3.1 Ensure that the --terminated-pod-gc-threshold argument is set as appropriate (Manual)	FAIL	Not supported.
1.3.2 Ensure that the --profiling argument is set to false (Automated)	PASS	
1.3.3 Ensure that the --use-service-account-credentials argument is set to true (Automated)	PASS	
1.3.4 Ensure that the --service-account-private-key-file argument is set as appropriate (Automated)	PASS	
1.3.5 Ensure that the --root-ca-file argument is set as appropriate (Automated)	PASS	
1.3.6 Ensure that the RotateKubeletServerCertificate argument is set to true (Automated)	NA	Not applicable for k3s.
1.3.7 Ensure that the --bind-address argument is set to 127.0.0.1 (Automated)	PASS	
1.4 Scheduler	PASS	
1.4.1 Ensure that the --profiling argument is set to false (Automated)	PASS	
1.4.2 Ensure that the --bind-address argument is set to 127.0.0.1 (Automated)	PASS	
2 Etcd Node Configuration		
2.1 Ensure that the --cert-file and --key-file arguments are set as appropriate (Automated)	FAIL	The etcd is not used, instead sqllite is used for cloud connector deployment. Not applicable for cloud connector deployment.
2.2 Ensure that the --client-cert-auth argument is set to true (Automated)	FAIL	The etcd is not used, instead sqllite is used for cloud connector

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Controls	Status	Comment
		deployment. Not applicable for cloud connector deployment.
2.3 Ensure that the --auto-tls argument is not set to true (Automated)	FAIL	The etcd is not used, instead sqllite is used for cloud connector deployment. Not applicable for cloud connector deployment.
2.4 Ensure that the --peer-cert-file and --peer-key-file arguments are set as appropriate (Automated)	FAIL	The etcd is not used, instead sqllite is used for cloud connector deployment. Not applicable for cloud connector deployment.
2.5 Ensure that the --peer-client-cert-auth argument is set to true (Automated)	FAIL	The etcd is not used, instead sqllite is used for cloud connector deployment. Not applicable for cloud connector deployment.
2.6 Ensure that the --peer-auto-tls argument is not set to true (Automated)	FAIL	The etcd is not used, instead sqllite is used for cloud connector deployment. Not applicable for cloud connector deployment.
2.7 Ensure that a unique Certificate Authority is used for etcd (Automated)	FAIL	The etcd is not used, instead sqllite is used for cloud connector deployment. Not applicable for cloud connector deployment.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Controls	Status	Comment
3 Control Plane Configuration		
3.1 Authentication and Authorization		
3.1.1 Client certificate authentication should not be used for users (Manual)	FAIL	Since the cloud connector is a self-installed agent and upgrades are managed over the air, no external user access is required.
3.1.2 Service account token authentication should not be used for users (Manual)	FAIL	Since the cloud connector is a self-installed agent and upgrades are managed over the air, no external user access is required.
3.1.3 Bootstrap token authentication should not be used for users (Manual)	PASS	
3.2 Logging		
3.2.1 Ensure that a minimal audit policy is created (Manual)	PASS	Audit policy is enabled (warn).
3.2.2 Ensure that the audit policy covers key security concerns (Manual)	FAIL	Not supported.
4.1 Worker Node Configuration Files		
4.1.1 Ensure that the kubelet service file permissions are set to 600 or more restrictive (Automated)	NA	Not applicable for k3s.
4.1.2 Ensure that the kubelet service file ownership is set to root (Automated)	NA	Not applicable for k3s.
4.1.3 If proxy kubeconfig file exists ensure permissions are set to 600 or more restrictive (Manual)	PASS	

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Controls	Status	Comment
4.1.4 If proxy kubeconfig file exists ensure ownership is set to root (Manual)	PASS	
4.1.5 Ensure that the --kubeconfig kubelet.conf file permissions are set to 600 or more restrictive (Automated)	PASS	
4.1.6 Ensure that the --kubeconfig kubelet.conf file ownership is set to root (Automated)	PASS	
4.1.7 Ensure that the certificate authorities file permissions are set to 600 or more restrictive (Manual)	PASS	
4.1.8 Ensure that the client certificate authorities file ownership is set to root (Manual)	PASS	
4.1.9 Ensure that the kubelet --config configuration file has permissions set to 600 or more restrictive (Automated)	PASS	
4.1.10 Ensure that the kubelet --config configuration file ownership is set to root (Automated)	PASS	
4.2 Kubelet		
4.2.1 Ensure that the --anonymous-auth argument is set to false (Automated)	PASS	
4.2.2 Ensure that the --authorization-mode argument is not set to AlwaysAllow (Automated)	PASS	
4.2.3 Ensure that the --client-ca-file argument is set as appropriate (Automated)	PASS	
4.2.4 Verify that the --read-only-port argument is set to 0 (Manual)	PASS	
4.2.5 Ensure that the --streaming-connection-idle-timeout argument is not set to 0 (Manual)	PASS	
4.2.6 Ensure that the --make-iptables-util-chains argument is set to true (Automated)	PASS	
4.2.7 Ensure that the --hostname-override argument is not set (Manual)	NA	Not applicable for k3s.
4.2.8 Ensure that the eventRecordQPS argument is set to a level which ensures appropriate event capture (Manual)	PASS	

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Controls	Status	Comment
4.2.9 Ensure that the --tls-cert-file and --tls-private-key-file arguments are set as appropriate (Manual)	NA	Not applicable for k3s.
4.2.10 Ensure that the --rotate-certificates argument is not set to false (Manual)	PASS	
4.2.11 Verify that the RotateKubeletServerCertificate argument is set to true (Manual)	FAIL	Does not affect the operation of the cloud connector, as external kube API access is required for any host level access for the cluster alone.
4.2.12 Ensure that the Kubelet only makes use of Strong Cryptographic Ciphers (Manual)	FAIL	The cloud connector runs on a single host machine.
4.2.13 Ensure that a limit is set on pod PIDs (Manual)	FAIL	Will be implemented in the upcoming release.
5.1 RBAC and Service Accounts		
5.1.1 Ensure that the cluster-admin role is only used where required (Manual)	FAIL	Since the cloud connector is a self-installed agent and upgrades are managed over the air, these permissions must be maintained for installation and upgrade functionality.
5.1.2 Minimize access to secrets (Manual)	FAIL	Since the cloud connector is a self-installed agent and upgrades are managed over the air, these permissions must be maintained

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Controls	Status	Comment
		for installation and upgrade functionality. Additionally CC doesn't have multiple users.
5.1.3 Minimize wildcard use in Roles and ClusterRoles (Manual)	PASS	
5.1.4 Minimize access to create pods (Manual)	FAIL	Since the cloud connector is a self-installed agent and upgrades are managed over the air, these permissions must be maintained for installation and upgrade functionality. Additionally CC doesn't have multiple users.
5.1.5 Ensure that default service accounts are not actively used. (Manual)	PASS	Separate service accounts are created for the cc namespace.
5.1.6 Ensure that Service Account Tokens are only mounted where necessary (Manual)	FAIL	Not supported.
5.1.7 Avoid use of system group (Manual)	FAIL	Since its a default feature, its not removed as of now
5.1.8 Limit use of the Bind, Impersonate and Escalate permissions in the Kubernetes cluster (Manual)	FAIL	The cluster access to be restricted by creating a less permissive user account.
5.1.9 Minimize access to create persistent volumes (Manual)	FAIL	Cloud connector uses persistent volumes

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Controls	Status	Comment
		for storing logs and maintaining other external libraries such iControl jar. The new pods should have the capability to support upgrade from GUI.
5.1.10 Minimize access to the proxy sub-resource of nodes (Manual)	FAIL	Not supported.
5.1.11 Minimize access to the approval sub-resource of certificatesigningrequests objects (Manual)	FAIL	This is not relevant since the cloud connector operates exclusively on Linux-based systems.
5.1.12 Minimize access to webhook configuration objects (Manual)	FAIL	Not supported.
5.1.13 Minimize access to the service account token creation (Manual)	FAIL	Not supported.
5.2 Pod Security Standards		
5.2.1 Ensure that the cluster has at least one active policy control mechanism in place (Manual)	PASS	Addressed as part of FP3.1 release*
5.2.2 Minimize the admission of privileged containers (Manual)	FAIL	Baseline policy will be enforced in FP3.1 (Tentative), restricted policy not supported*
5.2.3 Minimize the admission of containers wishing to share the host process ID namespace (Automated)	FAIL	Baseline policy will be enforced in FP3.1 (Tentative), restricted policy not supported*
5.2.4 Minimize the admission of containers wishing to share the host IPC namespace (Automated)	FAIL	Baseline policy will be enforced in FP3.1 (Tentative), restricted policy not supported*
5.2.5 Minimize the admission of containers wishing to share the host network namespace (Automated)	FAIL	Cloud connector uses host network for

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Controls	Status	Comment
		communicating outside with the devices.
5.2.6 Minimize the admission of containers with allowPrivilegeEscalation (Automated)	FAIL	avx-mid-server-platform pod will have allowPriviegeEscalation as true due to nmap command execution during network discovery use cases.
5.2.7 Minimize the admission of root containers (Automated)	FAIL	All the business pods in the cc namespace will be running as non-root. The pods are running using the host user id who installed the CC (avxctl refresh all to be fired). The log clean up cronjob will be running as root to support backward compatibility.
5.2.8 Minimize the admission of containers with the NET_RAW capability (Automated)	FAIL	Not supported.
5.2.9 Minimize the admission of containers with added capabilities (Automated)	FAIL	Not supported.
5.2.10 Minimize the admission of containers with capabilities assigned (Manual)	FAIL	Not supported.
5.2.11 Minimize the admission of Windows HostProcess containers (Manual)	FAIL	Not relevant since the cloud connector operates exclusively on Linux-based systems.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Controls	Status	Comment
5.2.12 Minimize the admission of HostPath volumes (Manual)	FAIL	The cloud connector requires mounting files from the host machine, so hostPath volumes must be allowed.
5.2.13 Minimize the admission of containers which use HostPorts (Manual)	FAIL	The cloud connector uses host ports to expose services like IoT. Host ports admission is required.
5.3 Network Policies and CNI		
5.3.1 Ensure that the CNI in use supports NetworkPolicies (Manual)	PASS	Network policies has been added for cc namespace.
5.3.2 Ensure that all Namespaces have NetworkPolicies defined (Manual)	FAIL	Network policies will be available as part of FP3.1 release for the cc namespace*
5.4 Secrets Management		
5.4.1 Prefer using Secrets as files over Secrets as environment variables (Manual)	PASS	No business specific secrets for the application are created.
5.4.2 Consider external secret storage (Manual)	FAIL	The cloud connector is a light weight agent installed on a single host. So it uses the default secret management system provided by k3s.
5.5 Extensible Admission Control		

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Controls	Status	Comment
5.5.1 Configure Image Provenance using ImagePolicyWebhook admission controller (Manual)	FAIL	No supported.
5.7 General Policies		
5.7.1 Create administrative boundaries between resources using namespaces (Manual)	PASS	Separated namespace has been created
5.7.2 Ensure that the seccomp profile is set to docker/default in your Pod definitions (Manual)	PASS	Addressed as port FP3.1.
5.7.3 Apply SecurityContext to your Pods and Containers (Manual)	PASS	
5.7.4 The default namespace should not be used (Manual)	PASS	

The report is prepared base on the k3s self assement guide v1.8.

* policies will require the cloud connector reinstallation to be effective

Appendix D: Verifying and Renewing Kubernetes Certificates in the AppViewX Cloud Connector

Checking Certificate Expiry

1. Login to the user account in which the AppViewX Cloud Connector has been installed.
2. Navigate to the AppViewX Cloud Connector home directory (where the **install.sh** script is located.
3. To check for certificate expiry, execute the following commands:
   - `cat deps/utils/kubeconfig.yaml | grep "client-certificate-data" | awk '{print $2}' | base64 --decode | openssl x509 -text -noout | grep 'Not After'`
   - `cat ~/.kube/config | grep "client-certificate-data" | awk '{print $2}' | base64 --decode | openssl x509 -text -noout | grep 'Not After'`

   If either of the above commands returns a date that is prior to today's date, the corresponding certificate is expired.

Renewing the Kubernetes Certificates in the AppViewX Cloud Connector

1. To identify the deployment method, execute the following commands:

   ```
   grep -i K3S_INSTALL_MODE <deps_path>/properties/appviewx.properties
   ```

   The **deps** directory is located in the same folder as the installation script.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

If the **K3S_INSTALL_MODE** flag is present in the properties file, the above command will return one of the following two modes as the output: **k3d** or **standalone**.

> **Note:** The properties file will be located here: **<CC_home_directory>/deps/properties/appviewx.properties**.

2. > **Note:** Execute this step only if the **K3S_INSTALL_MODE** flag is not present or empty.

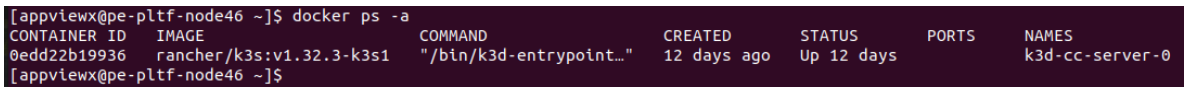
   Determine the installation mode of the AppViewX Cloud Connector and add it to the properties file.

   To do this:

   a. Execute the following command:

   ```
   docker ps -a
   ```

   If the output of this command returns a container named **k3d-cc-server-0**, as shown in the image below:

   
   ```
   [appviewx@pe-pltf-node46 ~]$ docker ps -a
   CONTAINER ID   IMAGE                      COMMAND                  CREATED       STATUS       PORTS     NAMES
   0edd22b19936   rancher/k3s:v1.32.3-k3s1   "/bin/k3d-entrypoint…"   12 days ago   Up 12 days             k3d-cc-server-0
   [appviewx@pe-pltf-node46 ~]$
   ```

   i. Modify the properties file to add the following: **K3S_INSTALL_MODE=k3d**

   If the output does not return the **k3d-cc-server-0** container:

   i. Execute the following command:

   ```
   systemctl status k3s
   ```

   ii. If the k3s server is running, modify the properties file to add the following: **K3S_INSTALL_MODE=standalone**.

3. If the AppViewX Cloud Connector version is **v24.3.0.0 or higher**:

   a. Navigate to the home directory of the AppViewX Cloud Connector (where the **install.sh** script file is located and execute the following command:

   ```
   ./avxctl rotate k3s-cert
   ```

   The command will automatically perform all operations needed to renew the expired Kubernetes certificate(s).

   If the AppViewX Cloud Connector version is **earlier than v24.3.0.0**, to renew the certificates, execute the following steps:

   a. Determine the installation mode of the AppViewX Cloud Connector.

   b. If the installation mode is **k3d**:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

i. Navigate to the AppViewX Cloud Connector home directory (where the **install.sh** script is located).

ii. Login to the Docker container.

```
docker exec -it k3d-cc-server-0 sh
```

iii. Rotate the certificate.

```
k3s certificate rotate
```

iv. Restart the Docker container.

```
sudo systemctl restart docker
```

v. For Ubuntu, update the nameserver in Coredns.

```
./deps/utils/update_nameserver_in_coredns.sh
```

vi. Backup the existing certificates.

```
mv "$(pwd)"/deps/utils/kubeconfig.yaml "$(pwd)"/deps/utils/bkp_kubeconfig.yaml

mv ~/.kube/config ~/.kube/bkp_config
```

vii. Renew the certificates.

```
deps/tools/k3d kubeconfig get cc >$(pwd)/deps/utils/kubeconfig.yaml

cp deps/utils/kubeconfig.yaml ~/.kube/config

chmod 600 ~/.kube/config

chmod 664 "$(pwd)"/deps/utils/kubeconfig.yaml
```

If the installation mode is **standalone**:

i. Navigate to the AppViewX Cloud Connector home directory (where the **install.sh** script is located).

ii. Restart the k3s service.

```
sudo systemctl restart k3s.service
```

iii. Check if the certificate has been renewed in the ***/etc*** location.

```
cat /etc/rancher/k3s/k3s.yaml | grep "client-certificate-data" | awk '{print $2}' | base64 --decode | openssl x509 -noout -enddate
```

iv. Copy the renewed certificate to the required locations and grant required permissions in the host machine.

```
cp /etc/rancher/k3s/k3s.yaml ~/.kube/config

cp /etc/rancher/k3s/k3s.yaml <deps_directory>/utils/kubeconfig.yaml

chmod 600 ~/.kube/config

chmod 664 "$(pwd)"/deps/utils/kubeconfig.yaml
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Post Certificate Renewal Checks

Once the certificate is renewed:

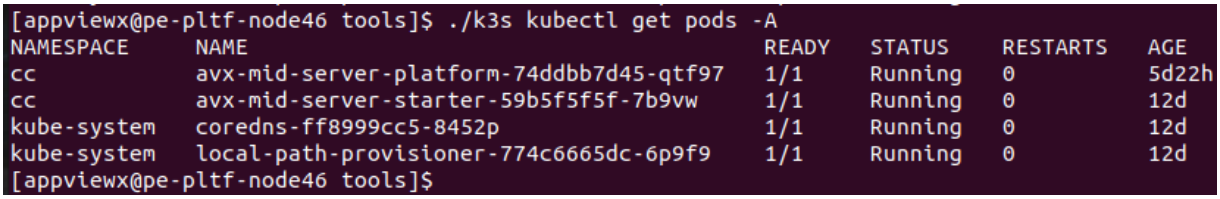
1. Navigate to the AppViewX Cloud Connector home directory (where the **install.sh** script is located).
2. To perform the post-renewal checks, execute the following commands:

```
cat deps/utils/kubeconfig.yaml | grep "client-certificate-data" | awk '{print $2}' | base64 --decode | openssl x509 -text -noout | grep 'Not After'

cat ~/.kube/config | grep "client-certificate-data" | awk '{print $2}' | base64 --decode | openssl x509 -text -noout | grep 'Not After'

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

The certificate will be renewed in both, the location and all the pods should be listed as shown in the image below:



Copyright © 2025 AppViewX Inc. All Rights Reserved.

Chapter 3: Managed Kubernetes

These Managed Kubernetes - Install and Upgrade Guides provides the prerequisites and the procedure for installing and accessing AppViewX on AKS, EKS, and GKE.

- AppViewX Install and Upgrade for AKS
- AppViewX Install and Upgrade for EKS
- AppViewX Install and Upgrade for GKE
- ManagedK8s Kubernetes Upgrade: Guidelines for Cluster Admins

AppViewX Install and Upgrade for AKS

This guide provides the prerequisites and the procedure for installing, upgrading, and accessing the AppViewX application.

- AppViewX Architecture
- Architecture Overview
- AppViewX Deployment Architecture
- Managed Kubernetes Architecture
- AKS Components
- Prerequisites
- Install AppViewX in Managed Kubernetes
- Upgrade AppViewX in Managed Kubernetes
- Downloading Images from AppViewX Repository
- Utility Tools and Functions
- Log Collection
- Uninstall and Cleanup
- Troubleshooting

AppViewX Architecture

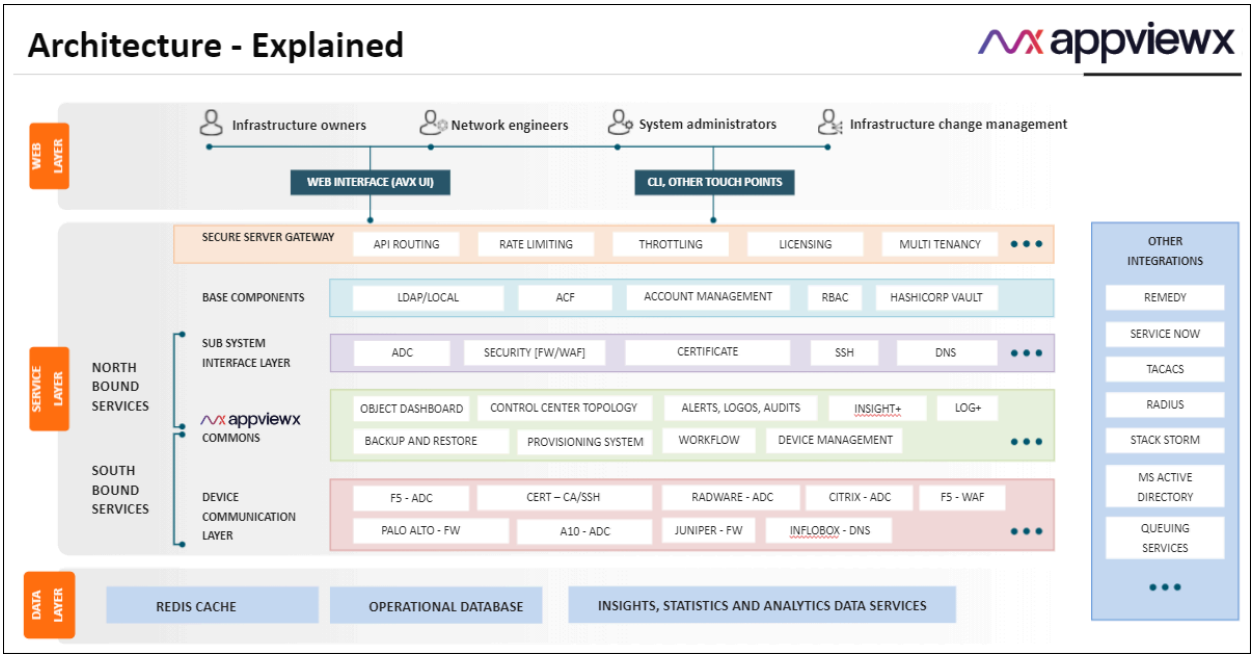
Architecture Explained

AppViewX is designed based on the microservice architecture and is deployed on Kubernetes—an open-source platform for deploying and managing containers.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

The microservice architecture of AppViewX makes it easier to move to containerized workloads and the containers being orchestrated using Kuberneters.

Kubernetes provides container runtime, orchestration, self-healing mechanisms, service discovery and load balancing and it is used for the deployment, scaling, management, and composition of application containers across clusters.



Benefits of AppViewX Architecture

In order to optimally utilize the resources, AppViewX has adopted Kubernetes to achieve higher security by adopting a zero trust network model. The features of AppViewX coupled with Kubernetes are given below.

- **Auto scaling** - AppViewX services can have a custom throttling capability based on pre-configured memory configuration per API. This will enable AppViewX services to utilize (scale up) resources optimally as the demand surges and scale down when not in use. This will help to horizontally scale the vendor components on demand and optimize the resource usage.
- **Resiliency** - There is no guarantee that AppViewX services may run without any interruptions and they are bound to fail. Kubernetes keeps deployments healthy by restarting containers that have failed,

Copyright © 2025 AppViewX Inc. All Rights Reserved.

by killing and replacing unresponsive containers based on health checks. This helps to mitigate the common pain point of the application's upkeep process.

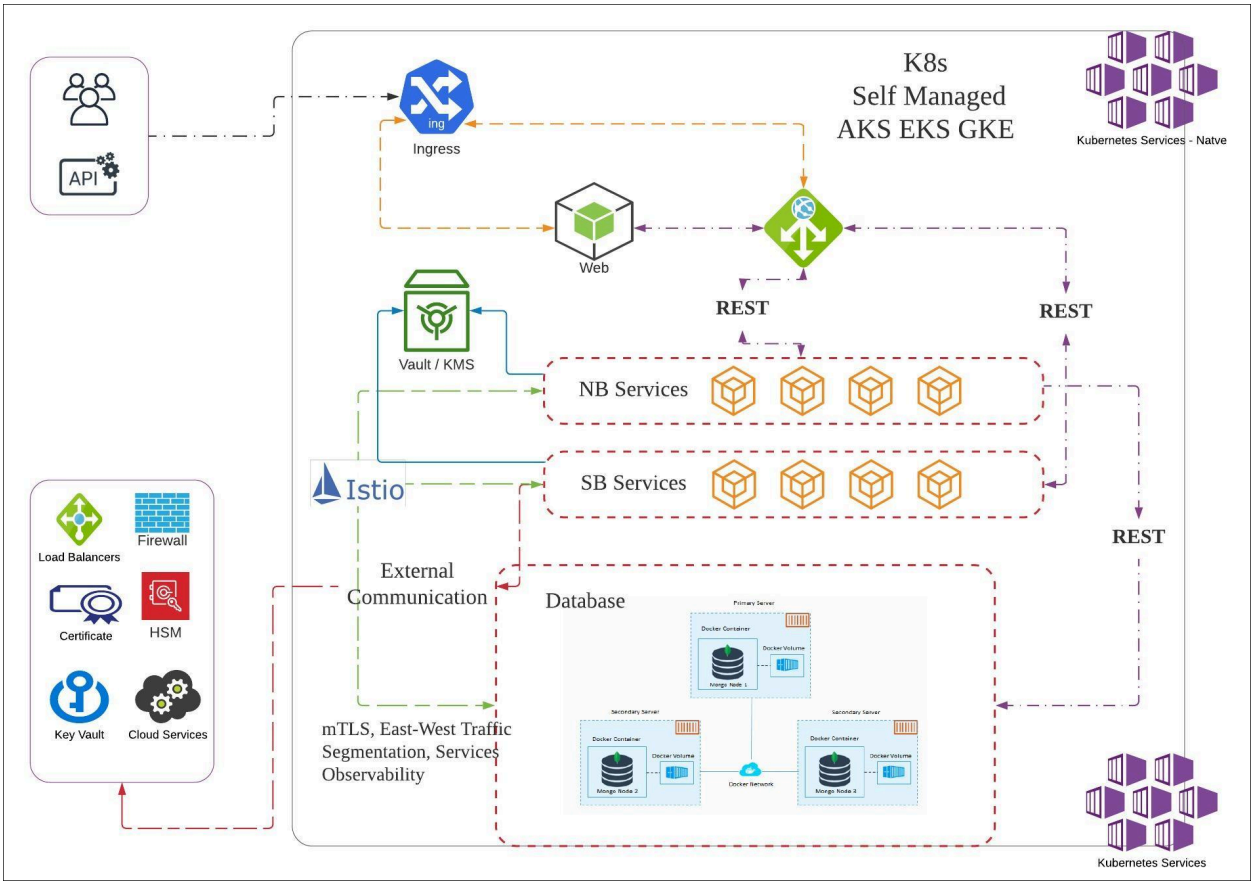
- **Security** - AppViewX architecture is designed around the concept of zero trust network model to enforce tighter security within the Kubernetes cluster. This means no one is trusted by default and requires verification to gain access to the services.

Architecture Overview

AppViewX Kubernetes Architecture

AppViewX workloads are containerized workloads running as microservices and these containers are orchestrated by managed Kubernetes services. Users can prefer the managed k8s platform of their choice.

AppViewX supports deployment on all the three public clouds AWS, Azure and GCP (Google Cloud Platform) using their managed kubernetes engine / services EKS, AKS and GKE specifically.



Copyright © 2025 AppViewX Inc. All Rights Reserved.

Benefits of AppViewX Architecture

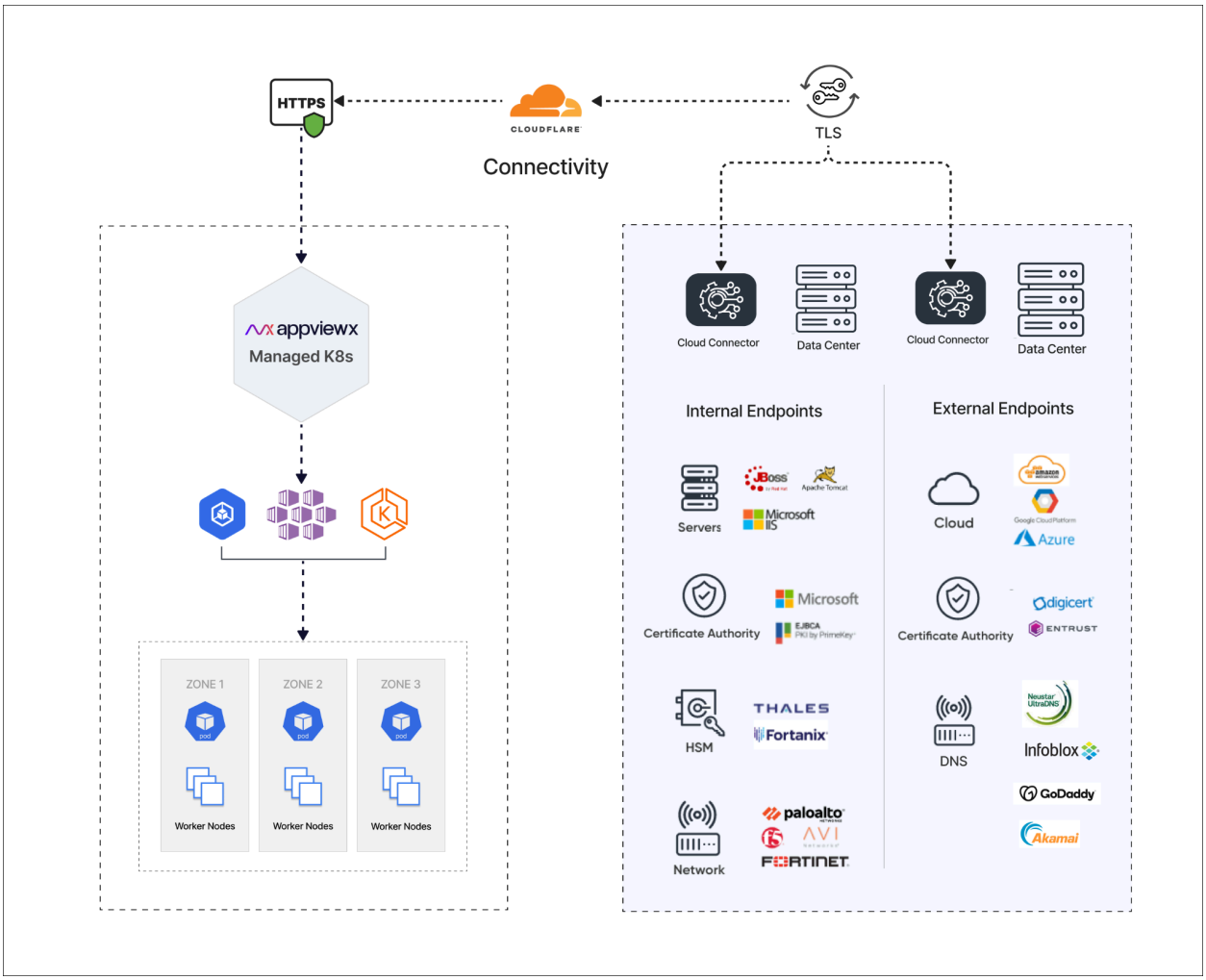
In order to optimally utilize the resources, AppViewX has adopted Kubernetes to achieve higher security by adopting a zero trust network model. The features of AppViewX coupled with Kubernetes are given below.

- **Auto scaling** - AppViewX services can have a custom throttling capability based on pre-configured memory configuration per API. This will enable AppViewX services to utilize (scale up) resources optimally as the demand surges and scale down when not in use. This will help to horizontally scale the vendor components on demand and optimize the resource usage.
- **Resiliency** - There is no guarantee that AppViewX services may run without any interruptions and they are bound to fail. Kubernetes keeps deployments healthy by restarting containers that have failed, by killing and replacing unresponsive containers based on health checks. This helps to mitigate the common pain point of the application's upkeep process.
- **Security** - AppViewX architecture is designed around the concept of zero trust network model to enforce tighter security within the Kubernetes cluster. This means no one is trusted by default and requires verification to gain access to the services.

AppViewX Deployment Architecture

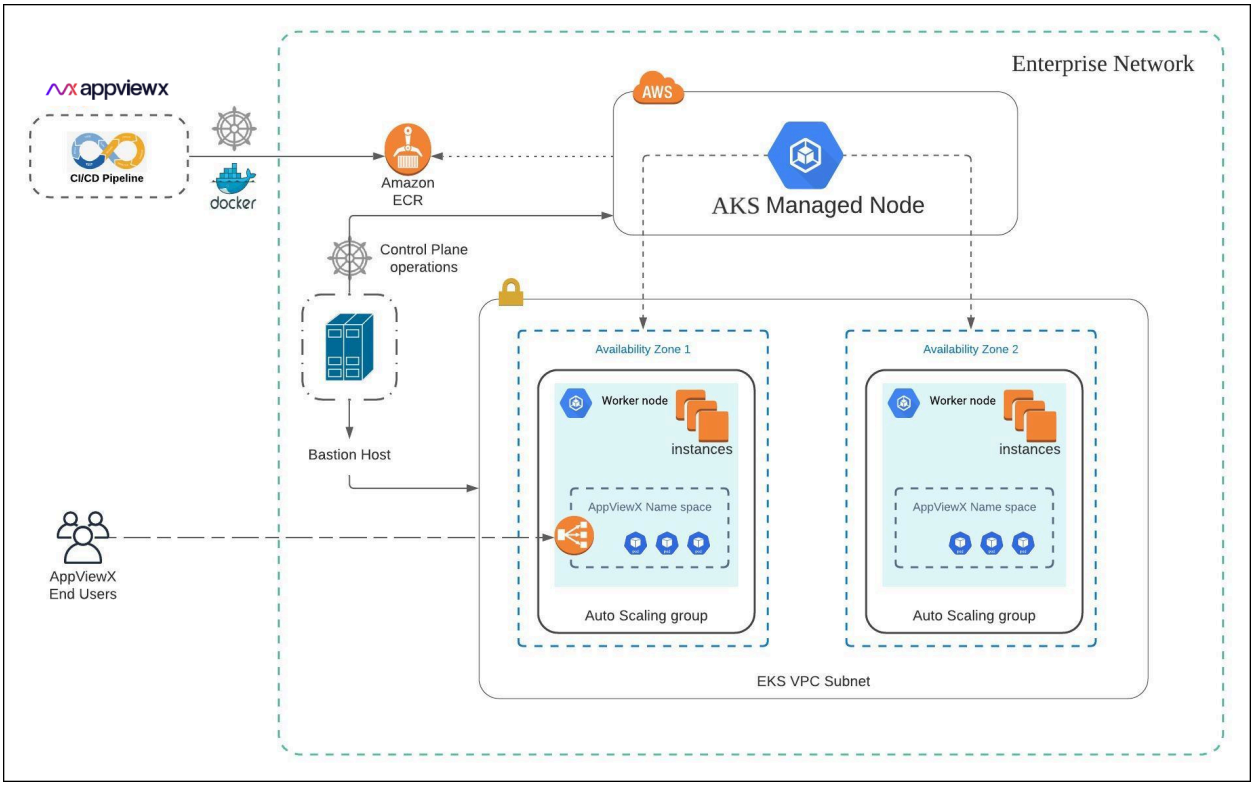
The figure below shows a standard AppViewX deployment architecture model via managed Kubernetes service for AKS.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



Copyright © 2025 AppViewX Inc. All Rights Reserved.

AKS Deployment Model



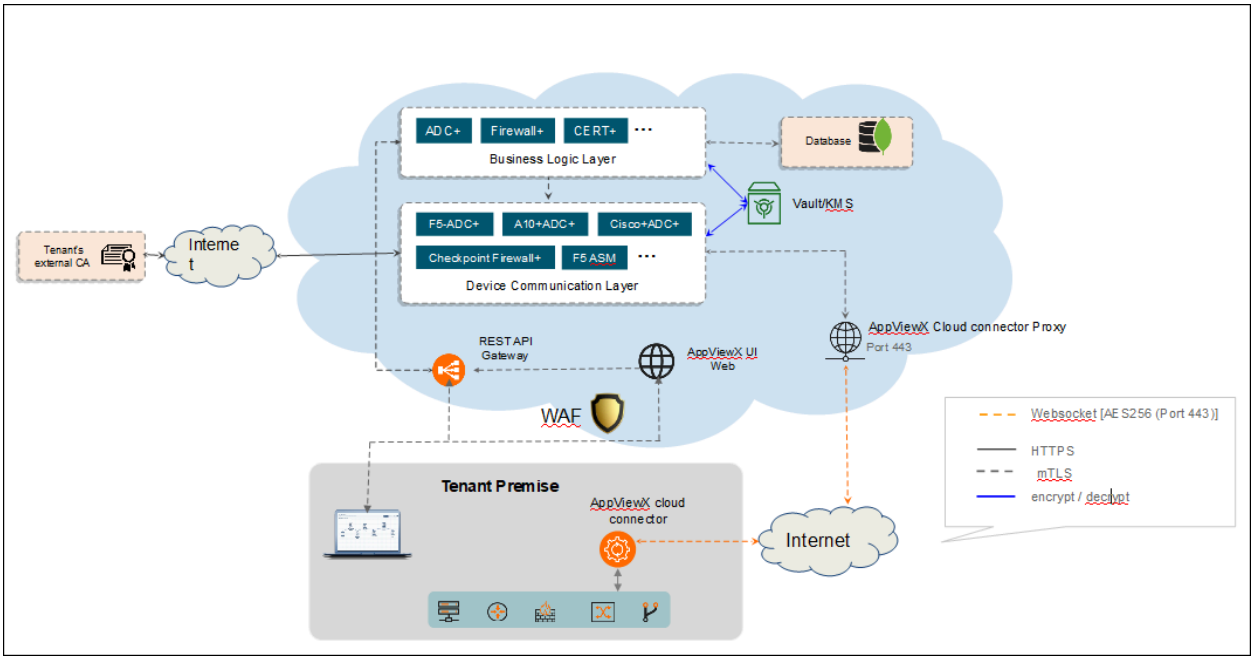
Cloud Connector

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 and your enterprise network without requiring any complex network or infrastructure configuration

Key features of the AppViewX Cloud Connector:

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.



For more details on cloud connectors refer to AppViewX Cloud Connector User Guide.

Note: The below steps have to be performed in all the cloud connector host machines after the 2022.1.0FP2 to FP3 patch upgrade and before the FP3 cloud connector upgrade.

1. Navigate to the installation path in the cloud connector host machine.
2. Execute the following command:

```
./deps/tools/k3s kubectl get deploy avx-mid-server-starter -n cc -o yaml > starter.yaml && sed -i "s/-Xmx2560m/-Xmx4g/g" starter.yaml

&& ./deps/tools/k3s kubectl replace -f starter.yaml
```

Managed Kubernetes Architecture

Managed Kubernetes clusters are composed of the following main components — a control plane and worker nodes. Each cluster runs in its own, fully managed Virtual Private Cloud (VPC).

- The **control plane** is composed of three master nodes, each running in a different Availability Zone to ensure high availability. Incoming traffic directed to the Kubernetes API passes through the respective cloud service load balancer.
- The **worker nodes** run on virtual instances located in a VPC. Managed kubernetes service engine provides managed node groups with automated lifecycle management. This lets users automatically create, update, or shut down nodes with one operation.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Managed Kubernetes service scales the Kubernetes control plane across multiple Availability Zones of the public cloud to ensure high availability and it automatically scales control plane instances based on load, detects and replaces unhealthy control plane instances, and automatically patches the control plane.

Managed Kubernetes workload instances are deployed in multiple availability zones within the region. Each instance has replicas of the services and nodes which exist across all the virtual instances.

Each zone or instance has an active pod listening to other instances. In case of a failure in any instance, the active pod ensures seamless functioning of the application by activating the nodes from any other working cluster.

AKS Components

The following AKS components are utilized by AppViewX:

- Azure Kubernetes service
- Storage account for storing mongodb and vault backups
- Service principal for accessing the ACR registry

Prerequisites

The following prerequisites must be met before the installation process.

- Managed Kubernetes Version Support Matrix
- Disks Used for AppViewX Installation
- AppViewX Docker Images
- AppViewX Helm Charts
- Bastion Host Setup
- AKS Cluster
- Azure Storage
- Azure Service Principal

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Managed Kubernetes Version Support Matrix

Public Cloud	
Mode of Deployment	**Azure**
Release, Vendor, & Product Support **AppViewX v2024.3.0.0**	
Managed K8s Deployment (AKS)	
K8s version 1.33	Yes

Disks Used for AppViewX Installation

Disks Used

Volume	Size	Quantity
logs volume	10Gi	1
avx-kafka	20Gi	3
zookeeper	20Gi	3
openbao	10Gi	3
mongodb	256Gi	3
redis	5Gi	3

If a third party is installed, the values are as follows:

Disks Used (Third Party)

Volume	Size	Quantity
Elasticsearch-ELK	10Gi	1
Elasticsearch-Insight	10Gi	1

Copyright © 2025 AppViewX Inc. All Rights Reserved.

AppViewX Docker Images

AppViewX Docker images are hosted in a private registry https://images.appviewx.com. These images can be pulled using an authentication token (contact AppViewX Support, help@appviewx.com for the authentication token) and can be hosted in the private or public repository at the customer end.

The list of docker images are

- <registry link>/appviewx/pilot:1.26.1
- <registry link>/appviewx/proxyv2:1.26.1
- <registry link>/appviewx/openbao:2.2.2
- <registry link>/appviewx/redis:8.0.2
- <registry link>/appviewx/mongo-init:<tag>
- <registry link>/appviewx/avx-cloud-gateway:<tag>
- <registry link>/appviewx/avx-cloud-web:<tag>
- <registry link>/appviewx/avx-cloud-mongoseed:<tag>
- <registry link>/appviewx/avx-cloud-managedservice-mks:<tag>
- <registry link>/appviewx/avx-platform-report-generator:<tag>
- <registry link>/appviewx/avx-python-sandbox:<tag>
- <registry link>appviewx/avx-mid-server-base:<tag>
- <registry link>/appviewx/kafka:0.46.0-kafka-3.9.0
- <registry link>/appviewx/operator:0.46.0
- <registry link>/appviewx/alpine:3.22.0
- <registry link>/appviewx/kube-metrics-adapter:v0.2.3
- <registry link>/appviewx/kube-state-metrics:v2.15.0
- <registry link>/appviewx/backup-utility-image:v2024.3.0
- <registry link>/appviewx/prometheus:v2.53.4
- <registry link>/appviewx/metrics-server:v0.7.2
- <registry link>/appviewx/elasticsearch:9.0.2
- <registry link>/appviewx/elasticsearch-insight:9.0.2
- <registry link>/appviewx/filebeat:9.0.2
- <registry link>/appviewx/grafana:11.6.2
- <registry link>/appviewx/kibana:9.0.2
- <registry link>/appviewx/logstash:9.0.2
- <registry link>/appviewx/logstash-syslog:9.0.2
- <registry link>/appviewx/alertmanager:v0.28.1
- <registry link>/appviewx/node-exporter:v1.9.1
- <registry link>/appviewx/redis_exporter:v1.73.0

Copyright © 2025 AppViewX Inc. All Rights Reserved.

The steps to download the images from AppViewX repository are as follows:

1. Get the source image repository credentials from AppViewX Support team.
2. Configure the docker using the command

```
docker login -u ${USERNAME} -p ${PASSWORD} ${DOCKER_REPOSITORY}
```

3. Configure the respective cloud provider CLI (Google cloud) and ensure you have access to push docker images to GCR.
4. To push the docker images, use the helper script provided by AppViewX. Follow the steps below.

   a. Download the artifact Managed-Kubernetes_helper_scripts.tar.gz to the bastion host and extract using the command:

   ```
   tar -xf Managed-Kubernetes_helper_scripts.tar.gz
   ```

   b. Navigate to the extracted directory **mk8s_helper_scripts**.

   ```
   cd mk8s_helper_scripts
   ```

   c. Execute the script **avx_image_pull_push.sh** using the command

   ```
   ./avx_image_pull_push.sh <Image tag> <customer registry url>
   ```

   > **Note:** Replace <Image tag> with **24.3.0.0-1505** and <customer registry url> with the actual values.

AppViewX Helm Charts

The helm charts used by AppViewX for installation are released as a part of the installer. The installer consists of helm charts and an AppViewX utility which helps orchestrate the deployment, patch, upgrade and maintenance of AppViewX across managed kubernetes deployment.

Bastion Host Setup

The following packages must be installed on the bastion host or the host/tool from where the installation is triggered

Azure CLI

To set up the Azure CLI refer to Install the Azure CLI on Linux on the Microsoft documentation website.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Execute the following command:

```
curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash
```

Kubectl

To set up Kubectl refer to Install and Set Up kubectl on Linux on the Microsoft documentation website.

Execute the following commands

- sudo curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
- sudo chmod +x kubectl
- sudo mv ./kubectl /usr/bin/#

Verify installation by executing the command

```
kubectl version
```

Helm

Helm is required only if the deployment is triggered from any other machine instead of the DevOps pipeline. To set up Helm refer to Installing Helm on the Helm documentation website.

Execute the following command:

- curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3
- chmod 700 get_helm.sh
- ./get_helm.sh#

Verify installation by executing the command

```
helm version
```

AKS Cluster

To create an AKS cluster refer to Microsoft's online manual - Azure Kubernetes Service (AKS). Although Microsoft manuals are always up-to-date, the recommended choice to make before creating the cluster is as follows:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- Kubernetes version: 1.31
- The network model:
  - Azure Kubenet (supported by AppViewX).
  - Azure CNI (Recommended for optimal performance).
- Managed identity: System assigned managed identity.
- Enable Kubernetes RBAC.
- **Agent nodepool**: Three nodes of Machine type **D2sv4** with Auto Scaling disabled. Add taint to agent nodepool as **CriticalAddonsOnly=true:NoSchedule** to disable scheduling of application pods to the agent nodepool.

> **Note:** The taint **CriticalAddonsOnly=true:NoSchedule** prevents the application pods from being scheduled on system node pools.

- **User nodepool**:
  - **appnodepool**: Three nodes of type **Da8sv4** with Auto Scaling disabled
  - **mongonodepool**: Three nodes of type **Da8sv4** with Auto Scaling disabled. Add label **mongo=true** and taint **designatedMongo=true:NoSchedule** to the nodepool (to be performed while creating the cluster).

> **Note:** A minimum of 3 availability zone are needed during cluster creation to support the single AZ failover.

- Select multi zones for the Agent nodepool and the User Nodepool.

> **Note:** The number of nodes mentioned here are applicable for managing up-to 25K certs. This number will vary if there are more certificates to manage.

Azure Storage

Azure Storage Account

A Storage account is required to store

- iControlJar
- MongoDB backup
- Vault backup

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- Axisjar
- javax.xml.soap-api

Always create a storage account with a valid name to indicate the storage account for a specific AKS cluster. A typical naming convention could be **<clusternamestorage>**.

> **Note:** Storage account access by the AKS pods is granted by the storage account connection string.

For more information on the Azure storage account, refer to Microsoft's online manual - Create a storage account.

Azure Storage Container

The following containers must be created in the storage accounts that have already been setup.

1. **icontroljar**: The iControlJar needs to be placed here before installing AppViewX plugins
2. **mongo-backup**: Backup job stores mongodb backup into the container.
3. **vault-backup**: Backup job stores vault backup into the container
4. **axisjar**: Axisjar, javax.xml.soap-api jar files must be stored in container **axisjar**.

For more information on the Azure storage containers, refer to Microsoft's online manual - Quickstart: Upload, download, and list blobs with the Azure portal.

Azure Service Principal

Azure Service Principal is used to create the image pull secrets and download images from ACR. When creating the Service Principal

- Set the expiration to never
- Assign the ACR pull access role

For information on commands to create the Azure Service Principal refer to Microsoft's online manual - Azure Container Registry authentication with service principals.

A summary of steps are as follows:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- To derive the service principal id and the password execute the helper script provided by AppViewX. To use this script follow the steps below.

1. Download the artifact Managed-Kubernetes_helper_scripts.tar.gz to the bastion host and extract using the command:

```
tar -xf Managed-Kubernetes_helper_scripts.tar.gz
```

2. Navigate to the extracted directory **mk8s_helper_scripts**.

```
cd mk8s_helper_scripts
```

3. Edit the file **acr_reg_config.sh** and replace ACR_NAME with the actual value.

4. Execute the **acr_reg_config.sh** file.

```
bash acr_reg_config.sh
```

Note: After the script execution, capture the outputs as they are required in the global utility config.

Install AppViewX in Managed Kubernetes

Migration Strategy

Attention: If you are performing a fresh install, then refer the next sub-topic **Installation Steps**.

To migrate from AppViewX on-prem versions (2022.1.0, 2021.1.0, and 2020.3.0) to Managed Kubernetes, it is important to take a backup of the mongodb and vault in the respective on-prem versions. Before you take the backup, execute the script below.

```
db.profile.update({'_id' : 'installationType'}, {$set : {"value" : "Managed_K8s"}})
```

Note: Refer to the specific version of the release documents from the release portal and perform the backups or contact the AppViewX support team.

After performing the backup, follow the installation steps detailed in the section below. At step 11 of the installation process, ensure to restore the data at this stage.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Installation Steps

This section describes the steps to for installing the AppViewX Stack on AKS.

1. Download the installer from the release portal.
2. Create a directory Managedk8s-installer in the bastion host and extract the installer file **tar -xf installer.tar.gz** in the same directory.
3. Verify that the extracted installer must have the following files
   - appviewxctl (binary)
   - helm_charts (directory of helm charts)
4. Generate the configuration files based on the cloud provider. If the cloud provider is **Azure**, execute the command below.

```
./appviewxctl config generate --provider azure
```

5. Verify that the execution of the above command creates the configuration files named **.appviewxctl.yaml** in the same location.
6. The file .appviewxctl will be populated with the fields necessary for installation, in particular cloud provider that was provided in previous command (**-- provider**).
7. Edit the **appviewxctl.yaml** file and populate the values as described below:

appviewxctl.yaml file - Parameters and Description

Parameters	Description of Values
chartPath	The path to the helm_charts which is to be installed. It points to the helm_charts directory extracted in step 3.
configFile	The path to the kube config file to be used by helm and kubectl. If the bastion host is already configured and kube config is under $HOME/.kube directory, then keep this field empty.
install.enableAppBackupCron	Boolean value to enable/disable the backup cronjobs. (True/False). This value is needed for self-managed mongodb only. For atlas backup this has to be scheduled in the atlas dashboard.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Parameters	Description of Values
install.enablePrivateImagePullSecret	Boolean value to enable image pull secret. Set values as **false** if the cluster already has access to the container registry. Otherwise set it to **true** and fill all the details of the access keys described in below sections.
install.enableThirdPartyInstall	Boolean value (True/False) to determine whether third party monitoring components such as ELK, Monitoring, and Insight needs to be installed.
install.thirdPartyApp.elk	Boolean value to add Elk component. Set to True if it needs to be installed.
install.thirdPartyApp.monitoring	Boolean value to add Monitoring component. Set to True if it needs to be installed.
install.thirdPartyApp.insight	Boolean value to add Insight component. Set to True if it needs to be installed.
install.imageRegistry	The URL of the container registry where the images are to be pulled from by the pods. *Example*: appviewx.azureacr.io
install.imageTag	The tag of the image that will be used for installation. *Example*: 2022.1.0_FP_750-alpine
install.isSaasEnabled	Boolean value for SaaS enablement. This value should be set to **true** for Managed K8s.
install.kafkaCloudConnector	It is a combination of three values. • enable • password • user Set **enable** to ***true*** and keep the user, password fields empty for Managed K8s. *Example*

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Parameters	Description of Values
	kafkaCloudConnector: enable: true password: "" user: ""
install.mongo	It is a combination of fields specific to the type of mongodb used.
dbIsolation	Boolean value to indicate whether the database isolation is to be enabled. In order for database isolation to work, the following prerequisite must be taken care of while creating the cluster node group. • Add label **mongo=true** and taint **designatedMongo=true:NoSchedule** to the nodepool to be used for mongodb.
mongoAtlas	The fields specific to mongo atlas are as folows: • **enable**: Boolean value to decide if mongo atlas to be used. If set to *false*, a self managed mongo cluster will be created. If set to *true* mongo atlas will be used and details of which are to be provided in below mentioned fields. • **host**: URL of the mongodb atlas cluster. • **password**: password of the mongodb atlas cluster. • **user**: username in the mongodb atlas cluster. *Example:* mongo: dbIsolation: false mongoAtlas: enable: true host: "managed-k8s.test.mongodb.net" password: "samplepassword" user: "user1"

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Parameters	Description of Values

install.useDockerPrivateRegistry	Set this to **true** if the dockerhub private repository is to be used for pulling the necessary images needed. Otherwise set the value **false** and the container registry ACR, ECR, and GCR will be used based on the cloud provider.
	
	If this value is set to *true*, populate the below values, otherwise keep it empty.
	• **dockerhub.pass**: password to be used for authenticating in the dockerhub private repository.
	• **dockerhub.username**: username configured in the dockerhub private repository.
	
	Example:
	
	`useDockerPrivateRegistry: true`
	`dockerhub:`
	`  pass: "testpassword"`
	`  username: "appviewx"`

install.size	The size of the installation. Based on the use cases and number of certs to be managed there different sizes (contact AppViewX for sizing recommendations). The supported size values are (case sensitive values) • xsmall • small • medium • large • xlarge • custom *Example:* `size: small`

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Parameters	Description of Values
	Note: The size provided must be taken into cluster creation and nodegroup sizes must be defined accordingly. Follow the same document link above for nodegroup sizes.

install.plugins	The list of plugins that will be installed. Each plugin will have three fields
	• enable
	• imageTag
	• name
	
	Set enable to **true** if the plugin is to be installed. If the same image tag is to be used as defined in the global ImageTag keep it **latest** otherwise override with some other tag of your choice.
	
	Example:
	
	`- enable: true`
	`imageTag: latest`
	`name: avx-config-server`
	
	To enable Cloud DC support in Managed Kubernetes, set plugins as follows:
	
	`- enable: true`
	`imageTag: latest`
	`name: avx-mid-server-platform`
	
	Note: Ensure that **install.isSaasEnabled** and **install.kafkaCloudConnector** are set to **true**.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Parameters	Description of Values
internalLoadBalancer	If set to **true**, all the Loadbalancers will be private and can only be accessed within the VPC else it will be public.
install.enableSftpStorage	Change to true to use SFTP server for mongodb, vault, and icontrol.jar storage. Boolean (Default: false)
install.sftpServerDetails. dbBackupPath	Provide the location of mongodb backup storage directory. String (Default: “”)
install.sftpServerDetails. vaultBackupPath	Provide the location of vault backup storage directory. String (Default: “”)
install.sftpServerDetails. sftpServerUserName	Provide the username of SFTP server. String (Default: “”)
install.sftpServerDetails. sftpServerIp	Provide the sftp server IP. String (Default: “”)
cloudConnectorEnabled	A boolean value (true/false) to denote the cloud connector usage for southbound communications. If a cloud connector is used set the value to **true**.
socketurl	Customers can now configure their own web-socket URL in the MongoDB by updating the `socketurl` parameter in the **.appviewxctl.yaml** file. After installation or upgrade, the specified socket URL will be recorded in the **saas_agent_meta** collection.

The next fields are to be filled with values that must be collected during the cluster creation and setup process and filled as mentioned below.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

appviewxctl.yaml file - Parameters and Description (for cluster creation)

Parameters	Description of Values
install.privateImagePullSecret	In this section populate the details of the access keys needed to authenticate and pull the image from the registry. They are not needed if the Dockerhub is used as described above. • **registry**: The ACR registry URL • **servicePrincipalPassword**: The service Principal Password for accessing the registry. • **servicePrincipalUsername**: The service Principal Username for accessing the registry. *Example:* registry: "appviewxsample.azurecr.io" servicePrincipalPassword: "qLPUSA4R1ALkA~GH6m4v70iAC_jajEo9T" servicePrincipalUsername: "20892076-ct8a-4700-a7c0-178u066q9a9c"
install.storageAccess	The storage bucket details to be used for setting up backup capability. • **bucketObject**: The storage bucket access string. • **serviceAccountAnnotation**: "none" *Example:* bucketObject: "DefaultEndpointsProtocol=https;AccountName=sampleappviewx;AccountKey=Qy0SKtry2MR4Ik0OIG+po3p0Kgla7u4KEjlYo10jHYdVlZXP2/v4lMomkZK6s58YLsLbzkutkyjHJINuCo2Y7w==;EndpointSuffix=core.windows.net" serviceAccountAnnotation: "none"

The following fields must be added to integrate the kubernetes cluster to the external vault.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

appviewxctl.yaml file - Parameters and Description (for external vault)

Parameters	Description
install.externalVault.enable	A boolean value (true/false) to denote if the external vault is to be used in the setup. True is to enable the external vault.
install.externalVault.externalVaultAddr	Contains the vault URL and listening port *Example*: https://pm-lxs-node01.lab.appviewx.net:8200
install.externalVault.externalVaultAuthRole	Name of the role created against the access kubernetes auth path
install.externalVault.externalVaultCACertSecret	Name of the secret where **vault-ca.crt** file is mounted.
install.externalVault.externalVaultDBRole	Static role created to access the database cred.
install.externalVault.externalVaultEnginePath	Enter the value “/database”
install.externalVault.externalVaultKubeAuthPath	The Kuberenetes access path created with cluster information for service account authentication.
install.externalVault.externalVaultSAName	The Service account used to create externalVaultAuthRole.
install.externalVault.mongoPasswordVaultEngine	Enter the value DATABASE

8. Once the values are filled in *.appviewxctl* as described in the step above, proceed with the installation. Before doing so, check if the the preconditions are met by executing the command

```
./appviewxctl preflight --config .appviewxctl.yaml
```

   This will prompt if the necessary prerequisites are met.

9. The metrics server in the Azure clusters comes pre-installed with the cluster, hence they must be disabled from the **avx_pre_req** chart.
   a. Navigate to helm_charts/avx_pre_req.
   b. Edit the **values.yaml** file by setting the following parameters.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
avx-metrics-server:
enable: false
```

The metrics server installation is disabled.

10. To proceed with installation, execute the command

```
./appviewxctl install --config .appviewxctl.yaml
```

Note: The installation will take several minutes to complete. Upon completion you see the following message:

`[Install] Successfully installed Appviewx infra stack`

This would imply the completion of infra component setup.

11. This step involves restoring the existing data from the previous AppViewX version's cluster in case there is a need to migrate from the older versions to the Managed K8s version. **Ignore this step if it's a fresh setup with no migration necessary.**

    To restore mongodb and vault fetch the backup files and place them in the bastion in a directory such as /home/user/backup execute the mongo_restore and vault_restore scripts as follows:

Note: Before you trigger the `mongo_restore.sh` command, ensure that the mongo restore file has the file extension as **.tar.gz**.

```
./mongo_restore.sh <mongo backup tar filepath>
```

```
./vault_restore.sh -p <vault backup filepath> --removedek
```

Attention: If the data is being restored from an older version (2020.3.0 - 2022.1) then use the command

```
./vault_restore.sh -p <vault backup filepath> --removedek
```

Note:

- The backup files must have extension as **.tar.gz**
- The above commands work for a self-managed mongodb setup. Setting up the mongodb atlas requires the installation of mongodb tools in the bastion host as described below.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

For an rpm based OS:

```
echo -e "[mongodb-org-4.2] \nname=MongoDB
 Repository\nbaseurl=https://repo.mongodb.org/yum/redhat/\$releasever/mongodb-org/4.2/x86_64/\ngpgcheck=1\nenabled=1\ngpgkey=https://
www.mongodb.org/static/pgp/server-4.2.asc" > /etc/yum.repos.d/mongodb-org-4.2.repo
yum install mongodb-org-shell-4.2.0
yum install mongodb-org-tools-4.2.0
```

For a debian based OS:

```
wget -qO - https://www.mongodb.org/static/pgp/server-6.0.asc | sudo apt-key add -
sudo apt-get install gnupg
wget -qO - https://www.mongodb.org/static/pgp/server-6.0.asc | sudo apt-key add -
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0 multiverse" | sudo
 tee /etc/apt/sources.list.d/mongodb-org-6.0.list
sudo apt-get update
sudo apt-get install -y mongodb-mongosh
sudo apt-get install -y mongodb-org-tools
```

Verify if the mongo restore commands have executed successfully using the command

```
mongorestore -- version
```

12. To proceed with the AppViewX application installation, execute the command:

```
./appviewxctl installapp --config .appviewxctl.yaml
```

Once installation is complete the following messages are displayed:

```
[Install] Appviewx infrastructure chart [avx-app] installed successfully
[Install] Successfully installed Appviewx application stack
[Install] Fetching login URL for app
[Install] Waiting for Public IP allotment for istio service
[Install] AppViewX Web URL: https://34.100.197.159/appviewx/
[Install] AppViewX Gateway URL: https://34.100.197.159/avxmgr/
[Install] Grafana URL: https://34.100.197.159/grafana/
[Install] Kibana URL: https://34.100.197.159/kibana/login
[Install] Run below commands to get mongo user credentials
 export MONGO_USER=$(kubectl get secret -n avx mongo-key -o=jsonpath='{.data.mongo-init-user}' | base64 -d)
 export MONGO_PASS=$(kubectl get secret -n avx mongo-key -o=jsonpath='{.data.mongo-init-pass}' | base64 -d)
[Install] Run below commands to get Elasticsearch and Kibana credentials
 export ES_PASS=$(kubectl get secret -n avx elasticsearch-pw-elasticsearch -o=jsonpath='{.data.password}' | base64 -d)
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
export KIBANA_PASS=$(kubectl get secret -n avx elasticsearch-pw-kibana -o=jsonpath='{.data.password}' | base64 -d

[Install] Application Installation completed successfully
```

Note: Follow the URLs and commands given in the output message to get the credentials and access the application.

13. If installation of the third party monitoring components was not enabled during the entire process, they can be installed later by the following steps:

    a. While installing the third party components (helm_charts/avx_third_party/values.yaml), the only that values are set to 'true' by default are - *prometheus, nodeexporter, kube-state metrics*. The other components are set as 'false' by default and must be to set to true if they are to be enabled, they are - *elk-elasticsearch, elk-filebeat, elk-kibana, elk-logstash, grafana, elasticsearch-insight, logstash-syslog.*

    b. Edit the **.appviewxctl.yaml** file and set **install.enableThirdPartyInstall** to ‘true’

    c. Configure the following **thirdPartyApp** parameters as true as per the requirements:
    - **install.thirdPartyApp.elk**
    - **install.thirdPartyApp.monitoring**
    - **install.thirdPartyApp.insight**

    d. Now, edit the file **values.yaml** present at location helm_charts/appviewx_monitoring/prometheus/chart/values.yaml and append the below values at the end of the file (only if that are not present).

```
limits:
  cpu_limit: 80
  memory_limit: 80
  disk_limit: 80
  timelimit_cpu_memory: 5
  timelimit_disk: 1
  timelimit_pod: 1
  timelimit_node: 1
```

    e. Run the command below

```
./appviewxctl installtpt --config .appviewxctl.yaml
```

What to do next:

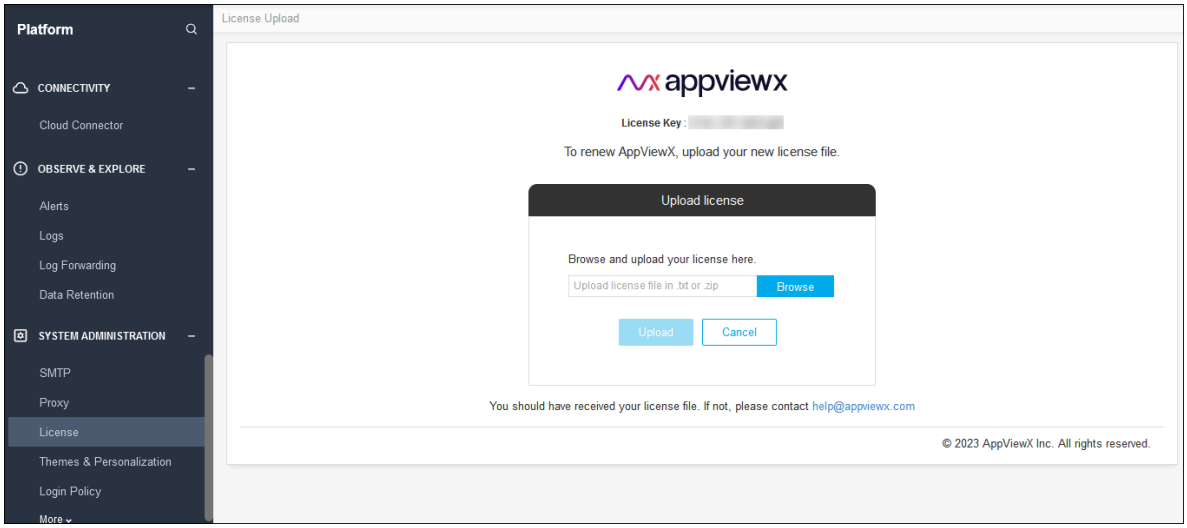
Copyright © 2025 AppViewX Inc. All Rights Reserved.

1. If you have to install the Helm based CC, then parch the internal tenant secret, using the command below:

```
kubectl patch secret internal-tenant -n avx --type='json' -p='[{"op": "add", "path": "/data/REPO_AUTH_TOKEN", "value": "<repoauthtokenvalue>"}]'
```

   Once the command is executed, restart the saas-proxy.

2. Customers migrating from AppViewX version 2020.3.0 to Managed Kubernetes, it is mandatory to upgrade the license.
   a. Login to the AppViewX with valid credentials.
   b. Navigate to Platform >> System Administration >> License page.
   c. Click **Upgrade License**.



   d. Click **Browse** to find the latest license key file.
   e. Click **Upload**.

> **Note:** For the licenses contact AppViewX Support at help@appviewx.com or customerlicences@appviewx.com.

Upgrade AppViewX in Managed Kubernetes

Before you begin: Before performing the infra upgrade, upgrade the kubernetes to the latest version **1.31**. Refer to the available Upgrade options for Azure Kubernetes Service (AKS) clusters.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Attention:

- If you are using the self managed private docker registry instead of AppViewX's docker registry, then before proceeding with the upgrade, ensure you have copied the latest images to your registry. The list of images can be found in the Prerequisite section - AppViewX Docker Images.

To upgrade AppViewX with a new image version, follow the steps below:

1. To take the backups, execute the commands below.
   For self-managed mongodb:

   ```
   kubectl create job --from=cronjob/mongo-backup -n avx mongo-backup-<unique-identifier>
   ```

   ```
   kubectl create job --from=cronjob/vault-backup -n avx vault-backup-<unique-identifier>
   ```

   Replace <unique-identifier> in above commands with some random string and run. Monitor the pods until completion and verify the backups are placed in the storage bucket.

   Note: Atlas backup must be taken in the atlas dashboard. Refer to the atlas snapshots section in the page Backup and Restore.

2. Download the upgrade tar file (**upgrade.tar.gz**) from the release portal and extract it to a suitable location. (The extracted files contain the binary and helm charts tar.)
3. Navigate to the folder where the upgrade tar is extracted.
4. Copy the **appviewxctl** binary from the current folder (extracted folder location) to the installer location.

   ```
   cp appviewxctl <absolute path of the installer directory>
   ```

5. Navigate to the installer directory.
6. To merge the old data of .appviewxctl.yaml to the new file, execute the command below.

   ```
   ./appviewxctl mergeConfForUpgrade --config .appviewxctl.yaml
   ```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
eksuser@bastion-hsot-qa-1:~/mar6$ ./appviewxctl mergeConfForUpgrade --config .appviewxctl.yaml
[Init] Using log file at [/home/eksuser/mar6/appviewxctl-554518342.log] to dump logs
[Init] Initialise persistent flag config
[Init] Using config file
Created backup file: .appviewxctl-old-20250310-134744.yaml
[Config] Generating config file with provider gcp - .appviewxctl.yaml
[Config] Successfully generated sample config file.
Waiting for config generation to complete...
Created backup file: .appviewxctl-tmp-bkp.yaml
Plugin 'avx-vendors-sync' has been added under install key. Do you want to enable it? (y/n): y
Field 'socketurl' has been added under root key. Please enter the new field value (default: ): web.socket.com
Plugin 'avx-vendors-sync' has been added under upgrade key. Do you want to enable it? (y/n): n

Summary of New Fields:
-------------------------------------------------------------
Field Path                          | Value
-------------------------------------------------------------
avx-vendors-sync                    | enable: true
socketurl                           | web.socket.com
avx-vendors-sync                    | enable: false
-------------------------------------------------------------

Do you want to proceed with these changes? (y/n): y
Merged old YAML values into new YAML.

Please enter the output file name for the new YAML. Hit enter to take the default value. (default: .appviewxctl.yaml):
Successfully saved YAML to .appviewxctl.yaml
```

During the execution of the above command, the fields of the old conf file are compared with that of the newly generated conf file and you will be prompted to enter values.

- If it's a plugin, you'll be prompted to enable or disable it during both installation and upgrade.
- If a new field has been added to the `.appviewxctl.yaml` file, you'll be prompted to provide a value for it. Pressing Enter without input will apply the default value.

Once all the values are entered, the process continues as follows:

- A summary table is displayed to confirm the provided inputs.
- If the user declines, the prompts will repeat.
- If the user confirms, the system proceeds to merge the old YAML with the new one.
- The values provided by the user for the newly added fields will be retained, while existing fields will retain values from the old configuration file.
- Users can specify a custom output filename for the merged YAML; if not specified, the default will be `.appviewxctl.yaml`.
- The merged YAML is successfully saved to the output file.

7. Refer to the tables below for the essential upgrade parameters and their description.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Upgrade Parameters in the appviewxctl.yaml file

Parameters	Description of Values
upgrade.imageRegistry	The URL of the container registry where the images are to be pulled from by the pods. *Example*: gcr.io/pe-qa-358108
upgrade.imageTag	The tag of the image that will be used for upgrade. *Example*: 2023.1.0_FP_750-alpine
install.imageTag	The tag of the image that will be used for installation. *Example*: 2022.1.0_FP_750-alpine
upgrade.isSaasEnabled	Boolean value for SaaS enablement. This value should be set to **true** for Managed K8s.

upgrade.plugins	The list of plugins that will be installed. Each plugin will have three fields
	• enable
	• imageTag
	• name
	
	Set enable to **true** if the plugin is to be upgraded. If the same image tag is to be used as defined in the global ImageTag keep it **latest** otherwise override with some other tag of your choice.
	
	Note: The list of plugins to be enabled should match the ones in the install section.
	
	Example:
	
	`- enable: true`
	`imageTag: latest`

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Parameters	Description of Values
	`name: avx-config-server`
upgrade.upgradeInfra	Boolean value to upgrade infra component. Set to True for upgrade.
upgrade.upgradeThirdParty	Boolean value to upgrade the monitoring (ELK, insight, and monitoring) components. Set to True for upgrade.
socketurl	Customers can now configure their own web-socket URL in the MongoDB by updating the `socketurl` parameter in the **.appviewxctl.yaml** file. After installation or upgrade, the specified socket URL will be recorded in the **saas_agent_meta** collection.

Component Parameters in the appviewxctl.yaml file

Parameters	Description of Values
install.thirdPartyApp.elk	Boolean value to add Elk component. Set to True for upgrade.
install.thirdPartyApp.monitoring	Boolean value to add Monitoring component. Set to True for upgrade.
install.thirdPartyApp.insight	Boolean value to add Insight component. Set to True for upgrade.

Install parameters in the appviewxctl.yaml file required to integrate the kubernetes cluster to the external vault.

Parameters	Description of Values
cloudConnectorEnabled	A boolean value (true/false) to denote the cloud connector usage for southbound communications. If a cloud connector is used set the value to **true**.
install.externalVault.enable	A boolean value (true/false) to denote if the external vault is to be used in the setup. True is to enable the external vault.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Parameters	Description of Values
install.externalVault.externalVaultAddr	Contains the vault URL and listening port *Example*: https://pm-lxs-node01.lab.appviewx.net:8200
install.externalVault.externalVaultAuthRole	Name of the role created against the access kubernetes auth path
install.externalVault.externalVaultCACertSecret	Name of the secret where **vault-ca.crt** file is mounted.
install.externalVault.externalVaultDBRole	Static role created to access the database cred.
install.externalVault.externalVaultEnginePath	Enter the value “/database”
install.externalVault.externalVaultKubeAuthPath	The Kuberenetes access path created with cluster information for service account authentication.
install.externalVault.externalVaultSAName	The Service account used to create externalVaultAuthRole.
install.externalVault.mongoPasswordVaultEngine	Enter the value DATABASE

Note: Path parameters should have a leading forward slash '/'.

8. Copy the **vault_restore_openbao_migration.sh** file from the current folder (extracted folder location) to installer location

Note: If this step was completed during the previous upgrade (2024.3.0.0), you can skip it. However, if you are upgrading from an older version (2023.1.3 or earlier), this step is required.

```
cp vault_restore_openbao_migration.sh <absolute path of the installer directory>
```

9. Copy the **mongo_restore.sh** file from the current folder (extracted folder location) to installer location

```
cp mongo_restore.sh <absolute path of the installer directory>
```

10. To upgrade AppViewX infra, execute the infraUpgrade command.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

> **Note:** If you plan on enabling additional 3pt monitoring components as part of the infra upgrade do the following:
>
> a. Navigate to <installer>/helm_charts/avx_thrid_party/.
>
> b. Edit the **values.yaml** file.
>
> c. Set "enable" to true for the components you wish to enable as part of the upgrade.

```
./appviewxctl infraUpgrade --config .appviewxctl.yaml
```



This will prompt the following message

```
Vault Backup taken (Yes/No):
```

If you haven't taken the backup, enter **No**, and execute the vault backup command in step 2. If you have already taken the backup, enter **Yes**. You will be prompted with the message below.

```
Please provide the path of updated helm charts tar. :
```

Enter the absolute path of the extracted Helm charts tar artifact to proceed with the infrastructure upgrade. Ensure you provide the full file path, not just the directory location.

> **Note:** After completing the infrastructure upgrade and before triggering the **installMongoRS**, ensure the `install.imageTag` in the `.appviewxctl.yml` file is updated with the latest imageTag.

11. Install **Mongo Replicaset** using the command below

> **Note:** If this step was completed during the previous upgrade (2024.3.0.0), you can skip it. However, if you are upgrading from an older version (2023.1.3 or earlier), this step is required.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
./appviewxctl --config .appviewxctl.yaml installMongoRS
```

12. Install **OpenBao** using the command below.

> **Note:** If this step was completed during the previous upgrade (2024.3.0.0), you can skip it. However, if you are upgrading from an older version (2023.1.3 or earlier), this step is required.

```
./appviewxctl --config .appviewxctl.yaml installOpenbao
```



> **Note:** The plugin pods must not be restarted after installing OpenBao until the plugin upgrade is complete. Restarting the pods prematurely may disrupt functionality, as Vault could lock the user if it detects multiple login attempts.

13. To restore the vault, fetch the backup files and place them in the bastion in a directory such as /home/user/backup execute the restore scripts below.

> **Note:** If this step was completed during the previous upgrade (2024.3.0.0), you can skip it. However, if you are upgrading from an older version (2023.1.3 or earlier), this step is required.

```
./vault_restore_openbao_migration.sh -p <absolute path of the vault_bkp_file_location>
```

14. To restore the mongo, fetch the backup files and place them in the bastion in a directory such as /home/user/backup and execute the restore scripts below.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Note:

a. If this step was completed during the previous upgrade (2024.3.0.0), you can skip it. However, if you are upgrading from an older version (2023.1.3 or earlier), this step is required.

b. Before you trigger the `mongo_restore.sh` command, ensure that the mongo restore file has the file extension as **.tar.gz**.

```
./mongo_restore.sh <absolute path of the mongo_bkp_file_location>
```

15. Run the plugin upgrade command below.

```
./appviewxctl upgrade --config .appviewxctl.yaml
```

Note: It is mandatory to carry out the Infra upgrade before the plugin upgrade.

Rollback Steps

a. Restore the DB using the restore scripts (step 11 in the Installation Steps section) for self-managed DB or in atlas using snapshot restore in the dashboard.

b. Update the **appviewxctl.yaml** upgrade section's values to the previous image tag and re-run the upgrade command.

- Troubleshooting for Upgrade Errors
- MongoShardDB Cleanup
- Vault/Consul Cleanup
- OpenBAO Cleanup
- Mongo Replicaset Cleanup

Post-Upgrade Actions

The following steps have to be performed after you completed the upgrade.

1. Retrieve the updated **web login IP** by executing:

```
kubectl get svc -n istio-system
```

From the output, locate the `ExternalIP` associated with the `istio-ingressgateway` and use that IP to access the web login page.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

2. Update the **RSocket URL** in the properties file and restart the `avx-mid-server-starter` and `avx-mid-server-platform` pods.

   a. In the UI, go to **Platform** > **Cloud Connector**.

   b. Copy the **Cloud Connector URL**.

   c. On the Cloud Connector (CC) node CLI, go to the properties directory:

   ```
   cd <Installer_Folder>/deps/properties
   ```

   d. Open the `appviewx.properties` file using a text editor:

   ```
   vi appviewx.properties
   ```

   e. Update the value of `RSOCKET_URL` with the copied Cloud Connector URL (e.g., `https://98.70.232.0:443`) and save the file.

   f. Restart the following pods to apply the changes:
   - avx-mid-server-starter
   - avx-mid-server-platform

   ```
   kubectl delete pods -n cc avx-mid-server-starter,avx-mid-server-platform --force
   ```

3. If GUI login fails:

   a. Restart all pods:

   ```
   kubectl delete pods -A --all --force
   ```

   b. Allow the pods to restart and verify that they are running.

   c. When all pods are running, log in again.

Troubleshooting for Upgrade Errors

UPGRADE FAILED: another operation (install/upgrade/rollback) is in progress

While performing the upgrade (`./appviewxctl upgrade --config .appviewxctl.yaml`), if the system encounters the error of `UPGRADE FAILED: another operation (install/upgrade/rollback) is in progress` as shown below.

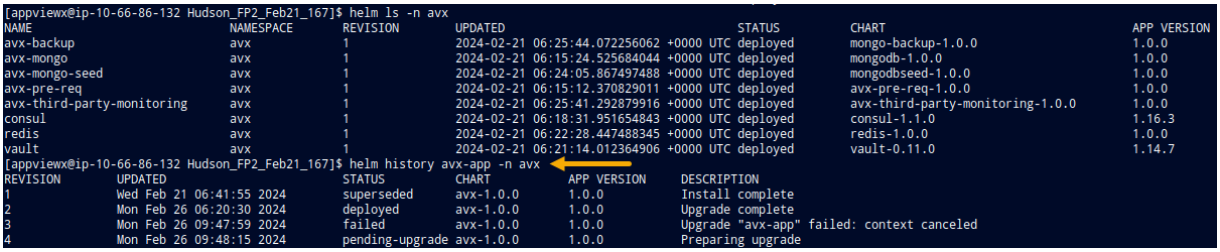
Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
[appviewx@ip-10-66-86-132 Hudson_FP2_Feb21_167]$ ./appviewxctl upgrade --config .appviewxctl.yaml
[Init] Using log file at [/home/appviewx/Hudson_FP2_Feb21_167/appviewxctl-4556929.log] to dump logs
[Init] Initialise persistent flag config
[Init] Using config file
[Upgrade] Upgrade of the AVX application started...
[Upgrade] Updating enabled plugins dependency and configurations
[Upgrade] updated dependency [avx_app]
[Upgrade] Starting Appviewx application chart [avx-app] upgrade
[Upgrade] failed to run upgrade with error: exit status 1: Error: UPGRADE FAILED: another operation
(install/upgrade/rollback) is in progress
```

Perform the following steps:

1. Fetch the deployment history of **avx-app helm** deployment using command mentioned below.

   ```
   helm history avx-app -n avx
   ```



2. Find the most recent revision of avx-app which is in deployed state.

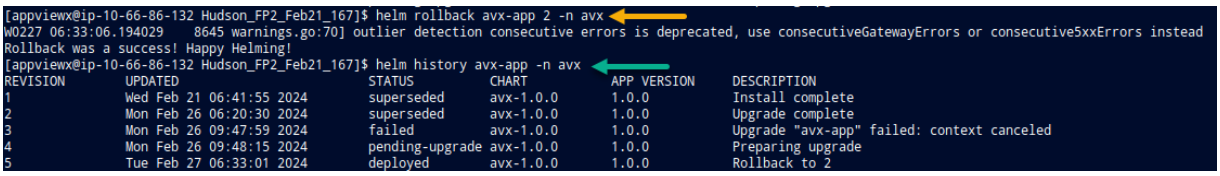
   In this case Revision 2 is the most recent upgrade.

3. Run the rollback command to retrieve the deployment from failed state.

   ```
   helm rollback avx-app <REVISION NUMBER> -n avx
   ```

   Example:

   ```
   helm rollback avx-app 2 -n avx
   ```



4. Re-run the `helm history avx-app -n avx` command to verify the rollback is complete.

Failure Issue during Upgrade

1. Go to the section Cleanup of OpenBao and follow the steps provided.
2. When the cleanup is complete, install in the following order:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
#Install OpenBao
./appviewxctl --config .appviewxctl.yaml installOpenbao

# Restore Vault
./vault_restore_openbao_migration.sh -p <vault backup file full path>

# Restore Mongo
./mongo_restore.sh <mongo backup file full path>

# Upgrade Plugins
./appviewxctl upgrade --config .appviewxctl.yaml
```

Login Issue after Upgrade

If a login issue occurs after the upgrade:

1. Perform he vault cleanup, refer the section Vault/Consul Cleanup.
2. Reinstall OpenBao.
3. Restore the Vault data.
4. Now, repeat the application upgrade process.

MongoShardDB Cleanup

After completing the plugins upgrade and validating the functionality, it is necessary to clean up (delete) all traces of the MongoDB shard database setup. Follow the steps outlined below:

1. Remove the Mongo ShardDB using the command below.

   ```
   helm uninstall avx-mongo -n avx
   ```

2. Delete the StatefulSet resource by executing the command below.

   ```
   kubectl delete sts mongo-shareddb mongo-configdb mongo-routerdb -n avx --force
   ```

3. Delete the specified Kubernetes services (service objects) in the cluster by executing the command below.

   ```
   kubectl delete svc mongo-shareddb-service mongo-configdb-service mongo-routerdb-service -n avx --force
   ```

4. Delete the MongoShardDB related **PVCs** by executing the commands below.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
kubectl delete pvc mongo-configdb-persistent-storage-claim-mongo-configdb-0 mongo-configdb-persistent-storage-claim-mongo-configdb-1
mongo-configdb-persistent-storage-claim-mongo-configdb-2 -n avx --force
```

```
kubectl delete pvc mongo-shareddb-persistent-storage-claim-mongo-shareddb-0 mongo-shareddb-persistent-storage-claim-mongo-shareddb-1
mongo-shareddb-persistent-storage-claim-mongo-shareddb-2 -n avx --force
```

5. Delete the MongoShardDB related kubernetes secrets using the commands below.

```
kubectl delete secret mongo-key -n avx --force
```

Vault/Consul Cleanup

After completing the plugins upgrade or if you have recently migrated/installed OpenBao, and validating the functionality, it is necessary to clean up (delete) all traces of the Vault/Consul setup. Follow the steps outlined below:

> ⚠ **Warning:** Do not perform these steps if the migration from Vault to OpenBao has not been completed

1. Remove the vault/consul and related pods using the commands below.

```
helm uninstall vault -n avx
```

```
helm uninstall consul -n avx
```

```
kubectl delete pods vault-0 vault-1 vault-2 -n avx --force
```

```
kubectl delete pods consul-consul-server-0 consul-consul-server-1 consul-consul-server-2 -n avx --force
```

2. Delete the specified Kubernetes services (service objects) in the cluster by executing the command below.

```
kubectl delete secret vault-key-0 vault-key-1 vault-key-2 -n avx --force
```

3. Delete the vault/consul related **PVCs** by executing the commands below.

```
kubectl patch pvc data-avx-consul-consul-server-0 -n avx -p '{"metadata":{"finalizers":null}}'
kubectl patch pvc data-avx-consul-consul-server-1 -n avx -p '{"metadata":{"finalizers":null}}'
kubectl patch pvc data-avx-consul-consul-server-2 -n avx -p '{"metadata":{"finalizers":null}}'
kubectl delete pvc data-avx-consul-consul-server-0 data-avx-consul-consul-server-1 data-avx-consul-consul-server-2 -n avx --force
```

4. Delete the configmap using the commands below.

```
kubectl delete cm avx-vault-configmap -n avx
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

OpenBAO Cleanup

If any issues are encountered while running the `installOpenbao` command, ensure that all traces of previous attempts are cleaned up before retrying by following these steps:

1. Remove OpenBao and releated pods using the commands below.

```
helm uninstall openbao -n avx
```

```
kubectl delete pods openbao-0 openbao-1 openbao-2 -n avx --force
```

2. Delete kubernetes secrets using the commands below.

```
kubectl delete secret openbao-key-0 openbao-key-1 openbao-key-2 -n avx --force
```

```
kubectl delete secret bao-key-secretid -n avx --force
```

```
kubectl delete secret bao-key-secretid -n avx-kafka --force
```

```
kubectl delete secret bao-key-secretid --force
```

3. Delete **PVCs** by executing the command:

```
kubectl delete pvc data-openbao-0 data-openbao-1 data-openbao-2 -n avx --force
```

4. Delete **configmap** by executing the command:

```
kubectl delete cm avx-openbao-configmap -n avx --force
```

Mongo Replicaset Cleanup

If any issues are encountered while running the `installMongoRS` command, ensure that all traces of previous attempts are cleaned up before retrying by following these steps:

1. Remove the Mongo Replicaset using the command below.

```
helm uninstall avx-mongo-rs -n avx
```

2. Delete the StatefulSet resource by executing the command below.

```
kubectl delete sts avx-mongo-mongodb -n avx --force
```

3. Delete the specified Kubernetes services (service objects) in the cluster by executing the command below.

```
kubectl delete svc avx-mongo-mongodb-headless -n avx --force
```

4. Delete the Mongo Replicaset related **PVCs** by executing the commands below.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
kubectl delete pvc avx-mongo-mongodb-persistent-storage-claim-avx-mongo-mongodb-0
 avx-mongo-mongodb-persistent-storage-claim-avx-mongo-mongodb-1 avx-mongo-mongodb-persistent-storage-claim-avx-mongo-mongodb-2 -n avx --force
```

5. Delete the Mongo Replicaset related kubernetes secrets using the commands below.

```
kubectl delete secret mongo-key-rs -n avx --force
```

Downloading Images from AppViewX Repository

Prerequisites

1. Get the source image repository credentials from AppViewX.
2. Configure the docker using the command

```
docker login -u ${USERNAME} -p ${PASSWORD} ${DOCKER_REPOSITORY}
```

3. Configure the respective cloud provider CLI (Azure) and ensure you have access to push docker images to ACR.

The script for image push and pull is as follows:

```
#!/bin/bash
# Pre requisite
# Get the source image repository credentials from Appviewx and then configure docker using `docker login -u ${USERNAME} -p ${PASSWORD}
 ${DOCKER_REPOSITORY}`
# Configure the respective cloud provider cli (aws/az/gcloud) and make sure that the user has access to push docker images to ACR/ECR/GCR.


appVersion=$1 # App image version. E.g: 24.3.0.0
targetImageRegistry=$2 # Image resgistry name


# Validate required inputs
if [ -z "$appVersion" ] || [ -z "$targetImageRegistry" ];then
{
    echo "Please provide script parametes as ./script.sh <appVersion> <targetImageRegistry>"
    exit
}
fi


# Set the registry login
if echo $targetImageRegistry | grep -iq "amazonaws";then
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
{
    registryProvider="ecr"
    region=$(echo $targetImageRegistry | cut -d "." -f4)
    aws ecr get-login-password --region $region | docker login --username AWS --password-stdin $targetImageRegistry
}
elif echo $targetImageRegistry | grep -iq "azurecr";then
{
    registryProvider="acr"
    az acr login -n $targetImageRegistry
}
elif echo $targetImageRegistry | grep -iq "gcr";then
{
    registryProvider="gcr"
    gcloud auth print-access-token | docker login -u oauth2accesstoken \
    --password-stdin $(echo $targetImageRegistry | cut -d '/' -f2)
}
else
{
    echo "Unknown registry provider"
    exit 2
}
fi


# Image tag mappings
imageTags='[
   {
      "imageName": "avx-cloud-managedservice-mks",
      "tagVersion": "appVersion",
      "upload": true
   },
   {
      "imageName": "avx-cloud-web",
      "tagVersion": "appVersion",
      "upload": true
   },
   {
      "imageName": "avx-cloud-gateway",
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
    "tagVersion": "appVersion",
    "upload": true
},
{
    "imageName": "avx-platform-report-generator",
    "tagVersion": "appVersion",
    "upload": true
},
{
    "imageName": "mongo-init",
    "tagVersion": "appVersion",
    "upload": true
},
{
    "imageName": "avx-cloud-mongoseed",
    "tagVersion": "appVersion",
    "upload": true
},
{
    "imageName": "avx-python-sandbox",
    "tagVersion": "appVersion",
    "upload": true
},
{
    "imageName": "avx-mid-server-base",
    "tagVersion": "appVersion",
    "upload": true
},
{
    "imageName": "alpine",
    "tagVersion": "3.22.0",
    "upload": true
},

{
    "imageName": "pilot",
    "tagVersion": "1.26.1",
```


Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
    "upload": true
},
{
    "imageName": "proxyv2",
    "tagVersion": "1.26.1",
    "upload": true
},
{
    "imageName": "openbao",
    "tagVersion": "2.2.2",
    "upload": true
},
{
    "imageName": "redis",
    "tagVersion": "8.0.2",
    "upload": true
},
{
    "imageName": "kafka",
    "tagVersion": "0.45.0-kafka-3.9.0",
    "upload": true
},
{
    "imageName": "operator",
    "tagVersion": "0.45.0",
    "upload": true
},
{
    "imageName": "kube-metrics-adapter",
    "tagVersion": "v0.2.3",
    "upload": true
},
{
    "imageName": "kibana",
    "tagVersion": "9.0.2",
    "upload": true
},
```


Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
{
    "imageName": "grafana",
    "tagVersion": "11.6.2",
    "upload": true
},
{
    "imageName": "filebeat",
    "tagVersion": "9.0.2",
    "upload": true
},
{
    "imageName": "logstash",
    "tagVersion": "9.0.2",
    "upload": true
},
{
    "imageName": "logstash-syslog",
    "tagVersion": "9.0.2",
    "upload": true
},
{
    "imageName": "elasticsearch",
    "tagVersion": "9.0.2",
    "upload": true
},
{
    "imageName": "elasticsearch-insight",
    "tagVersion": "9.0.2",
    "upload": true
},
{
    "imageName": "prometheus",
    "tagVersion": "v3.5.0",
    "upload": true
},
{
 "imageName" : "backup-utility-image",
```


Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
    "tagVersion": "v2024.3.0",
      "upload": true
   },
   {
    "imageName" : "alertmanager",
    "tagVersion": "v0.28.1",
      "upload": true
   },
   {
    "imageName" : "node-exporter",
    "tagVersion": "v1.9.1",
      "upload": true
   },
   {
    "imageName" : "metrics-server",
    "tagVersion": "v0.7.2",
      "upload": true
   },
   {
    "imageName" : "kube-state-metrics",
    "tagVersion": "v2.15.0",
      "upload": true
   },
   {
    "imageName" : "redis_exporter",
    "tagVersion": "v1.73.0",
      "upload": true
   }
]'

for row in $(echo "${imageTags}" | jq -r '.[] | @base64'); do
   _jq() {
    echo ${row} | base64 --decode | jq -r ${1}
   }
   imageUpload=$(_jq '.upload')
   tagVersion=$(_jq '.tagVersion')
   if [ $imageUpload == "true" ];then
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
  {
    if [ "${tagVersion}" == "appVersion" ];then
    {
      docker pull images.appviewx.com/appviewx/$(_jq '.imageName'):$appVersion
      docker tag images.appviewx.com/appviewx/$(_jq '.imageName'):$appVersion $targetImageRegistry/appviewx/$(_jq '.imageName'):$appVersion
      docker push $targetImageRegistry/appviewx/$(_jq '.imageName'):$appVersion
    }
    else
    {
      docker pull images.appviewx.com/appviewx/$(_jq '.imageName'):$(_jq '.tagVersion')
      docker tag images.appviewx.com/appviewx/$(_jq '.imageName'):$(_jq '.tagVersion') $targetImageRegistry/appviewx/$(_jq '.imageName'):$(_jq
'.tagVersion')
      docker push $targetImageRegistry/appviewx/$(_jq '.imageName'):$(_jq '.tagVersion')
    }
    fi
  }
  fi
done
```

Execute the Image Push-Pull Script

To execute the above image push-pull script, run the command

```
./avx_image_pull_push.sh <image-tag> <targetImageRegistry>
```

Note: Contact AppViewX Support for the <**image-tag**>.

Utility Tools and Functions

Updating Ingress Host in the Config Map

Customers can now update their own ingress host directly in the config map using the updateIngressHosts utility below. If they are using an external load balancer and wish to modify the ingress, it can be done using the following command:

```
./appviewxctl updateIngressHosts --config .appviewxctl.yaml
```

On executing the above utility function, you will be prompted with the following:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- *Enter the loadbalancer URL (Ingress Host):*
- *Enter the ingress port (default 443):* - If you're using the default port 443, simply press Enter. Otherwise, enter the desired port number and then press Enter.

```
eksuser@bastion-hsot-qa-1:~/mar6$ ./appviewxctl updateIngressHosts --config .appviewxctl.yaml
[Init] Using log file at [/home/eksuser/mar6/appviewxctl-2220583372.log] to dump logs
[Init] Initialise persistent flag config
[Init] Using config file
INFO[2025-03-10T11:55:47Z] Starting ingress host update process.
Enter the loadbalancer URL (Ingress host): hello.world.com
INFO[2025-03-10T11:55:59Z] Ingress host provided: hello.world.com
Enter the ingress port (default 443):
INFO[2025-03-10T11:56:02Z] No port provided. Using default port: 443
INFO[2025-03-10T11:56:02Z] Ingress port provided: 443
INFO[2025-03-10T11:56:02Z] Value has been updated in configmap          function="Patch Configmap" name=avx-common-config namespace=avx
[Install/Upgrade] key value pair has been updated in configmap avx-common-config
INFO[2025-03-10T11:56:02Z] Value has been updated in configmap          function="Patch Configmap" name=avx-common-config namespace=avx
[Install/Upgrade] key value pair has been updated in configmap avx-common-config
INFO[2025-03-10T11:56:02Z] ConfigMap updated successfully.
INFO[2025-03-10T11:56:02Z] Restarting updated plugin pods: [avx-config-server avx-commons avx-crontab avx-platform-amc avx-platform-core avx-platform-gateway avx
-platform-aep-gateway avx-platform-logforwarding avx-platform-queue avx-platform-report-generator avx-platform-web avx-subsystems avx-subsystems-sync avx-vendor-
cert-acme-agent avx-vendor-cert-est-agent avx-vendor-cert-cmp-agent avx-vendor-cert-intune-agent avx-vendor-cert-network-discovery avx-vendor-cert-scep-agent avx
-vendors avx-visual-page-builder avx-platform-hsm avx-saas-proxy avx-saas-proxy-web-socket avx-subsystem-codesigning avx-python-sandbox-sync avx-python-sandbox]
INFO[2025-03-10T11:56:08Z] Deleted plugin pods successfully: [avx-config-server-6b66b85cfc-f942s avx-config-server-6b66b85cfc-gzmfh avx-config-server-6b66b85cfc-
nr2jr avx-crontab-6fd765d945-cgrx4 avx-platform-aep-gateway-7d898c647c-7zznr avx-platform-aep-gateway-7d898c647c-jp2h4 avx-platform-aep-gateway-7d898c647c-m6rdz
avx-platform-core-75784b8cc7-25r6n avx-platform-core-75784b8cc7-jz9w6 avx-platform-core-75784b8cc7-rhbqj avx-platform-gateway-789dfd4cb6-ck2b2 avx-platform-gatew
ay-789dfd4cb6-ckjkp avx-platform-gateway-789dfd4cb6-hwcpk avx-platform-hsm-59f5b87648-5b4pr avx-platform-hsm-59f5b87648-7cgdj avx-platform-hsm-59f5b87648-slb6g a
vx-platform-logforwarding-bc976845c-8zbhj avx-platform-logforwarding-bc976845c-jt7vp avx-platform-logforwarding-bc976845c-kjwhg avx-platform-queue-959dd9978-9z2q
s avx-platform-queue-959dd9978-c2fz8 avx-platform-queue-959dd9978-wwjc2 avx-platform-report-generator-59cf7d57d5-rgh6t avx-platform-web-57ddfd7f54-5bpjr avx-plat
form-web-57ddfd7f54-wspxx avx-platform-web-57ddfd7f54-x64zk avx-python-sandbox-9758f4c7c-4lstv avx-python-sandbox-9758f4c7c-68drh avx-python-sandbox-9758f4c7c-7l
jcz avx-python-sandbox-9758f4c7c-999fl avx-python-sandbox-9758f4c7c-cltg8 avx-python-sandbox-9758f4c7c-clw8h avx-python-sandbox-9758f4c7c-f6xnt avx-python-sandbo
x-9758f4c7c-j48vx avx-python-sandbox-9758f4c7c-l26zb avx-python-sandbox-9758f4c7c-p6srx avx-python-sandbox-9758f4c7c-qpk42 avx-python-sandbox-9758f4c7c-v6p58 avx
-python-sandbox-sync-798f45fbff-6wwkk avx-python-sandbox-sync-798f45fbff-78xlx avx-python-sandbox-sync-798f45fbff-7bl2b avx-python-sandbox-sync-798f45fbff-hmqsc
avx-python-sandbox-sync-798f45fbff-jr5sg avx-python-sandbox-sync-798f45fbff-k5r5m avx-python-sandbox-sync-798f45fbff-krp6s avx-python-sandbox-sync-798f45fbff-nhq
jk avx-python-sandbox-sync-798f45fbff-sbvsc avx-python-sandbox-sync-798f45fbff-t7vqj avx-python-sandbox-sync-798f45fbff-x55cw avx-python-sandbox-sync-798f45fbff-
zxhbl avx-saas-proxy-79874d8df5-6qt28 avx-saas-proxy-79874d8df5-fbdds avx-saas-proxy-79874d8df5-tvg2q avx-saas-proxy-web-socket-5f57cf774b-mqgmf avx-saas-proxy-w
eb-socket-5f57cf774b-nmp7x avx-saas-proxy-web-socket-5f57cf774b-pjktb avx-subsystem-codesigning-67c6cc76bf-br84d avx-subsystem-codesigning-67c6cc76bf-k5nt2 avx-s
ubsystem-codesigning-67c6cc76bf-mz2b6 avx-subsystems-57985d5dcb-7hc4n avx-subsystems-57985d5dcb-865sp avx-subsystems-57985d5dcb-tj52b avx-subsystems-sync-6c6846d
c79-952x8 avx-subsystems-sync-6c6846dc79-gl6m6 avx-subsystems-sync-6c6846dc79-n6l8v avx-subsystems-sync-6c6846dc79-nbs2v avx-subsystems-sync-6c6846dc79-qfmpv avx
-subsystems-sync-6c6846dc79-swz7b avx-vendor-cert-acme-agent-5845745995-7p6mg avx-vendor-cert-acme-agent-5845745995-8hgfq avx-vendor-cert-acme-agent-5845745995-b
6twh avx-vendor-cert-cmp-agent-6cd576878-2s79d avx-vendor-cert-cmp-agent-6cd576878-qdf8n avx-vendor-cert-cmp-agent-6cd576878-sdxdr avx-vendor-cert-est-agent-7bdc
```

Note: After config map is updated, all the plug-in pods will get restarted automatically.

Updating Istio Certificates in the tls-credential secret

Customers can now upload/update their own istio certificates using the updateIngressCerts utility provided below:

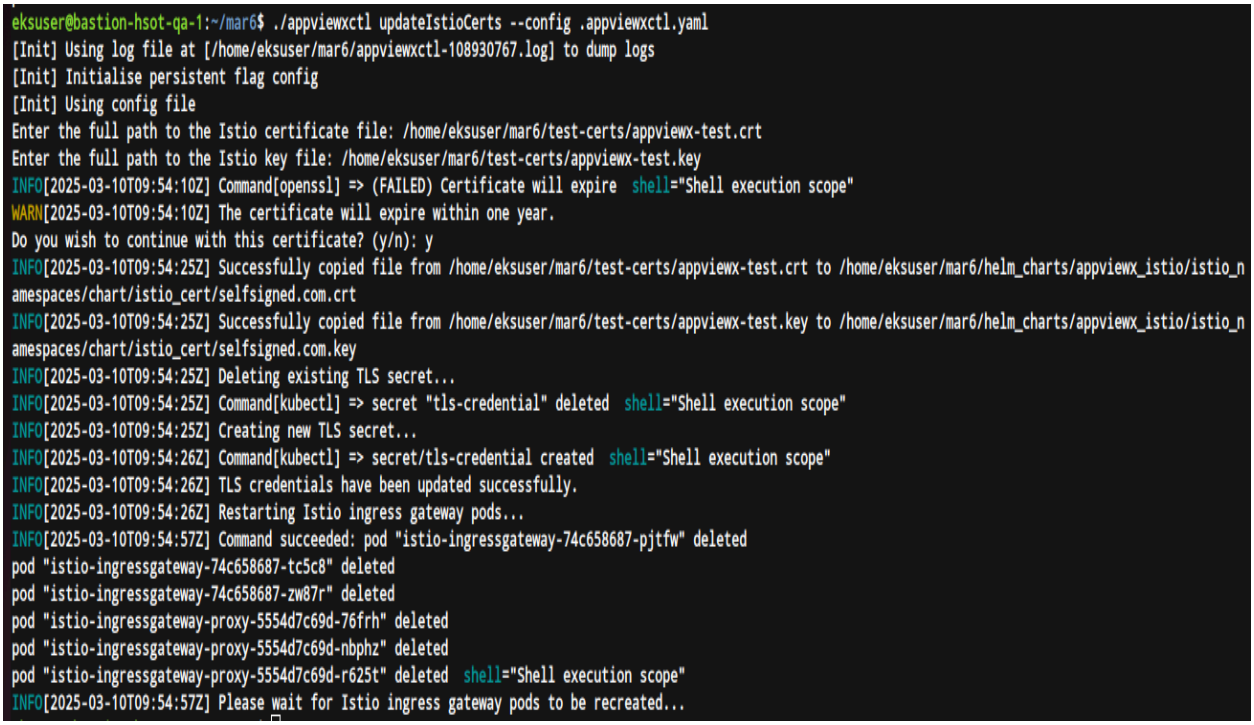
```
./appviewxctl updateIngressCerts --config .appviewxctl.yaml
```

On executing the above utility function, you will be prompted with the following:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- *Enter the full path to the Istio certificate file:* - Enter the directory path including the certificate file.
- *Enter the full path to the Istio key file:* - Enter the directory path including the certificate key.

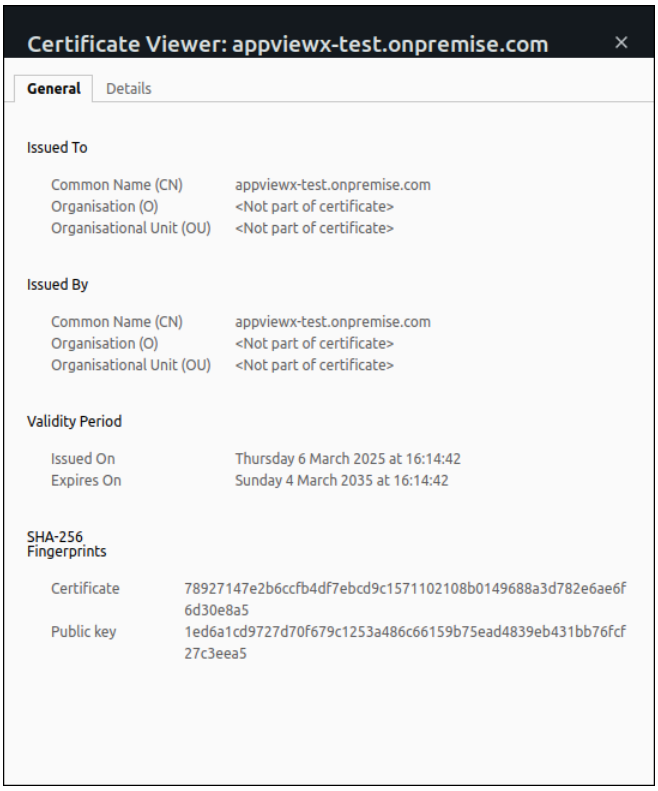
- If the entered certificate is due to expire within one year then you will prompted with *Do you wish to continue with this certificate? (y/n):*



The entered certificate file and key are placed in the **istio_certs** folder. The existing tls-credential secret is deleted and a new one is created using the entered certificate and key. The exiting istio ingress gateway and the istio ingress gateway pods are deleted. The istio ingress gateway pods are then recreated and once they are up and running the utility program ends.

The certificates can be verified on the browser as shown below.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

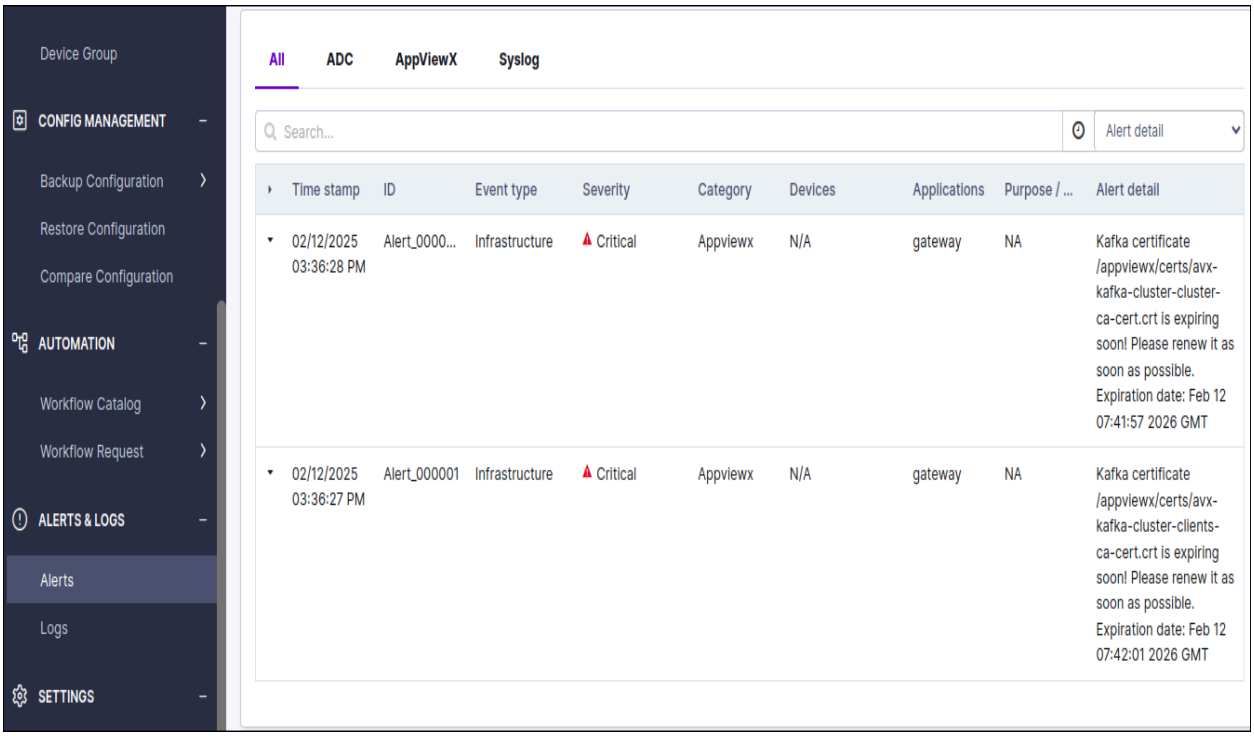


Kafka Cert Utility

If the Kafka certificates are nearing expiry, a weekly cron job is available that sends alerts to the gateway if they expire within the next 3 months. The cron job alert details can be viewed in the product by accessing the product menu, **ALERTS & LOGS** > **Alerts**.

Note: To make the alerts section visible on the gateway, first enable the following checkbox: **Identity → Role → Select appropriate role → Authorized Functions → Platform → Observe and Explore - Alerts (Checkbox).**

Copyright © 2025 AppViewX Inc. All Rights Reserved.



Utility function to renew kafka certs: Users will be able to renew kafka certs with a new utility, by executing the single command below.

```
./appviewxctl renewKafkaCerts --config .appviewxctl.yaml
```

Log Collection

The log collection script is designed to collect logs in a managed kubernetes setup. It is compatible with all mk8s environments, regardless of the cloud provider, and also retrieves logs from the backup folder.

Prerequisites: The bastion host should have kubectl and have access to the compute cluster to collect the logs.

1. Login to bastion host.
2. Download the helper tarball from the release portal.
3. Execute the following command:

```
./mk8s_collect_logs.sh
```

> **Note:** The script can be run anywhere in the bastion host.

4. Fill the interactive questions as follows:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

a. ***Do you want to collect logs for all plugins?***

- Answer **no** if you want to collect selective plugin logs. You will then be prompted to enter the names of the desired plugins.
- Answer **yes** to collect logs for all enabled plugins.

> **Note:** To view all the enabled plugins in the setup, run:
>
> ```
> ./mk8s_collect_logs.sh -operation get-enabled-plugins
> ```

b. ***Do you want to collect mongo logs?***

- Answer **yes** to include MongoDB logs in the collection.
- Answer **no** to exclude MongoDB logs.

c. ***Enter the directory to save logs:***

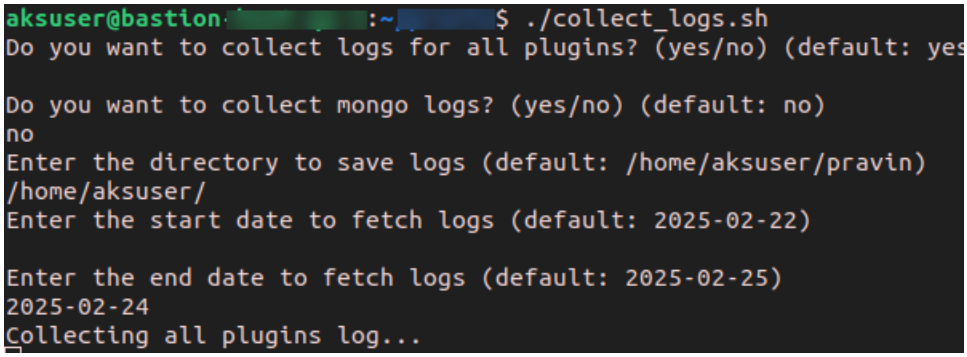
- The collected logs will be saved in the provided directory. If no directory is specified, the default path will be the directory where the script is executed.

d. ***Enter the start date to fetch logs:***

- Enter the date from which you'd like to begin collecting logs.

e. ***Enter the end date to fetch logs:***

- Enter the date until which you want to collect logs.



Alternative options: Passing Parameters via Command Line

Alternatively, you can provide the inputs as parameters instead of entering them interactively. To view the available parameters, run the command below.

```
./mk8s_collect_logs.sh --help
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Uninstall and Cleanup

The process of uninstalling requires one to navigate to the installer directory and execute the following command

```
./appviewxctl uninstall --config .appviewxctl.yaml
```

The following messages are displayed after the uninstall command is executed successfully.

```
./appviewxctl uninstall --config .appviewxctl.yaml

[Init] Using log file at [/avx/appviewxctl-3196327299.log] to dump logs
[Init] Initialise persistent flag config
[Init] Using config file
[Uninstall] Uninstalling appviewx application
[Uninstall] Uninstalling Appviewx application helm chart
[Uninstall] Uninstalling application backup helm chart
[Uninstall] Uninstalling Infra application helm chart
[Uninstall] Uninstalling Third party application helm chart
[Uninstall] Uninstalling IstioOperator from the cluster
[Uninstall] Uninstalling PVCs from the avx namespace
[Uninstall] Uninstalling Pre-requisite helm chart
[Uninstall] Uninstalling Appviewx installed namespaces
[Uninstall] Successfully uninstalled appviewx application and all the related
```

Note: In the Managed K8s environments removal of PVCs do not occur at times as it may require patching PVCs first before deletion. This may cause certain error messages to display, indicating that PVC has changed. In case such an error occurs, re-run the above command to solve the issue and uninstall the application.

Sometimes the namespaces take a longer time to be removed. Hence, post installation, check if namespaces are in the terminating state (use the command: **kubectl get namespace**). If any namespace is in the terminating state, manually remove the namespaces by executing the commands below:

```
kubectl get namespace "istio-operator" -o json | tr -d "\n" | sed "s/\"finalizers\": \[[^]]\+\]/\"finalizers\": []/" | kubectl replace --raw /api/v1/namespaces/istio-operator/finalize -f - 2>/dev/null
```

```
kubectl get namespace "istio-system" -o json | tr -d "\n" | sed "s/\"finalizers\": \[[^]]\+\]/\"finalizers\": []/" | kubectl replace --raw /api/v1/namespaces/istio-system/finalize -f - 2>/dev/null
```

```
kubectl get namespace "avx" -o json | tr -d "\n" | sed "s/\"finalizers\": \[[^]]\+\]/\"finalizers\": []/" | kubectl replace --raw /api/v1/namespaces/avx/finalize -f - 2>/dev/null
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
kubectl get namespace "avx-kafka" -o json | tr -d "\n" | sed "s/\"finalizers\": \[[^]]\+\]/\"finalizers\": []/" | kubectl replace --raw /api/v1/namespaces/avx-kafka/finalize -f - 2>/dev/null
```

```
kubectl delete ns istio-operator --force 2>/dev/null
```

```
kubectl delete ns istio-system --force 2>/dev/null
```

```
kubectl delete ns avx --force 2>/dev/null
```

```
kubectl delete ns avx-kafka --force 2>/dev/null
```

Troubleshooting

Error	Resolution
403: Access Forbidden - License not available. This error occurs when only the managed kubernetes cluster is restarted every morning.	In case of managed kubernetes, when the cluster is restarted it is recommend to restart all pods.

AppViewX Install and Upgrade for EKS

This guide provides the prerequisites and the procedure for installing, upgrading, and accessing the AppViewX application..

- AppViewX Architecture
- Architecture Overview
- AppViewX Deployment Architecture
- Managed Kubernetes Architecture
- EKS Components
- Prerequisites
- Install AppViewX in Managed Kubernetes
- Upgrade AppViewX in Managed Kubernetes
- Downloading Images from AppViewX Repository
- Utility Tools and Functions
- Log Collection

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- Uninstall and Cleanup
- Troubleshooting

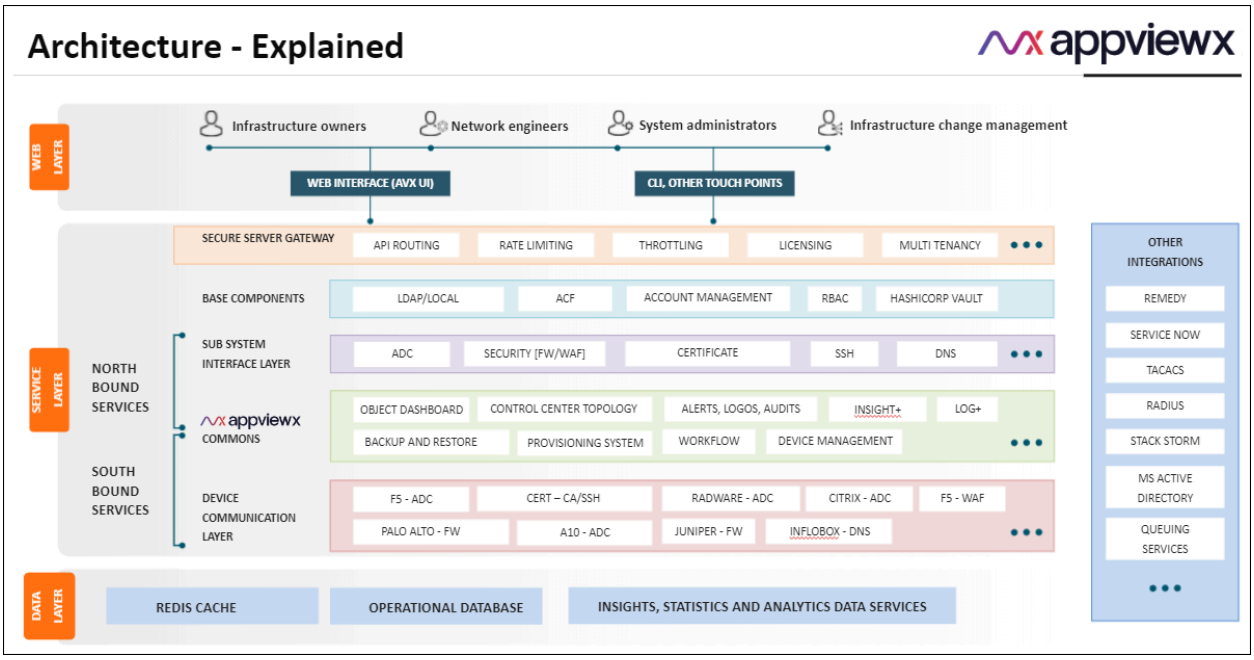
AppViewX Architecture

Architecture Explained

AppViewX is designed based on the microservice architecture and is deployed on Kubernetes—an open-source platform for deploying and managing containers.

The microservice architecture of AppViewX makes it easier to move to containerized workloads and the containers being orchestrated using Kuberneters.

Kubernetes provides container runtime, orchestration, self-healing mechanisms, service discovery and load balancing and it is used for the deployment, scaling, management, and composition of application containers across clusters.



Benefits of AppViewX Architecture

In order to optimally utilize the resources, AppViewX has adopted Kubernetes to achieve higher security by adopting a zero trust network model. The features of AppViewX coupled with Kubernetes are given below.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- **Auto scaling** - AppViewX services can have a custom throttling capability based on pre-configured memory configuration per API. This will enable AppViewX services to utilize (scale up) resources optimally as the demand surges and scale down when not in use. This will help to horizontally scale the vendor components on demand and optimize the resource usage.
- **Resiliency** - There is no guarantee that AppViewX services may run without any interruptions and they are bound to fail. Kubernetes keeps deployments healthy by restarting containers that have failed, by killing and replacing unresponsive containers based on health checks. This helps to mitigate the common pain point of the application's upkeep process.
- **Security** - AppViewX architecture is designed around the concept of zero trust network model to enforce tighter security within the Kubernetes cluster. This means no one is trusted by default and requires verification to gain access to the services.

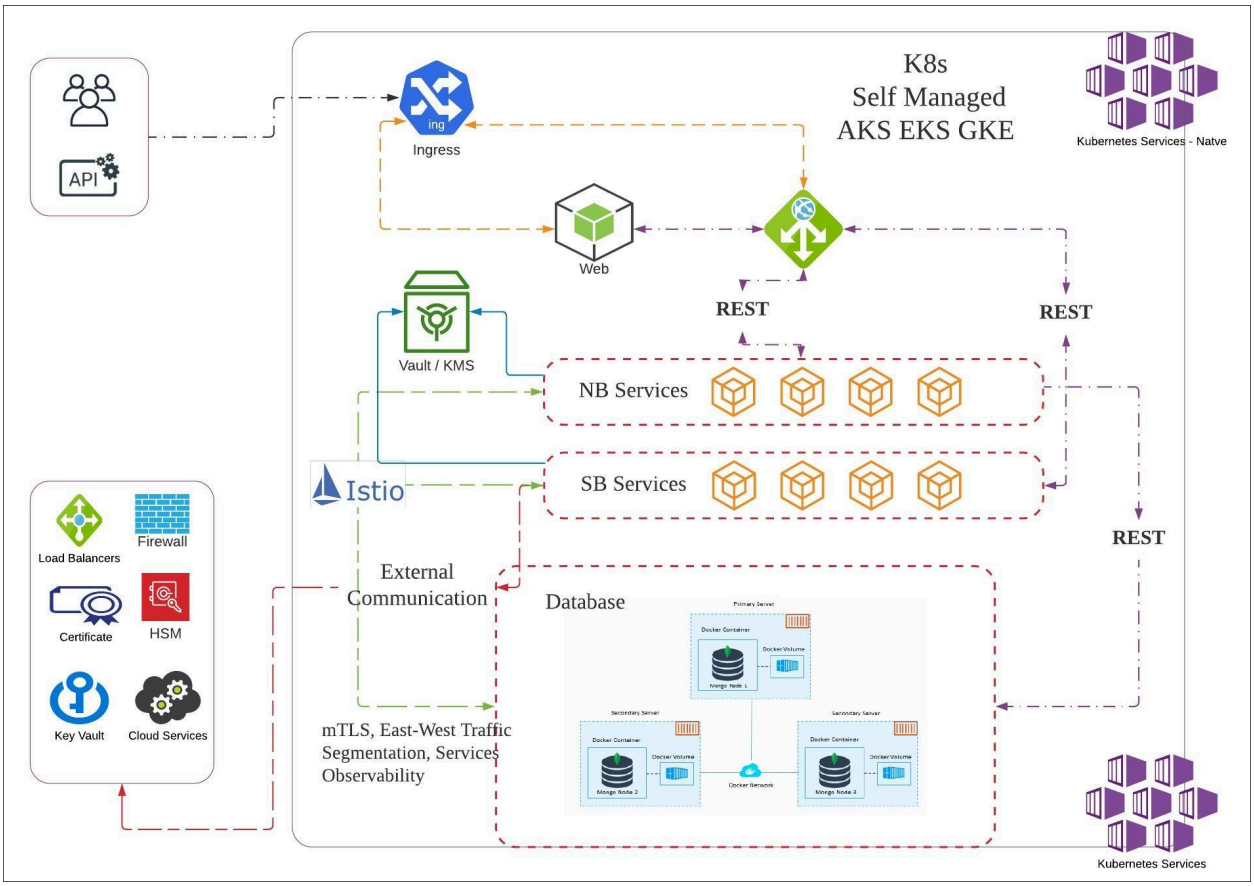
Architecture Overview

AppViewX Kubernetes Architecture

AppViewX workloads are containerized workloads running as microservices and these containers are orchestrated by managed Kubernetes services. Users can prefer the managed k8s platform of their choice.

AppViewX supports deployment on all the three public clouds AWS, Azure and GCP (Google Cloud Platform) using their managed kubernetes engine / services EKS, AKS and GKE specifically.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



Benefits of AppViewX Architecture

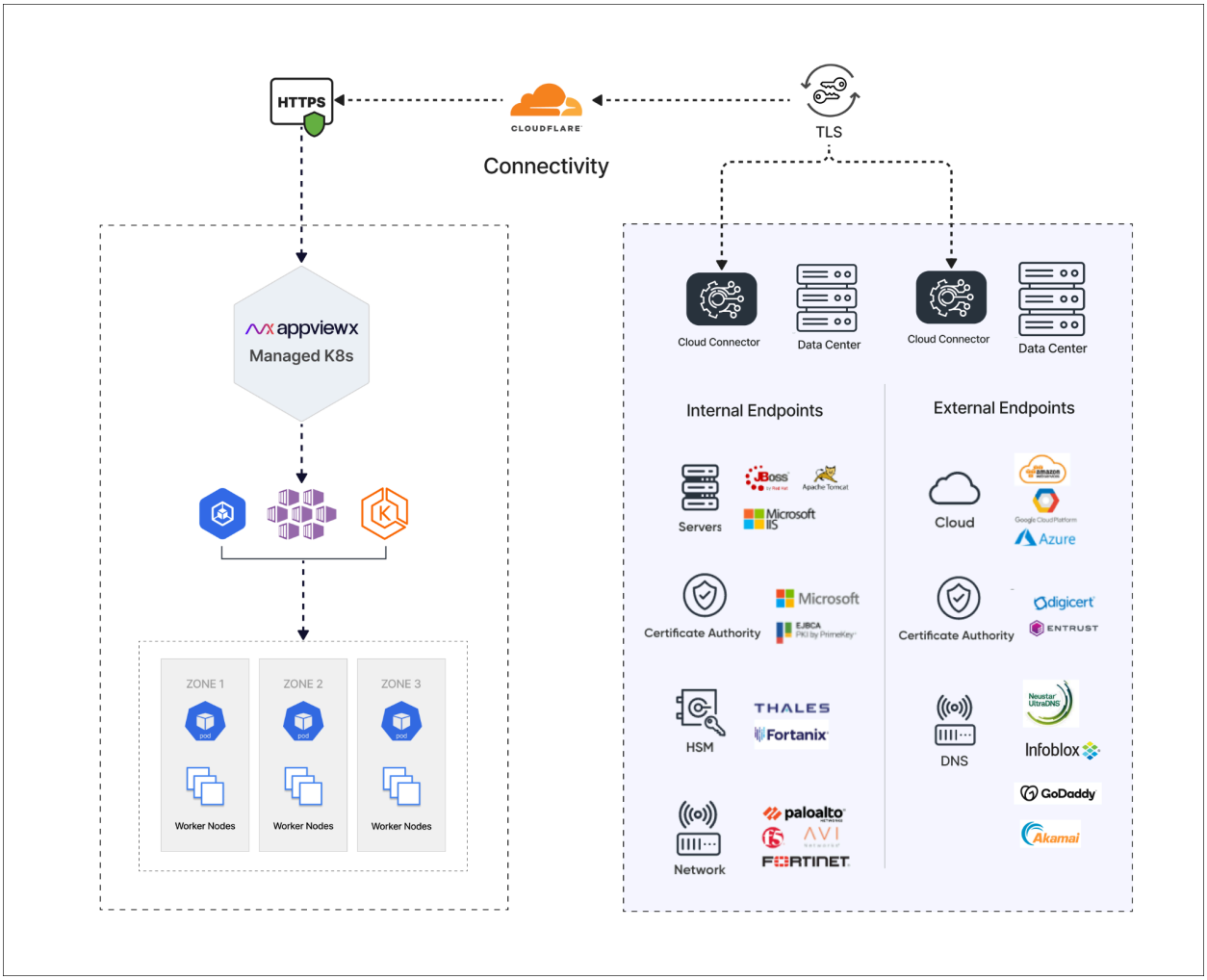
In order to optimally utilize the resources, AppViewX has adopted Kubernetes to achieve higher security by adopting a zero trust network model. The features of AppViewX coupled with Kubernetes are given below.

- **Auto scaling** - AppViewX services can have a custom throttling capability based on pre-configured memory configuration per API. This will enable AppViewX services to utilize (scale up) resources optimally as the demand surges and scale down when not in use. This will help to horizontally scale the vendor components on demand and optimize the resource usage.
- **Resiliency** - There is no guarantee that AppViewX services may run without any interruptions and they are bound to fail. Kubernetes keeps deployments healthy by restarting containers that have failed, by killing and replacing unresponsive containers based on health checks. This helps to mitigate the common pain point of the application's upkeep process.
- **Security** - AppViewX architecture is designed around the concept of zero trust network model to enforce tighter security within the Kubernetes cluster. This means no one is trusted by default and requires verification to gain access to the services.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

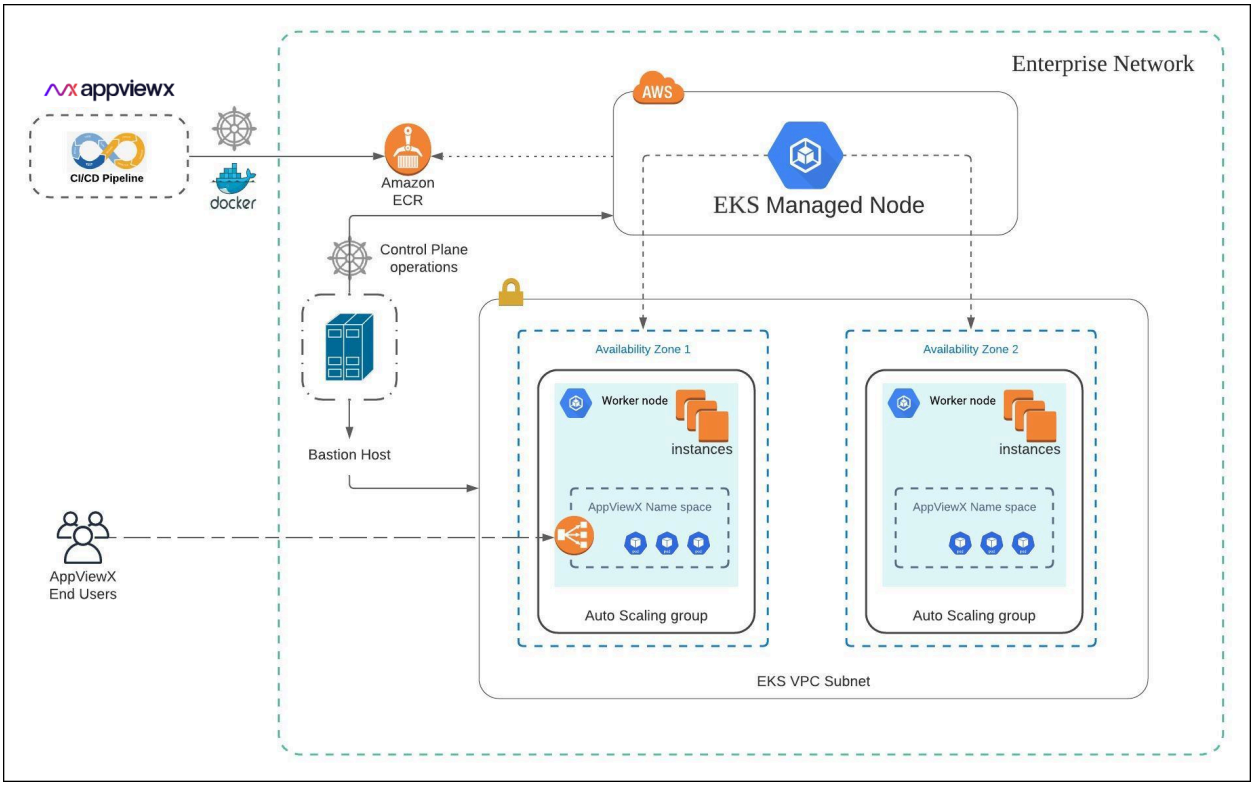
AppViewX Deployment Architecture

The figure below shows a standard AppViewX deployment architecture model via managed Kubernetes service for AKS.



Copyright © 2025 AppViewX Inc. All Rights Reserved.

EKS Deployment Model



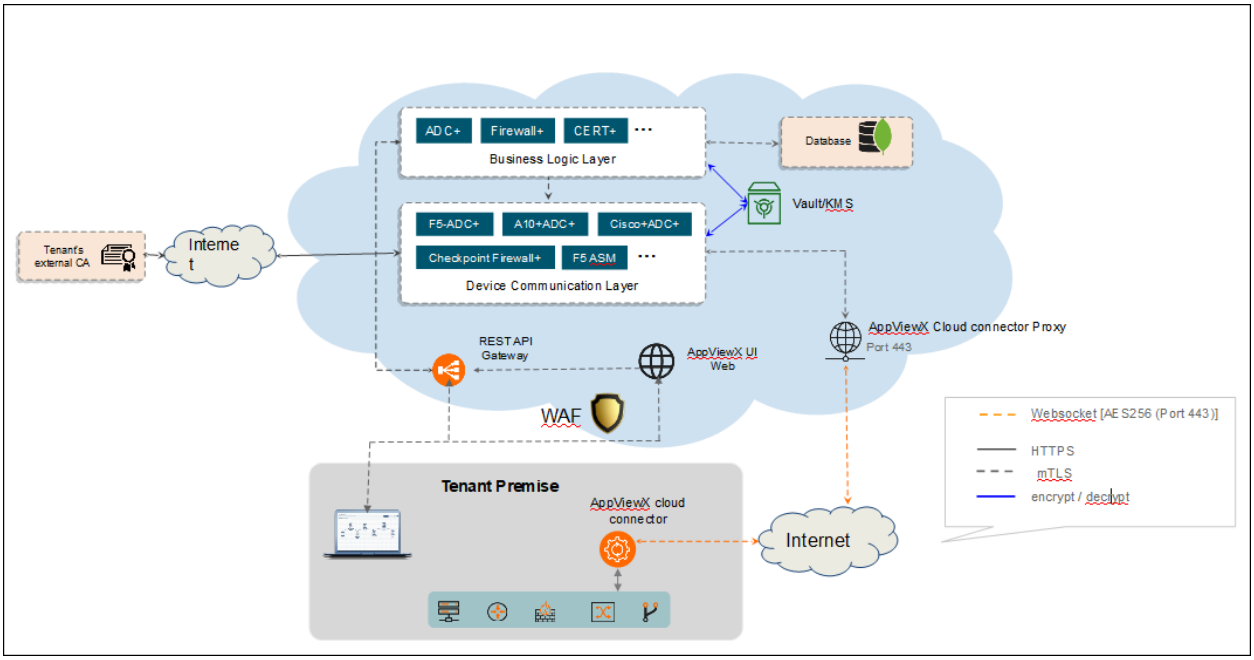
Cloud Connector

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 and your enterprise network without requiring any complex network or infrastructure configuration

Key features of the AppViewX Cloud Connector:

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.



For more details on cloud connectors refer to AppViewX Cloud Connector User Guide.

Note: The below steps have to be performed in all the cloud connector host machines after the 2022.1.0FP2 to FP3 patch upgrade and before the FP3 cloud connector upgrade.

1. Navigate to the installation path in the cloud connector host machine.
2. Execute the following command:

```
./deps/tools/k3s kubectl get deploy avx-mid-server-starter -n cc -o yaml > starter.yaml && sed -i "s/-Xmx2560m/-Xmx4g/g" starter.yaml

&& ./deps/tools/k3s kubectl replace -f starter.yaml
```

Managed Kubernetes Architecture

Managed Kubernetes clusters are composed of the following main components — a control plane and worker nodes. Each cluster runs in its own, fully managed Virtual Private Cloud (VPC).

- The **control plane** is composed of three master nodes, each running in a different Availability Zone to ensure high availability. Incoming traffic directed to the Kubernetes API passes through the respective cloud service load balancer.
- The **worker nodes** run on virtual instances located in a VPC. Managed kubernetes service engine provides managed node groups with automated lifecycle management. This lets users automatically create, update, or shut down nodes with one operation.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Managed Kubernetes service scales the Kubernetes control plane across multiple Availability Zones of the public cloud to ensure high availability and it automatically scales control plane instances based on load, detects and replaces unhealthy control plane instances, and automatically patches the control plane.

Managed Kubernetes workload instances are deployed in multiple availability zones within the region. Each instance has replicas of the services and nodes which exist across all the virtual instances.

Each zone or instance has an active pod listening to other instances. In case of a failure in any instance, the active pod ensures seamless functioning of the application by activating the nodes from any other working cluster.

EKS Components

The following EKS components are utilized by AppViewX:

- Storage bucket for storing mongodb and vault backups
- Amazon Kubernetes engine

Prerequisites

The following prerequisites must be met before the installation process.

- Managed Kubernetes Version Support Matrix
- Disks Used for AppViewX Installation
- AppViewX Docker Images
- AppViewX Helm Charts
- Bastion Host Setup
- EKS Cluster
- AWS S3 Bucket
- Configuring CSI

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Managed Kubernetes Version Support Matrix

Public Cloud	
Mode of Deployment	**Amazon**
Release, Vendor, & Product Support **AppViewX v2024.3.0.0**	
Managed K8s Deployment (EKS)	
K8s version 1.33	Yes

Disks Used for AppViewX Installation

Disks Used

Volume	Size	Quantity
logs volume	10Gi	1
avx-kafka	20Gi	3
zookeeper	20Gi	3
openbao	10Gi	3
mongodb	256Gi	3
redis	5Gi	3

If a third party is installed, the values are as follows:

Disks Used (Third Party)

Volume	Size	Quantity
Elasticsearch-ELK	10Gi	1
Elasticsearch-Insight	10Gi	1

Copyright © 2025 AppViewX Inc. All Rights Reserved.

AppViewX Docker Images

AppViewX Docker images are hosted in a private registry https://images.appviewx.com. These images can be pulled using an authentication token (contact AppViewX Support, *help@appviewx.com* for the authentication token) and can be hosted in the private or public repository at the customer end.

The list of docker images are

- <registry link>/appviewx/pilot:1.26.1
- <registry link>/appviewx/proxyv2:1.26.1
- <registry link>/appviewx/openbao:2.2.2
- <registry link>/appviewx/redis:8.0.2
- <registry link>/appviewx/mongo-init:<tag>
- <registry link>/appviewx/avx-cloud-gateway:<tag>
- <registry link>/appviewx/avx-cloud-web:<tag>
- <registry link>/appviewx/avx-cloud-mongoseed:<tag>
- <registry link>/appviewx/avx-cloud-managedservice-mks:<tag>
- <registry link>/appviewx/avx-platform-report-generator:<tag>
- <registry link>/appviewx/avx-python-sandbox:<tag>
- <registry link>appviewx/avx-mid-server-base:<tag>
- <registry link>/appviewx/kafka:0.46.0-kafka-3.9.0
- <registry link>/appviewx/operator:0.46.0
- <registry link>/appviewx/alpine:3.22.0
- <registry link>/appviewx/kube-metrics-adapter:v0.2.3
- <registry link>/appviewx/kube-state-metrics:v2.15.0
- <registry link>/appviewx/backup-utility-image:v2024.3.0
- <registry link>/appviewx/prometheus:v2.53.4
- <registry link>/appviewx/metrics-server:v0.7.2
- <registry link>/appviewx/elasticsearch:9.0.2
- <registry link>/appviewx/elasticsearch-insight:9.0.2
- <registry link>/appviewx/filebeat:9.0.2
- <registry link>/appviewx/grafana:11.6.2
- <registry link>/appviewx/kibana:9.0.2
- <registry link>/appviewx/logstash:9.0.2
- <registry link>/appviewx/logstash-syslog:9.0.2
- <registry link>/appviewx/alertmanager:v0.28.1
- <registry link>/appviewx/node-exporter:v1.9.1
- <registry link>/appviewx/redis_exporter:v1.73.0

Copyright © 2025 AppViewX Inc. All Rights Reserved.

The steps to download the images from AppViewX repository are as follows:

1. Get the source image repository credentials from AppViewX Support team.
2. Configure the docker using the command

```
docker login -u ${USERNAME} -p ${PASSWORD} ${DOCKER_REPOSITORY}
```

3. Configure the respective cloud provider CLI (Google cloud) and ensure you have access to push docker images to GCR.
4. To push the docker images, use the helper script provided by AppViewX. Follow the steps below.

   a. Download the artifact Managed-Kubernetes_helper_scripts.tar.gz to the bastion host and extract using the command:

   ```
   tar -xf Managed-Kubernetes_helper_scripts.tar.gz
   ```

   b. Navigate to the extracted directory **mk8s_helper_scripts**.

   ```
   cd mk8s_helper_scripts
   ```

   c. Execute the script **avx_image_pull_push.sh** using the command

   ```
   ./avx_image_pull_push.sh <Image tag> <customer registry url>
   ```

   Note: Replace <Image tag> with **24.3.0.0-1505** and <customer registry url> with the actual values.

AppViewX Helm Charts

The helm charts used by AppViewX for installation are released as a part of the installer. The installer consists of helm charts and an AppViewX utility which helps orchestrate the deployment, patch, upgrade and maintenance of AppViewX across managed kubernetes deployment.

Bastion Host Setup

The following packages must be installed on the bastion host or the host/tool (AWS DevOps) from where the installation is triggered

AWS CLI

To set up the AWS CLI refer to Installing or updating the latest version of the AWS CLI on the AWS documentation website.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Kubectl

To set up Kubectl refer to Install and Set Up kubectl on Linux on the Kubernetes documentation website.

Execute the following commands:

- sudo curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
- sudo chmod +x kubectlmv
- sudo mv ./kubectl /usr/bin/#

Verify installation by executing the command

```
kubectl version
```

Helm

Helm is required only if the deployment is triggered from any other machine instead of the DevOps pipeline. To set up Helm refer to Installing Helm on the Helm documentation website.

Execute the following command:

- curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3
- chmod 700 get_helm.sh
- ./get_helm.sh#

Verify installation by executing the command

```
helm version
```

jq

To set up the jq refer to Download jq on the github.

EKS Cluster

To create an EKS cluster refer to AWS documentation website - Creating an Amazon EKS cluster. Although Microsoft manuals are always up-to-date, the recommended choice to make before creating the cluster is as follows:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- Kubernetes version: 1.31
- User nodepool:
  - **appnodepool**: Three nodes of type **t3.2xlarge** with Auto Scaling disabled
  - **mongonodepool**: Three nodes of type **t3.2xlarge** with Auto Scaling disabled. Add label **mongo=true** and taint **designatedMongo=true:NoSchedule** to the nodepool (to be performed while creating the cluster).

> **Note:** A minimum of 3 availability zones are needed during cluster creation to support the single AZ failover.

- Select multi zones for both Agent nodepool and User nodepool.

Get EKS Cluster kubeconfig

To get the EKS cluster kubeconfig, execute the script below:

```
eksClusterName="<EKS_CLUSTER_NAME>"

aws eks update-kubeconfig --name $eksClusterName --region ap-south-1
```

AWS S3 Bucket

A S3 bucket is required to store

- iControlJar: Directory name is **icontroljar** and the jar has to be placed here.
- MongoDB backup: Directory name should be **mongo-backup**.
- Vault backup: Directory name should be **vault-backup**.
- Axisjar, javax.xml.soap-api: Directory name should be **axisjar**.

Lets understand the different approaches to create a S3 bucket and configure S3 buckets that are accessible by EKS nodes.

Approach 1

In this approach,

1. Create a bucket.
2. Create an IAM policy.
3. Attach this policy to the node groups with read/write access to the bucket.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Approach 2 (Recommended)

A standard and secure way of attaching permissions to pods in kubernetes are the AWS IRSA (IAM role for service account). Users can create a role and policy and then add an annotation to the pod service account. Follow the AWS official documentation website - IAM roles for service accounts.

The steps to create a S3 bucket and configure the IAM roles for IRSA are as follows:

- This step can also be performed using a helper script provided by AppViewX. To use this script follow the steps below.

1. Download the artifact Managed-Kubernetes_helper_scripts.tar.gz to the bastion host and extract using the command:

```
tar -xf Managed-Kubernetes_helper_scripts.tar.gz
```

2. Navigate to the extracted directory **mk8s_helper_scripts**.

```
cd mk8s_helper_scripts
```

3. Edit the file **eks_config.sh** and replace <actualBucketname>, <actualAccountNumber>, <eksClusterName>, and <awsRegionName> with the actual values.

4. Execute the **eks_config.sh** file.

```
bash eks_config.sh
```

Attention: Please enter the actual values in the script below before executing it.

After the script is executed,

- Capture the output **Annotation** which is required in the global utility config. (This value must be added to the sub-field **serviceAccountAnnotation** of the parameter **storageAccess**.)
- Configure the **Authentication to AWS ECR** (AWS Image registry) to pull images from ECR.
- Get the **Image registry** name (images are stored here) and the **AccessKey/secretKey** which are required in the global utility config.

Configuring CSI

To configure CSI

Copyright © 2025 AppViewX Inc. All Rights Reserved.

1. Execute the command below

```
helm repo add aws-ebs-csi-driver https://kubernetes-sigs.github.io/aws-ebs-csi-driver
```

2. Execute the command below

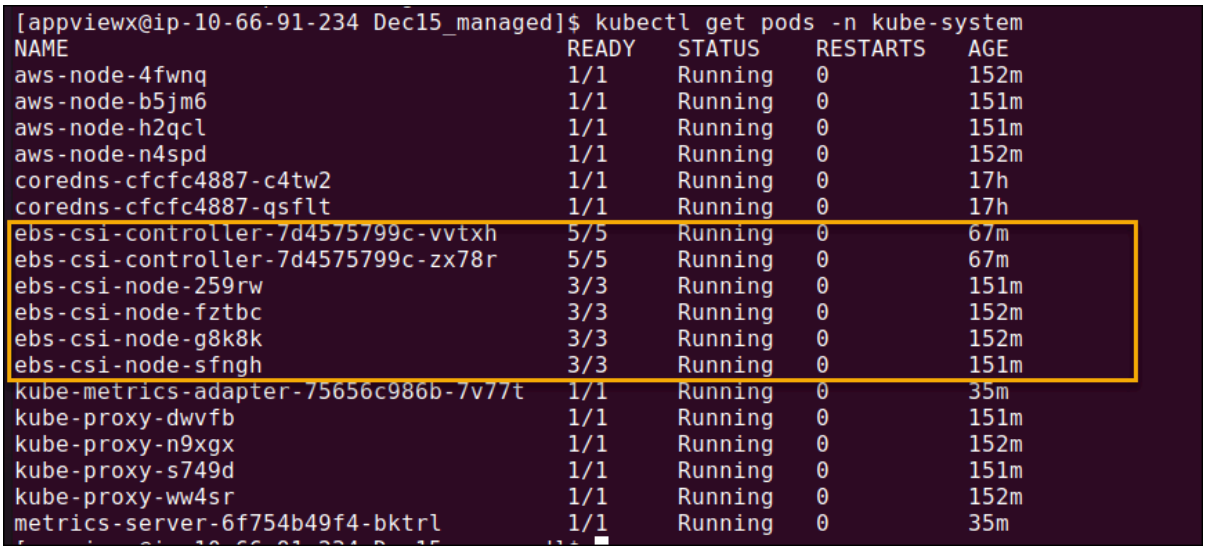
```
helm repo update
```

3. Execute the command below

```
helm upgrade --install aws-ebs-csi-driver --namespace kube-system aws-ebs-csi-driver/aws-ebs-csi-driver
```

4. Verify the status of the pods (CSI) by executing the command:

```
kubectl get pods -n kube-system
```



5. The creation of Amazon EBS CSI plugin IAM role with the AWS CLI is handled in the script contained in the **Approach 2 (Recommended)** section of the topic AWS S3 Bucket. If you consider **Approach 1** in the previous section, refer the following guide Amazon EKS User Guide - Create CSI IAM Role to perform the role creations.

> **Attention:** Ignore Step 4 from the content in the AWS CLI tab if you are not using encrypted volume.

Install AppViewX in Managed Kubernetes

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Migration Strategy

Attention: If you are performing a fresh install, then refer the next sub-topic **Installation Steps**.

To migrate from AppViewX on-prem versions (2022.1.0, 2021.1.0, and 2020.3.0) to Managed Kubernetes, it is important to take a backup of the mongodb and vault in the respective on-prem versions. Before you take the backup, execute the script below.

```
db.profile.update({'_id' : 'installationType'}, {$set : {"value" : "Managed_K8s"}})
```

Note: Refer to the specific version of the release documents from the release portal and perform the backups or contact the AppViewX support team.

After performing the backup, follow the installation steps detailed in the section below. At step 10 of the installation process, ensure to restore the data at this stage.

Installation Steps

This section describes the steps to for installing the AppViewX Stack on EKS.

1. Download the installer from the release portal.
2. Create a directory Managedk8s-installer in the bastion host and extract the installer file**tar -xf installer.tar.gz** in the same directory.
3. Verify that the extracted installer must have the following files
   - appviewxctl (binary)
   - helm_charts (directory of helm charts)
4. Generate the configuration files based on the cloud provider. If the cloud provider is **Amazon**, execute the command below.

   ```
   ./appviewxctl config generate --provider aws
   ```

5. Verify that the execution of the above command creates the configuration files named **.appviewxctl.yaml** in the same location.
6. The file .appviewxctl will be populated with the fields necessary for installation, in particular cloud provider that was provided in the previous command (**-- provider**).
7. Edit the .appviewxctl.yaml file and populate the values as described below:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

appviewxctl.yaml file - Parameters and Description

Parameters	Description of Values
chartPath	The path to the helm_charts which is to be installed. It points to the helm_charts directory extracted in step 3.
configFile	The path to the kube config file to be used by helm and kubectl. If the bastion host is already configured and kube config is under $HOME/.kube directory, then keep this field empty.
install.enableAppBackupCron	Boolean value to enable/disable the backup cronjobs. (True/False). This value is needed for self-managed mongodb only. For atlas backup this has to be scheduled in the atlas dashboard.
install.enablePrivateImagePullSecret	Boolean value to enable image pull secret. Set values as **false** if the cluster already has access to the container registry. Otherwise set it to **true** and fill all the details of the access keys described in below sections.
install.enableThirdPartyInstall	Boolean value (True/False) to determine whether third party monitoring components such as ELK, Monitoring, and Insight needs to be installed.
install.thirdPartyApp.elk	Boolean value to add Elk component. Set to True if it needs to be installed.
install.thirdPartyApp.monitoring	Boolean value to add Monitoring component. Set to True if it needs to be installed.
install.thirdPartyApp.insight	Boolean value to add Insight component. Set to True if it needs to be installed.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Parameters	Description of Values
install.imageRegistry	The URL of the container registry where the images are to be pulled from by the pods.
install.imageTag	The tag of the image that will be used for installation. *Example*: 2023.1.0_FP_750-alpine
install.isSaasEnabled	Boolean value to enable SaaS. This value should be set to **true** for Managed K8s.
install.kafkaCloudConnector	It is a combination of three values. • enable • password • user Set **enable** to ***true*** and keep the user, password fields empty for Managed K8s. *Example* kafkaCloudConnector: enable: true password: "" user: ""
install.mongo	It is a combination of fields specific to the type of mongodb used.
dbIsolation	Boolean value to indicate whether the database isolation is to be enabled. In order for database isolation to work, the following prerequisite must be taken care of while creating the cluster node group. • Add label **mongo=true** and taint **designatedMongo=true:NoSchedule** to the nodepool to be used for mongodb.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Parameters	Description of Values

mongoAtlas	The fields specific to mongodb atlas are as follows:
	• **enable**: Boolean value to decide if mongodb atlas to be used. If set to *false*, a self managed mongodb cluster will be created. If set to *true* mongodb atlas will be used and details of which are to be provided in below mentioned fields.
	• **host**: URL of the mongodb atlas cluster.
	• **password**: password of the mongodb atlas cluster.
	• **user**: username in the mongodb atlas cluster.
	
	Example:
	
	`mongo:`
	`  dbIsolation: false`
	`  mongoAtlas:`
	`  enable: true`
	`  host: "managed-k8s.test.mongodb.net"`
	`  password: "samplepassword"`
	`  user: "user1"`

install.useDockerPrivateRegistry	Set this to **true** if the dockerhub private repository is to be used for pulling the necessary images needed. Otherwise set the value **false** and the container registry ACR, ECR, and GCR will be used based on the cloud provider. If this value is set to *true*, populate the below values, otherwise keep it empty. • **dockerhub.pass**: password to be used for authenticating in the dockerhub private repository. • **dockerhub.username**: username configured in the dockerhub private repository. *Example:*

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Parameters	Description of Values
	useDockerPrivateRegistry: true dockerhub: pass: "testpassword" username: "appviewx"
install.size	The size of the installation. Based on the use cases and number of certs to be managed there different sizes (contact AppViewX for sizing recommendations). The sizes supported are (case sensitive values) • xsmall • small • medium • large • xlarge • custom *Example:* size: small **Note:** The size provided must be taken into cluster creation and nodegroup sizes must be defined accordingly. Follow the same document link above for nodegroup sizes.
install.plugins	The list of plugins that will be installed. Each plugin will have three fields • enable • imageTag • name Set enable to **true** if the plugin is to be installed. If the same image tag is to be used as defined in the global ImageTag keep it **latest** otherwise override with some other tag of your choice.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Parameters	Description of Values

	Example:
	`- enable: true`
	`imageTag: latest`
	`name: avx-config-server`
	To enable Cloud DC support in Managed Kubernetes, set plugins as follows:
	`- enable: true`
	`imageTag: latest`
	`name: avx-mid-server-platform`
	Note: Ensure that **install.isSaasEnabled** and **install.kafkaCloudConnector** are set to **true**.

storageAccess	This parameter contains two sub fields bucketObject and serviceAccountAnnotation as described below. • **bucketObject**: name of the S3 bucket created in the topic AWS S3 Bucket. • **serviceAccountAnnotation**: the Annotation value captured in the **Approach 2 (Recommended)** section of the topic AWS S3 Bucket.
internalLoadBalancer	If set to **true**, all the Loadbalancers will be private and can only be accessed within the VPC else it will be public.
install.enableSftpStorage	Change to true to use SFTP server for mongodb, vault, and icontrol.jar storage. Boolean (Default: false)
install.sftpServerDetails. dbBackupPath	Provide the location of mongodb backup storage directory. String (Default: “”)

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Parameters	Description of Values
install.sftpServerDetails. vaultBackupPath	Provide the location of vault backup storage directory. String (Default: “”)
install.sftpServerDetails. sftpServerUserName	Provide the username of SFTP server. String (Default: “”)
install.sftpServerDetails. sftpServerIp	Provide the sftp server IP. String (Default: “”)
cloudConnectorEnabled	A boolean value (true/false) to denote the cloud connector usage for southbound communications. If a cloud connector is used set the value to **true**.
socketurl	Customers can now configure their own web-socket URL in the MongoDB by updating the `socketurl` parameter in the **.appviewxctl.yaml** file. After installation or upgrade, the specified socket URL will be recorded in the **saas_agent_meta** collection.

The next fields are to be filled with values that must be collected during the cluster creation and setup process and filled as mentioned below.

appviewxctl.yaml file - Parameters and Description (during cluster creation)

Parameters	Description of Values

install.privateImagePullSecret	In this section populate the details of the access keys needed to authenticate and pull the image from the registry. They are not needed if the Dockerhub is used as described above.
	• **accessKeyId**: The access key ID of the ECR.
	• **secretAccessKey**: The secret access key of the ECR.
	• **registry**: The ECR registry URL
	
	Example:
	`accessKeyId: AKIAIOSFODNN7EXAMPLE`
	`secretAccessKey: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`
	`registry: 467889948468.dkr.ecr.ap-south-1.amazonaws.com`

The following fields must be added to integrate the kubernetes cluster to the external vault.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

appviewxctl.yaml file - Parameters and Description (for external vault)

Parameters	Description
install.externalVault.enable	A boolean value (true/false) to denote if the external vault is to be used in the setup. True is to enable the external vault.
install.externalVault.externalVaultAddr	Contains the vault URL and listening port *Example*: https://pm-lxs-node01.lab.appviewx.net:8200
install.externalVault.externalVaultAuthRole	Name of the role created against the access kubernetes auth path
install.externalVault.externalVaultCACertSecret	Name of the secret where **vault-ca.crt** file is mounted.
install.externalVault.externalVaultDBRole	Static role created to access the database cred.
install.externalVault.externalVaultEnginePath	Enter the value “/database”
install.externalVault.externalVaultKubeAuthPath	The Kuberenetes access path created with cluster information for service account authentication.
install.externalVault.externalVaultSAName	The Service account used to create externalVaultAuthRole.
install.externalVault.mongoPasswordVaultEngine	Enter the value DATABASE

8. Once the values are filled in *.appviewxctl* as described in the step above, proceed with the installation. Before doing so, check if the the preconditions are met by executing the command

```
./appviewxctl preflight --config .appviewxctl.yaml
```

   This will prompt if the necessary prerequisites are met.

9. To proceed with installation, execute the command

```
./appviewxctl install --config .appviewxctl.yaml
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Note: The installation will take several minutes to complete. Upon completion you see the following message:

`[Install] Successfully installed Appviewx infra stack`

This would imply the completion of infra component setup.

10. This step involves restoring the existing data from the previous AppViewX version's cluster in case there is a need to migrate from the older versions to the Managed K8s version. **Ignore this step if it's a fresh setup with no migration necessary.**

    To restore mongodb and vault fetch the backup files and place them in the bastion in a directory such as /home/user/backup execute the mongo_restore and vault_restore scripts as follows:

    Note: Before you trigger the `mongo_restore.sh` command, ensure that the mongo restore file has the file extension as **.tar.gz**.

    ```
    ./mongo_restore.sh <mongo backup tar filepath>
    ```

    ```
    ./vault_restore.sh -p <vault backup filepath>
    ```

    Attention: If the data is being restored from an older version (2020.3.0 - 2022.1) then use the command

    ```
    ./vault_restore.sh -p <vault backup filepath> --removedek
    ```

    Note:
    - The backup files must have extension as **.tar.gz**
    - The above commands work for a self-managed mongodb setup. Setting up the mongodb atlas requires the installation of mongodb tools in the bastion host as described below.

    For an rpm based OS:

    ```
    echo -e "[mongodb-org-4.2] \nname=MongoDB
     Repository\nbaseurl=https://repo.mongodb.org/yum/redhat/\$releasever/mongodb-org/4.2/x86_64/\ngpgcheck=1\nenabled=1\ngpgkey=https://
    www.mongodb.org/static/pgp/server-4.2.asc" > /etc/yum.repos.d/mongodb-org-4.2.repo
    yum install mongodb-org-shell-4.2.0
    yum install mongodb-org-tools-4.2.0
    ```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

For a debian based OS:

```
wget -qO - https://www.mongodb.org/static/pgp/server-6.0.asc | sudo apt-key add -

sudo apt-get install gnupg

wget -qO - https://www.mongodb.org/static/pgp/server-6.0.asc | sudo apt-key add -

echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0 multiverse" | sudo

 tee /etc/apt/sources.list.d/mongodb-org-6.0.list

sudo apt-get update

sudo apt-get install -y mongodb-mongosh

sudo apt-get install -y mongodb-org-tools
```

Verify if the mongodb restore commands have executed successfully using the command

```
mongorestore -- version
```

11. To proceed with the AppViewX application installation, execute the command:

```
./appviewxctl installapp --config .appviewxctl.yaml
```

Once installation is complete the following messages are displayed:

```
[Install] Appviewx infrastructure chart [avx-app] installed successfully

[Install] Successfully installed Appviewx application stack

[Install] Fetching login URL for app

[Install] Waiting for Public IP allotment for istio service

[Install] AppViewX Web URL: https://34.100.197.159/appviewx/

[Install] AppViewX Gateway URL: https://34.100.197.159/avxmgr/

[Install] Grafana URL: https://34.100.197.159/grafana/

[Install] Kibana URL: https://34.100.197.159/kibana/login

[Install] Run below commands to get mongo user credentials

 export MONGO_USER=$(kubectl get secret -n avx mongo-key -o=jsonpath='{.data.mongo-init-user}' | base64 -d)

 export MONGO_PASS=$(kubectl get secret -n avx mongo-key -o=jsonpath='{.data.mongo-init-pass}' | base64 -d)

[Install] Run below commands to get Elasticsearch and Kibana credentials

 export ES_PASS=$(kubectl get secret -n avx elasticsearch-pw-elasticsearch -o=jsonpath='{.data.password}' | base64 -d)

 export KIBANA_PASS=$(kubectl get secret -n avx elasticsearch-pw-kibana -o=jsonpath='{.data.password}' | base64 -d

[Install] Application Installation completed successfully
```

Note: Follow the URLs and commands given in the output message to get the credentials and access the application.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

12. If installation of the third party monitoring components was not enabled during the entire process, they can be installed later by the following steps:

    a. While installing the third party components (helm_charts/avx_third_party/values.yaml), the only that values are set to 'true' by default are - *prometheus, nodeexporter, kube-state metrics*. The other components are set as 'false' by default and must be to set to true if they are to be enabled, they are - *elk-elasticsearch, elk-filebeat, elk-kibana, elk-logstash, grafana, elasticsearch-insight, logstash-syslog.*

    b. Edit the **.appviewxctl.yaml** file and set **install.enableThirdPartyInstall** to ‘true’

    c. Configure the following **thirdPartyApp** parameters as true as per the requirements:
    - **install.thirdPartyApp.elk**
    - **install.thirdPartyApp.monitoring**
    - **install.thirdPartyApp.insight**

    d. Now, edit the file **values.yaml** present at location helm_charts/appviewx_monitoring/prometheus/chart/values.yaml and append the below values at the end of the file (only if that are not present).

```
limits:
  cpu_limit: 80
  memory_limit: 80
  disk_limit: 80
  timelimit_cpu_memory: 5
  timelimit_disk: 1
  timelimit_pod: 1
  timelimit_node: 1
```

    e. Run the command below

```
./appviewxctl installtpt --config .appviewxctl.yaml
```

What to do next

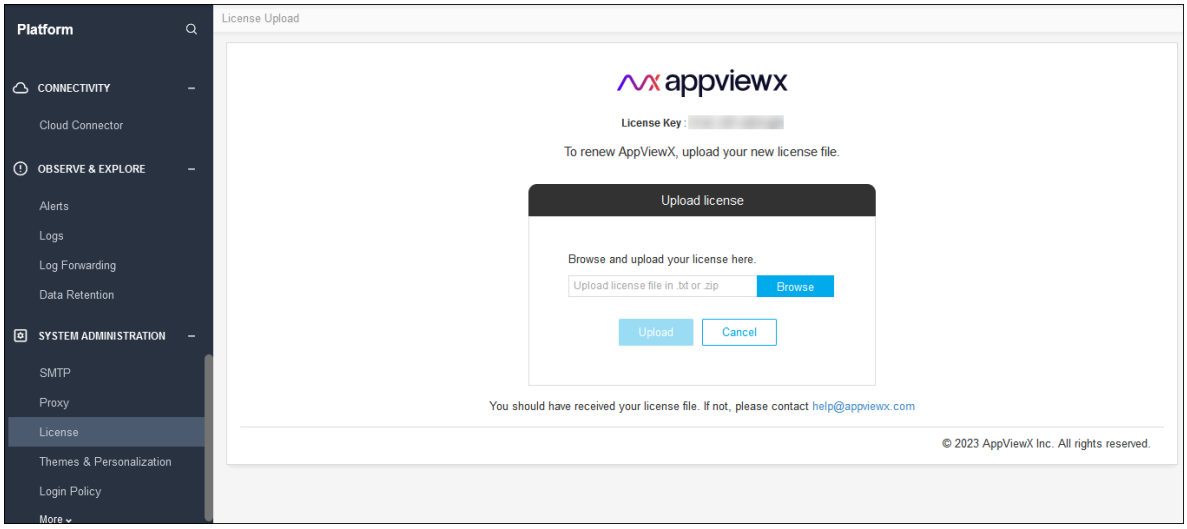
1. If you have to install the Helm based CC, then parch the internal tenant secret, using the command below:

```
kubectl patch secret internal-tenant -n avx --type='json' -p='[{"op": "add", "path": "/data/REPO_AUTH_TOKEN", "value": "<repoauthtokenvalue>"}]'
```

   Once the command is executed, restart the saas-proxy.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

2. Customers migrating from AppViewX version 2020.3.0 to Managed Kubernetes, it is mandatory to upgrade the license.
   a. Login to the AppViewX with valid credentials.
   b. Navigate to Platform >> System Administration >> License page.
   c. Click **Upgrade License**.



   d. Click **Browse** to find the latest license key file.
   e. Click **Upload**.

> **Note:** For the licenses contact AppViewX Support at help@appviewx.com or customerlicences@appviewx.com.

Upgrade AppViewX in Managed Kubernetes

Before you begin: Before performing the infra upgrade, upgrade the kubernetes to the latest version **1.31**. Refer to the update the Kubernetes version for your Amazon EKS cluster webpage.

> **Attention:**
>
> • If you are using the self managed private docker registry instead of AppViewX's docker registry, then before proceeding with the upgrade, ensure you have copied the latest images to your registry. The list of images can be found in the Prerequisite section - AppViewX Docker Images.

To upgrade AppViewX with a new image version, follow the steps below:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

1. To take the backups, execute the commands below.

   For self-managed mongodb:

   ```
   kubectl create job --from=cronjob/mongo-backup -n avx mongo-backup-<unique-identifier>
   ```

   ```
   kubectl create job --from=cronjob/vault-backup -n avx vault-backup-<unique-identifier>
   ```

   Replace <unique-identifier> in above commands with some random string and run. Monitor the pods until completion and verify the backups are placed in the storage bucket.

   > **Note:** Atlas backup must be taken in the atlas dashboard. Refer to the atlas snapshots section in the page Backup and Restore. .

2. Download the upgrade tar file (**upgrade.tar.gz**) from the release portal and extract it to a suitable location. (The extracted files contain the binary and helm charts tar.)
3. Navigate to the folder where the upgrade tar is extracted.
4. Copy the **appviewxctl** binary from the current folder (extracted folder location) to the installer location.

   ```
   cp appviewxctl <absolute path of the installer directory>
   ```

5. Navigate to the installer directory.
6. To merge the old data of .appviewxctl.yaml to the new file, execute the command below.

   ```
   ./appviewxctl mergeConfForUpgrade --config .appviewxctl.yaml
   ```

```
eksuser@bastion-hsot-qa-1:~/mar6$ ./appviewxctl mergeConfForUpgrade --config .appviewxctl.yaml
[Init] Using log file at [/home/eksuser/mar6/appviewxctl-554518342.log] to dump logs
[Init] Initialise persistent flag config
[Init] Using config file
Created backup file: .appviewxctl-old-20250310-134744.yaml
[Config] Generating config file with provider gcp - .appviewxctl.yaml
[Config] Successfully generated sample config file.
Waiting for config generation to complete...
Created backup file: .appviewxctl-tmp-bkp.yaml
Plugin 'avx-vendors-sync' has been added under install key. Do you want to enable it? (y/n): y
Field 'socketurl' has been added under root key. Please enter the new field value (default: ): web.socket.com
Plugin 'avx-vendors-sync' has been added under upgrade key. Do you want to enable it? (y/n): n

Summary of New Fields:
------------------------------------------------------------
Field Path                          | Value
------------------------------------------------------------
avx-vendors-sync                    | enable: true
socketurl                           | web.socket.com
avx-vendors-sync                    | enable: false
------------------------------------------------------------

Do you want to proceed with these changes? (y/n): y
Merged old YAML values into new YAML.

Please enter the output file name for the new YAML. Hit enter to take the default value. (default: .appviewxctl.yaml):
Successfully saved YAML to .appviewxctl.yaml
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

During the execution of the above command, the fields of the old conf file are compared with that of the newly generated conf file and you will be prompted to enter values.

- If it's a plugin, you'll be prompted to enable or disable it during both installation and upgrade.
- If a new field has been added to the `.appviewxctl.yaml` file, you'll be prompted to provide a value for it. Pressing Enter without input will apply the default value.

Once all the values are entered, the process continues as follows:

- A summary table is displayed to confirm the provided inputs.
- If the user declines, the prompts will repeat.
- If the user confirms, the system proceeds to merge the old YAML with the new one.
- The values provided by the user for the newly added fields will be retained, while existing fields will retain values from the old configuration file.
- Users can specify a custom output filename for the merged YAML; if not specified, the default will be `.appviewxctl.yaml`.
- The merged YAML is successfully saved to the output file.

7. Refer to the tables below for the essential upgrade parameters and their description.

Upgrade Parameters in the appviewxctl.yaml file

Parameters	Description of Values
upgrade.imageRegistry	The URL of the container registry where the images are to be pulled from by the pods. *Example*: gcr.io/pe-qa-358108
upgrade.imageTag	The tag of the image that will be used for upgrade. *Example*: 2023.1.0_FP_750-alpine
install.imageTag	The tag of the image that will be used for installation. *Example*: 2022.1.0_FP_750-alpine
upgrade.isSaasEnabled	Boolean value for SaaS enablement. This value should be set to **true** for Managed K8s.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Parameters	Description of Values

upgrade.plugins	The list of plugins that will be installed. Each plugin will have three fields
	• enable
	• imageTag
	• name
	
	Set enable to **true** if the plugin is to be upgraded. If the same image tag is to be used as defined in the global ImageTag keep it **latest** otherwise override with some other tag of your choice.
	
	Note: The list of plugins to be enabled should match the ones in the install section.
	
	Example:
	`- enable: true`
	`imageTag: latest`
	`name: avx-config-server`

upgrade.upgradeInfra	Boolean value to upgrade infra component. Set to True for upgrade.
upgrade.upgradeThirdParty	Boolean value to upgrade the monitoring (ELK, insight, and monitoring) components. Set to True for upgrade.
socketurl	Customers can now configure their own web-socket URL in the MongoDB by updating the `socketurl` parameter in the **.appviewxctl.yaml** file. After installation or upgrade, the specified socket URL will be recorded in the **saas_agent_meta** collection.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Component Parameters in the appviewxctl.yaml file

Parameters	Description of Values
install.thirdPartyApp.elk	Boolean value to add Elk component. Set to True for upgrade.
install.thirdPartyApp.monitoring	Boolean value to add Monitoring component. Set to True for upgrade.
install.thirdPartyApp.insight	Boolean value to add Insight component. Set to True for upgrade.

Install parameters in the appviewxctl.yaml file required to integrate the kubernetes cluster to the external vault.

Parameters	Description of Values
cloudConnectorEnabled	A boolean value (true/false) to denote the cloud connector usage for southbound communications. If a cloud connector is used set the value to **true**.
install.externalVault.enable	A boolean value (true/false) to denote if the external vault is to be used in the setup. True is to enable the external vault.
install.externalVault.externalVaultAddr	Contains the vault URL and listening port *Example*: https://pm-lxs-node01.lab.appviewx.net:8200
install.externalVault.externalVaultAuthRole	Name of the role created against the access kubernetes auth path
install.externalVault.externalVaultCACertSecret	Name of the secret where **vault-ca.crt** file is mounted.
install.externalVault.externalVaultDBRole	Static role created to access the database cred.
install.externalVault.externalVaultEnginePath	Enter the value “/database”

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Parameters	Description of Values
install.externalVault.externalVaultKubeAuthPath	The Kuberenetes access path created with cluster information for service account authentication.
install.externalVault.externalVaultSAName	The Service account used to create externalVaultAuthRole.
install.externalVault.mongoPasswordVaultEngine	Enter the value DATABASE

Note: Path parameters should have a leading forward slash '/'.

8. Copy the **vault_restore_openbao_migration.sh** file from the current folder (extracted folder location) to installer location

Note: If this step was completed during the previous upgrade (2024.3.0.0), you can skip it. However, if you are upgrading from an older version (2023.1.3 or earlier), this step is required.

```
cp vault_restore_openbao_migration.sh <absolute path of the installer directory>
```

9. Copy the **mongo_restore.sh** file from the current folder (extracted folder location) to installer location

```
cp mongo_restore.sh <absolute path of the installer directory>
```

10. To upgrade AppViewX infra, execute the command

Note: If you plan on enabling additional 3pt monitoring components as part of the infra upgrade do the following:

a. Navigate to <installer>/helm_charts/avx_thrid_party/.
b. Edit the **values.yaml** file.
c. Set "enable" to true for the components you wish to enable as part of the upgrade.

```
./appviewxctl infraUpgrade --config .appviewxctl.yaml
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.



This will prompt the following message

```
Vault Backup taken (Yes/No):
```

If you haven't taken the backup, enter **No**, and execute the vault backup command in step 2. If you have already taken the backup, enter **Yes**. You will be prompted with the message below.

```
Please provide the path of updated helm charts tar. :
```

Enter the absolute path of the extracted Helm charts tar artifact to proceed with the infrastructure upgrade. Ensure you provide the full file path, not just the directory location.

> **Note:** After completing the infrastructure upgrade and before triggering the **installMongoRS**, ensure the `install.imageTag` in the `.appviewxctl.yml` file is updated with the latest imageTag.

11. Install **Mongo Replicaset** using the command below

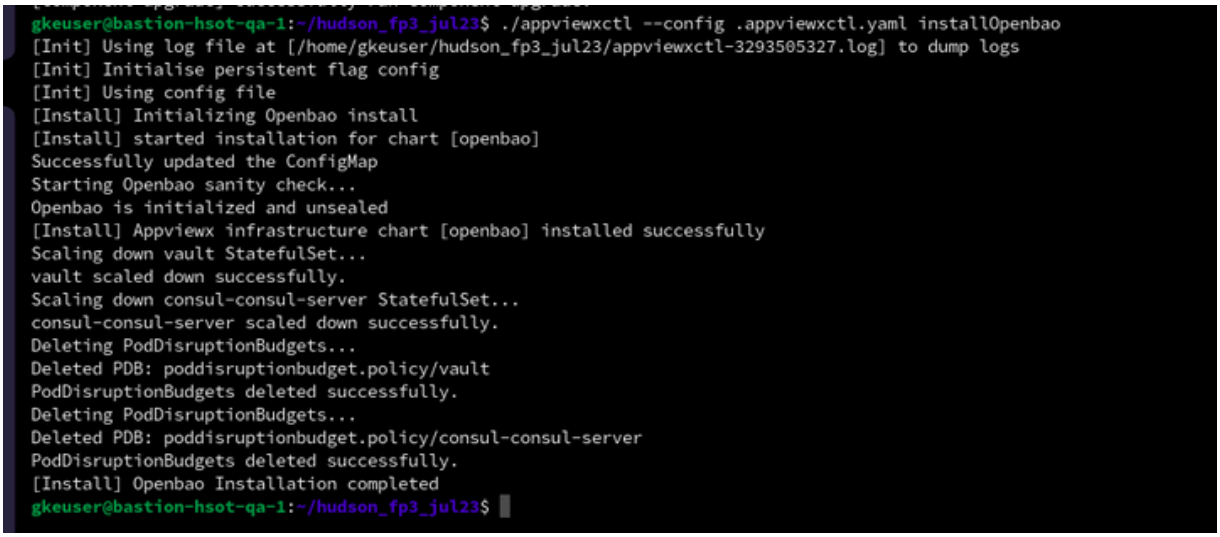
> **Note:** If this step was completed during the previous upgrade (2024.3.0.0), you can skip it. However, if you are upgrading from an older version (2023.1.3 or earlier), this step is required.

```
./appviewxctl --config .appviewxctl.yaml installMongoRS
```

12. Install **OpenBao** using the command below.

```
./appviewxctl --config .appviewxctl.yaml installOpenbao
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.



Note: The plugin pods must not be restarted after installing OpenBao until the plugin upgrade is complete. Restarting the pods prematurely may disrupt functionality, as Vault could lock the user if it detects multiple login attempts.

13. To restore the vault, fetch the backup files and place them in the bastion in a directory such as /home/user/backup execute the restore scripts below.

Note: If this step was completed during the previous upgrade (2024.3.0.0), you can skip it. However, if you are upgrading from an older version (2023.1.3 or earlier), this step is required.

```
./vault_restore_openbao_migration.sh -p <absolute path of the vault_bkp_file_location>
```

14. To restore the mongo, fetch the backup files and place them in the bastion in a directory such as /home/user/backup and execute the restore scripts below.

Note:

a. If this step was completed during the previous upgrade (2024.3.0.0), you can skip it. However, if you are upgrading from an older version (2023.1.3 or earlier), this step is required.

b. Before you trigger the `mongo_restore.sh` command, ensure that the mongo restore file has the file extension as **.tar.gz**.

```
./mongo_restore.sh <absolute path of the mongo_bkp_file_location>
```

15. Run the plugin upgrade command below.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
./appviewxctl upgrade --config .appviewxctl.yaml
```

Note: It is mandatory to carry out the Infra upgrade before the plugin upgrade.

Rollback Steps

a. Restore the DB using the restore scripts (step 11 in the Installation Steps section) for self-managed DB or in atlas using snapshot restore in the dashboard.
b. Update the **appviewxctl.yaml** upgrade section's values to the previous image tag and re-run the upgrade command.

- Troubleshooting for Upgrade Errors
- MongoShardDB Cleanup
- Vault/Consul Cleanup
- OpenBAO Cleanup
- Mongo Replicaset Cleanup

Post-Upgrade Actions

The following steps have to be performed after you completed the upgrade.

1. Retrieve the updated **web login IP** by executing:

   ```
   kubectl get svc -n istio-system
   ```

   From the output, locate the `ExternalIP` associated with the `istio-ingressgateway` and use that IP to access the web login page.

2. Update the **RSocket URL** in the properties file and restart the `avx-mid-server-starter` and `avx-mid-server-platform` pods.

   a. In the UI, go to **Platform** > **Cloud Connector**.

   b. Copy the **Cloud Connector URL**.

   c. On the Cloud Connector (CC) node CLI, go to the properties directory:

   ```
   cd <Installer_Folder>/deps/properties
   ```

   d. Open the `appviewx.properties` file using a text editor:

   ```
   vi appviewx.properties
   ```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

e. Update the value of `RSOCKET_URL` with the copied Cloud Connector URL (e.g., `https://98.70.232.0:443`) and save the file.

f. Restart the following pods to apply the changes:
- avx-mid-server-starter
- avx-mid-server-platform

```
kubectl delete pods -n cc avx-mid-server-starter,avx-mid-server-platform --force
```

3. If GUI login fails:

a. Restart all pods:

```
kubectl delete pods -A --all --force
```

b. Allow the pods to restart and verify that they are running.

c. When all pods are running, log in again.

What to do next:

Complete the following steps after the pluging upgrade ONLY if they have not been performed previously.

- Remove/delete all traces MongoShardDB setup, refer to the section MongoShardDB Cleanup.
- Remove/delete all traces of vault setup, refer to the section Vault/Consul Cleanup.

Troubleshooting for Upgrade Errors

UPGRADE FAILED: another operation (install/upgrade/rollback) is in progress

While performing the upgrade (`./appviewxctl upgrade --config .appviewxctl.yaml`), if the system encounters the error of `UPGRADE FAILED: another operation (install/upgrade/rollback) is in progress` as shown below.

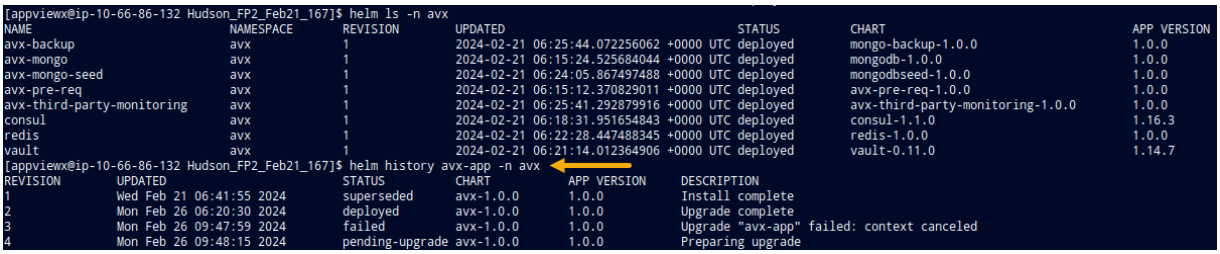
```
[appviewx@ip-10-66-86-132 Hudson_FP2_Feb21_167]$ ./appviewxctl upgrade --config .appviewxctl.yaml
[Init] Using log file at [/home/appviewx/Hudson_FP2_Feb21_167/appviewxctl-4556929.log] to dump logs
[Init] Initialise persistent flag config
[Init] Using config file
[Upgrade] Upgrade of the AVX application started...
[Upgrade] Updating enabled plugins dependency and configurations
[Upgrade] updated dependency [avx_app]
[Upgrade] Starting Appviewx application chart [avx-app] upgrade
[Upgrade] failed to run upgrade with error: exit status 1: Error: UPGRADE FAILED: another operation
(install/upgrade/rollback) is in progress
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Perform the following steps:

1. Fetch the deployment history of **avx-app helm** deployment using command mentioned below.

```
helm history avx-app -n avx
```



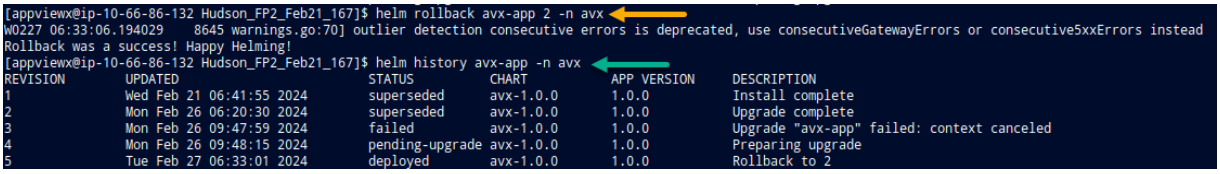
2. Find the most recent revision of avx-app which is in deployed state.

   In this case Revision 2 is the most recent upgrade.
3. Run the rollback command to retrieve the deployment from failed state.

```
helm rollback avx-app <REVISION NUMBER> -n avx
```

Example:

```
helm rollback avx-app 2 -n avx
```



4. Re-run the `helm history avx-app -n avx` command to verify the rollback is complete.

Failure Issue during Upgrade

1. Go to the section Cleanup of OpenBao and follow the steps provided.
2. When the cleanup is complete, install in the following order:

```
#Install OpenBao

./appviewxctl --config .appviewxctl.yaml installOpenbao


# Restore Vault

./vault_restore_openbao_migration.sh -p <vault backup file full path>


# Restore Mongo
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
./mongo_restore.sh <mongo backup file full path>


# Upgrade Plugins
./appviewxctl upgrade --config .appviewxctl.yaml
```

Login Issue after Upgrade

If a login issue occurs after the upgrade:

1. Perform he vault cleanup, refer the section Vault/Consul Cleanup.
2. Reinstall OpenBao.
3. Restore the Vault data.
4. Now, repeat the application upgrade process.

MongoShardDB Cleanup

After completing the plugins upgrade and validating the functionality, it is necessary to clean up (delete) all traces of the MongoDB shard database setup. Follow the steps outlined below:

1. Remove the Mongo ShardDB using the command below.

```
helm uninstall avx-mongo -n avx
```

2. Delete the StatefulSet resource by executing the command below.

```
kubectl delete sts mongo-shareddb mongo-configdb mongo-routerdb -n avx --force
```

3. Delete the specified Kubernetes services (service objects) in the cluster by executing the command below.

```
kubectl delete svc mongo-shareddb-service mongo-configdb-service mongo-routerdb-service -n avx --force
```

4. Delete the MongoShardDB related **PVCs** by executing the commands below.

```
kubectl delete pvc mongo-configdb-persistent-storage-claim-mongo-configdb-0 mongo-configdb-persistent-storage-claim-mongo-configdb-1
 mongo-configdb-persistent-storage-claim-mongo-configdb-2 -n avx --force
```

```
kubectl delete pvc mongo-shareddb-persistent-storage-claim-mongo-shareddb-0 mongo-shareddb-persistent-storage-claim-mongo-shareddb-1
 mongo-shareddb-persistent-storage-claim-mongo-shareddb-2 -n avx --force
```

5. Delete the MongoShardDB related kubernetes secrets using the commands below.

```
kubectl delete secret mongo-key -n avx --force
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Vault/Consul Cleanup

After completing the plugins upgrade or if you have recently migrated/installed OpenBao, and validating the functionality, it is necessary to clean up (delete) all traces of the Vault/Consul setup. Follow the steps outlined below:

> ⚠ **Warning:** Do not perform these steps if the migration from Vault to OpenBao has not been completed

1. Remove the vault/consul and related pods using the commands below.

```
helm uninstall vault -n avx
```

```
helm uninstall consul -n avx
```

```
kubectl delete pods vault-0 vault-1 vault-2 -n avx --force
```

```
kubectl delete pods consul-consul-server-0 consul-consul-server-1 consul-consul-server-2 -n avx --force
```

2. Delete the specified Kubernetes services (service objects) in the cluster by executing the command below.

```
kubectl delete secret vault-key-0 vault-key-1 vault-key-2 -n avx --force
```

3. Delete the vault/consul related **PVCs** by executing the commands below.

```
kubectl patch pvc data-avx-consul-consul-server-0 -n avx -p '{"metadata":{"finalizers":null}}'
kubectl patch pvc data-avx-consul-consul-server-1 -n avx -p '{"metadata":{"finalizers":null}}'
kubectl patch pvc data-avx-consul-consul-server-2 -n avx -p '{"metadata":{"finalizers":null}}'
kubectl delete pvc data-avx-consul-consul-server-0 data-avx-consul-consul-server-1 data-avx-consul-consul-server-2 -n avx --force
```

4. Delete the configmap using the commands below.

```
kubectl delete cm avx-vault-configmap -n avx
```

OpenBAO Cleanup

If any issues are encountered while running the `installOpenbao` command, ensure that all traces of previous attempts are cleaned up before retrying by following these steps:

1. Remove openbao release and pods using the commands below.

```
helm uninstall openbao -n avx
```

```
kubectl delete pods openbao-0 openbao-1 openbao-2 -n avx --force
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

2. Delete kubernetes secrets using the commands below.

```
kubectl delete secret openbao-key-0 openbao-key-1 openbao-key-2 -n avx --force
```

```
kubectl delete secret bao-key-secretid -n avx --force
```

```
kubectl delete secret bao-key-secretid -n avx-kafka --force
```

```
kubectl delete secret bao-key-secretid --force
```

3. Delete **pvcs** by executing the command:

```
kubectl delete pvc data-openbao-0 data-openbao-1 data-openbao-2 -n avx --force
```

4. Delete **configmap** by executing the command:

```
kubectl delete cm avx-openbao-configmap -n avx --force
```

Mongo Replicaset Cleanup

If any issues are encountered while running the `installMongoRS` command, ensure that all traces of previous attempts are cleaned up before retrying by following these steps:

1. Remove the Mongo Replicaset using the command below.

```
helm uninstall avx-mongo-rs -n avx
```

2. Delete the StatefulSet resource by executing the command below.

```
kubectl delete sts avx-mongo-mongodb -n avx --force
```

3. Delete the specified Kubernetes services (service objects) in the cluster by executing the command below.

```
kubectl delete svc avx-mongo-mongodb-headless -n avx --force
```

4. Delete the Mongo Replicaset related **PVCs** by executing the commands below.

```
kubectl delete pvc avx-mongo-mongodb-persistent-storage-claim-avx-mongo-mongodb-0
 avx-mongo-mongodb-persistent-storage-claim-avx-mongo-mongodb-1 avx-mongo-mongodb-persistent-storage-claim-avx-mongo-mongodb-2 -n avx --force
```

5. Delete the Mongo Replicaset related kubernetes secrets using the commands below.

```
kubectl delete secret mongo-key-rs -n avx --force
```

Downloading Images from AppViewX Repository

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Prerequisites

1. Get the source image repository credentials from AppViewX.
2. Configure the docker using the command

```
docker login -u ${USERNAME} -p ${PASSWORD} ${DOCKER_REPOSITORY}
```

3. Configure the respective cloud provider CLI (AWS) and ensure you have access to push docker images to ECR.

The script for image push and pull is as follows:

```
#!/bin/bash
# Pre requisite
# Get the source image repository credentials from Appviewx and then configure docker using `docker login -u ${USERNAME} -p ${PASSWORD}
 ${DOCKER_REPOSITORY}`
# Configure the respective cloud provider cli (aws/az/gcloud) and make sure that the user has access to push docker images to ACR/ECR/GCR.


appVersion=$1 # App image version. E.g: 24.3.0.0
targetImageRegistry=$2 # Image resgistry name


# Validate required inputs
if [ -z "$appVersion" ] || [ -z "$targetImageRegistry" ];then
{
    echo "Please provide script parametes as ./script.sh <appVersion> <targetImageRegistry>"
    exit
}
fi


# Set the registry login
if echo $targetImageRegistry | grep -iq "amazonaws";then
{
    registryProvider="ecr"
    region=$(echo $targetImageRegistry | cut -d "." -f4)
    aws ecr get-login-password --region $region | docker login --username AWS --password-stdin $targetImageRegistry
}
elif echo $targetImageRegistry | grep -iq "azurecr";then
{
    registryProvider="acr"
    az acr login -n $targetImageRegistry
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
}
elif echo $targetImageRegistry | grep -iq "gcr";then
{
    registryProvider="gcr"
    gcloud auth print-access-token | docker login -u oauth2accesstoken \
    --password-stdin $(echo $targetImageRegistry | cut -d '/' -f2)
}
else
{
    echo "Unknown registry provider"
    exit 2
}
fi

# Image tag mappings
imageTags='[
   {
      "imageName": "avx-cloud-managedservice-mks",
      "tagVersion": "appVersion",
      "upload": true
   },
   {
      "imageName": "avx-cloud-web",
      "tagVersion": "appVersion",
      "upload": true
   },
   {
      "imageName": "avx-cloud-gateway",
      "tagVersion": "appVersion",
      "upload": true
   },
   {
      "imageName": "avx-platform-report-generator",
      "tagVersion": "appVersion",
      "upload": true
   },
   {
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
    "imageName": "mongo-init",
    "tagVersion": "appVersion",
    "upload": true
},
{
    "imageName": "avx-cloud-mongoseed",
    "tagVersion": "appVersion",
    "upload": true
},
{
    "imageName": "avx-python-sandbox",
    "tagVersion": "appVersion",
    "upload": true
},
{
    "imageName": "avx-mid-server-base",
    "tagVersion": "appVersion",
    "upload": true
},
{
    "imageName": "alpine",
    "tagVersion": "3.22.0",
    "upload": true
},

{
    "imageName": "pilot",
    "tagVersion": "1.26.1",
    "upload": true
},
{
    "imageName": "proxyv2",
    "tagVersion": "1.26.1",
    "upload": true
},
{
    "imageName": "openbao",
```


Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
    "tagVersion": "2.2.2",
    "upload": true
},
{
    "imageName": "redis",
    "tagVersion": "8.0.2",
    "upload": true
},
{
    "imageName": "kafka",
    "tagVersion": "0.45.0-kafka-3.9.0",
    "upload": true
},
{
    "imageName": "operator",
    "tagVersion": "0.45.0",
    "upload": true
},
{
    "imageName": "kube-metrics-adapter",
    "tagVersion": "v0.2.3",
    "upload": true
},
{
    "imageName": "kibana",
    "tagVersion": "9.0.2",
    "upload": true
},
{
    "imageName": "grafana",
    "tagVersion": "11.6.2",
    "upload": true
},
{
    "imageName": "filebeat",
    "tagVersion": "9.0.2",
    "upload": true
```


Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
},
{
    "imageName": "logstash",
    "tagVersion": "9.0.2",
    "upload": true
},
{
    "imageName": "logstash-syslog",
    "tagVersion": "9.0.2",
    "upload": true
},
{
    "imageName": "elasticsearch",
    "tagVersion": "9.0.2",
    "upload": true
},
{
    "imageName": "elasticsearch-insight",
    "tagVersion": "9.0.2",
    "upload": true
},
{
    "imageName": "prometheus",
    "tagVersion": "v3.5.0",
    "upload": true
},
{
 "imageName" : "backup-utility-image",
 "tagVersion": "v2024.3.0",
    "upload": true
},
{
 "imageName" : "alertmanager",
 "tagVersion": "v0.28.1",
    "upload": true
},
{
```


Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
    "imageName" : "node-exporter",
    "tagVersion": "v1.9.1",
      "upload": true
  },
  {
    "imageName" : "metrics-server",
    "tagVersion": "v0.7.2",
      "upload": true
  },
  {
    "imageName" : "kube-state-metrics",
    "tagVersion": "v2.15.0",
      "upload": true
  },
  {
    "imageName" : "redis_exporter",
    "tagVersion": "v1.73.0",
      "upload": true
  }
]'

for row in $(echo "${imageTags}" | jq -r '.[] | @base64'); do
  _jq() {
    echo ${row} | base64 --decode | jq -r ${1}
  }
  imageUpload=$(_jq '.upload')
  tagVersion=$(_jq '.tagVersion')
  if [ $imageUpload == "true" ];then
  {
      if [ "${tagVersion}" == "appVersion" ];then
      {
          docker pull images.appviewx.com/appviewx/$(_jq '.imageName'):$appVersion
          docker tag images.appviewx.com/appviewx/$(_jq '.imageName'):$appVersion $targetImageRegistry/appviewx/$(_jq '.imageName'):$appVersion
          docker push $targetImageRegistry/appviewx/$(_jq '.imageName'):$appVersion
      }
      else
      {
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
        docker pull images.appviewx.com/appviewx/$(_jq '.imageName'):$(_jq '.tagVersion')
        docker tag images.appviewx.com/appviewx/$(_jq '.imageName'):$(_jq '.tagVersion') $targetImageRegistry/appviewx/$(_jq '.imageName'):$(_jq
'.tagVersion')
        docker push $targetImageRegistry/appviewx/$(_jq '.imageName'):$(_jq '.tagVersion')
      }
      fi
   }
   fi
done
```

Execute the Image Push-Pull Script

To execute the above image push-pull script, run the command

```
./avx_image_pull_push.sh <image-tag> <targetImageRegistry>
```

Note: Contact AppViewX Support for the <**image-tag**>.

Utility Tools and Functions

Updating Ingress Host in the Config Map

Customers can now update their own ingress host directly in the config map using the updateIngressHosts utility below. If they are using an external load balancer and wish to modify the ingress, it can be done using the following command:

```
./appviewxctl updateIngressHosts --config .appviewxctl.yaml
```

On executing the above utility function, you will be prompted with the following:

- *Enter the loadbalancer URL (Ingress Host):*
- *Enter the ingress port (default 443):* - If you're using the default port 443, simply press Enter. Otherwise, enter the desired port number and then press Enter.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
eksuser@bastion-hsot-qa-1:~/mar6$ ./appviewxctl updateIngressHosts --config .appviewxctl.yaml
[Init] Using log file at [/home/eksuser/mar6/appviewxctl-2220583372.log] to dump logs
[Init] Initialise persistent flag config
[Init] Using config file
INFO[2025-03-10T11:55:47Z] Starting ingress host update process.
Enter the loadbalancer URL (Ingress host): hello.world.com
INFO[2025-03-10T11:55:59Z] Ingress host provided: hello.world.com
Enter the ingress port (default 443):
INFO[2025-03-10T11:56:02Z] No port provided. Using default port: 443
INFO[2025-03-10T11:56:02Z] Ingress port provided: 443
INFO[2025-03-10T11:56:02Z] Value has been updated in configmap          function="Patch Configmap" name=avx-common-config namespace=avx
[Install/Upgrade] key value pair has been updated in configmap avx-common-config
INFO[2025-03-10T11:56:02Z] Value has been updated in configmap          function="Patch Configmap" name=avx-common-config namespace=avx
[Install/Upgrade] key value pair has been updated in configmap avx-common-config
INFO[2025-03-10T11:56:02Z] ConfigMap updated successfully.
INFO[2025-03-10T11:56:02Z] Restarting updated plugin pods: [avx-config-server avx-commons avx-crontab avx-platform-amc avx-platform-core avx-platform-gateway avx
-platform-aep-gateway avx-platform-logforwarding avx-platform-queue avx-platform-report-generator avx-platform-web avx-subsystems avx-subsystems-sync avx-vendor-
cert-acme-agent avx-vendor-cert-est-agent avx-vendor-cert-cmp-agent avx-vendor-cert-intune-agent avx-vendor-cert-network-discovery avx-vendor-cert-scep-agent avx
-vendors avx-visual-page-builder avx-platform-hsm avx-saas-proxy avx-saas-proxy-web-socket avx-subsystem-codesigning avx-python-sandbox-sync avx-python-sandbox]
INFO[2025-03-10T11:56:08Z] Deleted plugin pods successfully: [avx-config-server-6b66b85cfc-f942s avx-config-server-6b66b85cfc-gzmfh avx-config-server-6b66b85cfc-
nr2jr avx-crontab-6fd765d945-cgrx4 avx-platform-aep-gateway-7d898c647c-7zznr avx-platform-aep-gateway-7d898c647c-jp2h4 avx-platform-aep-gateway-7d898c647c-m6rdz
avx-platform-core-75784b8cc7-25r6n avx-platform-core-75784b8cc7-jz9w6 avx-platform-core-75784b8cc7-rhbqj avx-platform-gateway-789dfd4cb6-ck2b2 avx-platform-gatew
ay-789dfd4cb6-ckjkp avx-platform-gateway-789dfd4cb6-hwcpk avx-platform-hsm-59f5b87648-5b4pr avx-platform-hsm-59f5b87648-7cgdj avx-platform-hsm-59f5b87648-slb6g a
vx-platform-logforwarding-bc976845c-8zbhj avx-platform-logforwarding-bc976845c-jt7vp avx-platform-logforwarding-bc976845c-kjwhg avx-platform-queue-959dd9978-9z2q
s avx-platform-queue-959dd9978-c2fz8 avx-platform-queue-959dd9978-wwjc2 avx-platform-report-generator-59cf7d57d5-rgh6t avx-platform-web-57ddfd7f54-5bpjr avx-plat
form-web-57ddfd7f54-wspxx avx-platform-web-57ddfd7f54-x64zk avx-python-sandbox-9758f4c7c-4lstv avx-python-sandbox-9758f4c7c-68drh avx-python-sandbox-9758f4c7c-7l
jcz avx-python-sandbox-9758f4c7c-999fl avx-python-sandbox-9758f4c7c-cltg8 avx-python-sandbox-9758f4c7c-clw8h avx-python-sandbox-9758f4c7c-f6xnt avx-python-sandbo
x-9758f4c7c-j48vx avx-python-sandbox-9758f4c7c-l26zb avx-python-sandbox-9758f4c7c-p6srx avx-python-sandbox-9758f4c7c-qpk42 avx-python-sandbox-9758f4c7c-v6p58 avx
-python-sandbox-sync-798f45fbff-6wwkk avx-python-sandbox-sync-798f45fbff-78xlx avx-python-sandbox-sync-798f45fbff-7bl2b avx-python-sandbox-sync-798f45fbff-hmqsc
avx-python-sandbox-sync-798f45fbff-jr5sg avx-python-sandbox-sync-798f45fbff-k5r5m avx-python-sandbox-sync-798f45fbff-krp6s avx-python-sandbox-sync-798f45fbff-nhq
jk avx-python-sandbox-sync-798f45fbff-sbvsc avx-python-sandbox-sync-798f45fbff-t7vqj avx-python-sandbox-sync-798f45fbff-x55cw avx-python-sandbox-sync-798f45fbff-
zxhbl avx-saas-proxy-79874d8df5-6qt28 avx-saas-proxy-79874d8df5-fbdds avx-saas-proxy-79874d8df5-tvg2q avx-saas-proxy-web-socket-5f57cf774b-mqgmf avx-saas-proxy-w
eb-socket-5f57cf774b-nmp7x avx-saas-proxy-web-socket-5f57cf774b-pjktb avx-subsystem-codesigning-67c6cc76bf-br84d avx-subsystem-codesigning-67c6cc76bf-k5nt2 avx-s
ubsystem-codesigning-67c6cc76bf-mz2b6 avx-subsystems-57985d5dcb-7hc4n avx-subsystems-57985d5dcb-865sp avx-subsystems-57985d5dcb-tj52b avx-subsystems-sync-6c6846d
c79-952x8 avx-subsystems-sync-6c6846dc79-gl6m6 avx-subsystems-sync-6c6846dc79-n6l8v avx-subsystems-sync-6c6846dc79-nbs2v avx-subsystems-sync-6c6846dc79-qfmpv avx
-subsystems-sync-6c6846dc79-swz7b avx-vendor-cert-acme-agent-5845745995-7p6mg avx-vendor-cert-acme-agent-5845745995-8hgfq avx-vendor-cert-acme-agent-5845745995-b
6twh avx-vendor-cert-cmp-agent-6cd576878-2s79d avx-vendor-cert-cmp-agent-6cd576878-qdf8n avx-vendor-cert-cmp-agent-6cd576878-sdxdr avx-vendor-cert-est-agent-7bdc
```

Note: After config map is updated, all the plug-in pods will get restarted automatically.

Updating Istio Certificates in the tls-credential secret

Customers can now upload/update their own istio certificates using the updateIngressCerts utility provided below:

```
./appviewxctl updateIngressCerts --config .appviewxctl.yaml
```

On executing the above utility function, you will be prompted with the following:

- *Enter the full path to the Istio certificate file:* - Enter the directory path including the certificate file.
- *Enter the full path to the Istio key file:* - Enter the directory path including the certificate key.

- If the entered certificate is due to expire within one year then you will prompted with *Do you wish to continue with this certificate? (y/n):*

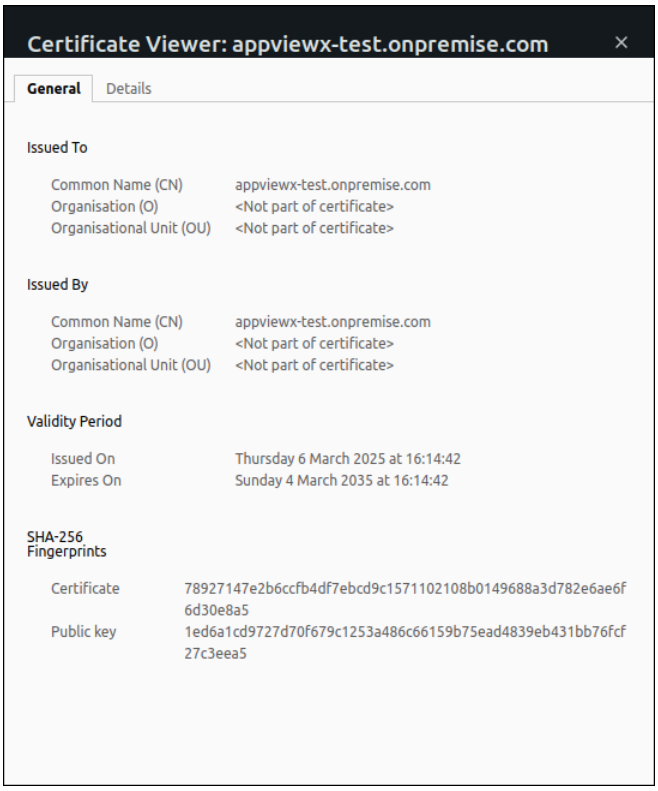
Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
eksuser@bastion-hsot-qa-1:~/mar6$ ./appviewxctl updateIstioCerts --config .appviewxctl.yaml
[Init] Using log file at [/home/eksuser/mar6/appviewxctl-108930767.log] to dump logs
[Init] Initialise persistent flag config
[Init] Using config file
Enter the full path to the Istio certificate file: /home/eksuser/mar6/test-certs/appviewx-test.crt
Enter the full path to the Istio key file: /home/eksuser/mar6/test-certs/appviewx-test.key
INFO[2025-03-10T09:54:10Z] Command[openssl] => (FAILED) Certificate will expire  shell="Shell execution scope"
WARN[2025-03-10T09:54:10Z] The certificate will expire within one year.
Do you wish to continue with this certificate? (y/n): y
INFO[2025-03-10T09:54:25Z] Successfully copied file from /home/eksuser/mar6/test-certs/appviewx-test.crt to /home/eksuser/mar6/helm_charts/appviewx_istio/istio_n
amespaces/chart/istio_cert/selfsigned.com.crt
INFO[2025-03-10T09:54:25Z] Successfully copied file from /home/eksuser/mar6/test-certs/appviewx-test.key to /home/eksuser/mar6/helm_charts/appviewx_istio/istio_n
amespaces/chart/istio_cert/selfsigned.com.key
INFO[2025-03-10T09:54:25Z] Deleting existing TLS secret...
INFO[2025-03-10T09:54:25Z] Command[kubectl] => secret "tls-credential" deleted  shell="Shell execution scope"
INFO[2025-03-10T09:54:25Z] Creating new TLS secret...
INFO[2025-03-10T09:54:26Z] Command[kubectl] => secret/tls-credential created  shell="Shell execution scope"
INFO[2025-03-10T09:54:26Z] TLS credentials have been updated successfully.
INFO[2025-03-10T09:54:26Z] Restarting Istio ingress gateway pods...
INFO[2025-03-10T09:54:57Z] Command succeeded: pod "istio-ingressgateway-74c658687-pjtfw" deleted
pod "istio-ingressgateway-74c658687-tc5c8" deleted
pod "istio-ingressgateway-74c658687-zw87r" deleted
pod "istio-ingressgateway-proxy-5554d7c69d-76frh" deleted
pod "istio-ingressgateway-proxy-5554d7c69d-nbphz" deleted
pod "istio-ingressgateway-proxy-5554d7c69d-r625t" deleted  shell="Shell execution scope"
INFO[2025-03-10T09:54:57Z] Please wait for Istio ingress gateway pods to be recreated...
```

The entered certificate file and key are placed in the **istio_certs** folder. The existing tls-credential secret is deleted and a new one is created using the entered certificate and key. The exiting istio ingress gateway and the istio ingress gateway pods are deleted. The istio ingress gateway pods are then recreated and once they are up and running the utility program ends.

The certificates can be verified on the browser as shown below.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

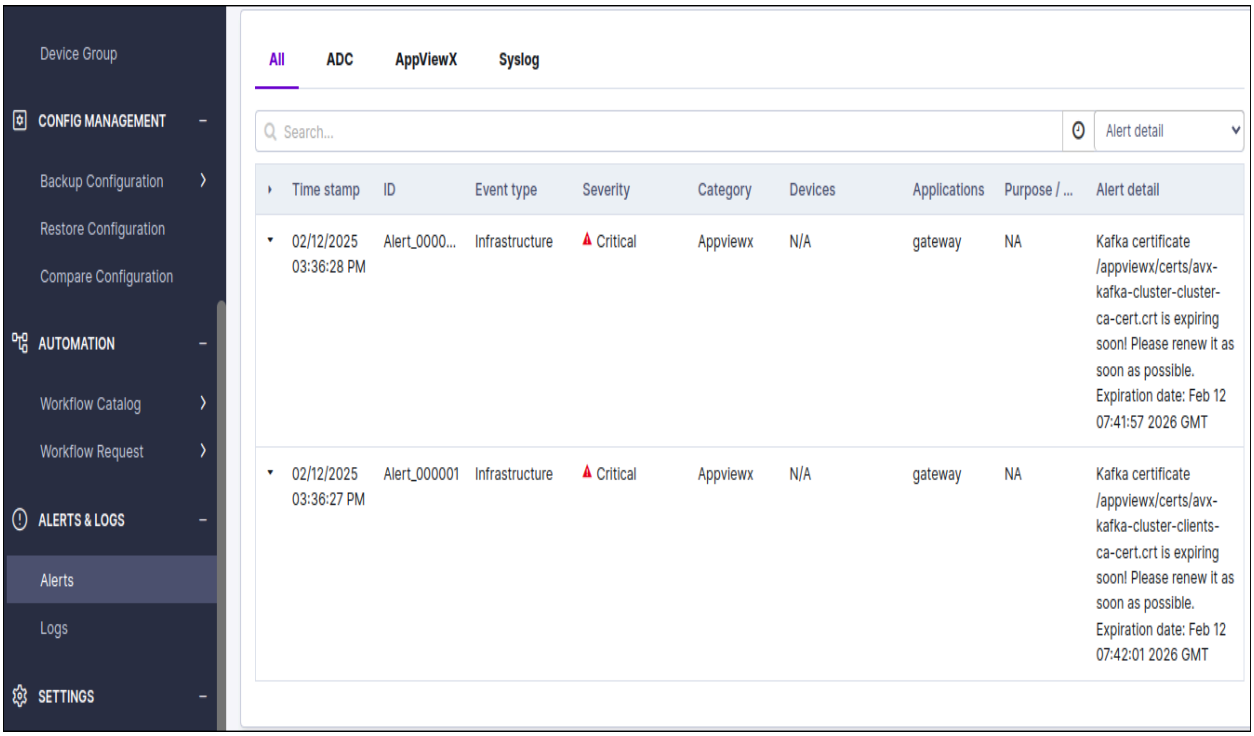


Kafka Cert Utility

If the Kafka certificates are nearing expiry, a weekly cron job is available that sends alerts to the gateway if they expire within the next 3 months. The cron job alert details can be viewed in the product by accessing the product menu, **ALERTS & LOGS** > **Alerts**.

Note: To make the alerts section visible on the gateway, first enable the following checkbox: **Identity → Role → Select appropriate role → Authorized Functions → Platform → Observe and Explore - Alerts (Checkbox).**

Copyright © 2025 AppViewX Inc. All Rights Reserved.



Utility function to renew kafka certs: Users will be able to renew kafka certs with a new utility, by executing the single command below.

```
./appviewxctl renewKafkaCerts --config .appviewxctl.yaml
```

Log Collection

The log collection script is designed to collect logs in a managed kubernetes setup. It is compatible with all mk8s environments, regardless of the cloud provider, and also retrieves logs from the backup folder.

Prerequisites: The bastion host should have kubectl and have access to the compute cluster to collect the logs.

1. Login to bastion host.
2. Download the helper tarball from the release portal.
3. Execute the following command:

```
./mk8s_collect_logs.sh
```

> **Note:** The script can be run anywhere in the bastion host.

4. Fill the interactive questions as follows:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

a. ***Do you want to collect logs for all plugins?***

- Answer **no** if you want to collect selective plugin logs. You will then be prompted to enter the names of the desired plugins.
- Answer **yes** to collect logs for all enabled plugins.

> **Note:** To view all the enabled plugins in the setup, run:
>
> ```
> ./mk8s_collect_logs.sh -operation get-enabled-plugins
> ```

b. ***Do you want to collect mongo logs?***

- Answer **yes** to include MongoDB logs in the collection.
- Answer **no** to exclude MongoDB logs.

c. ***Enter the directory to save logs:***

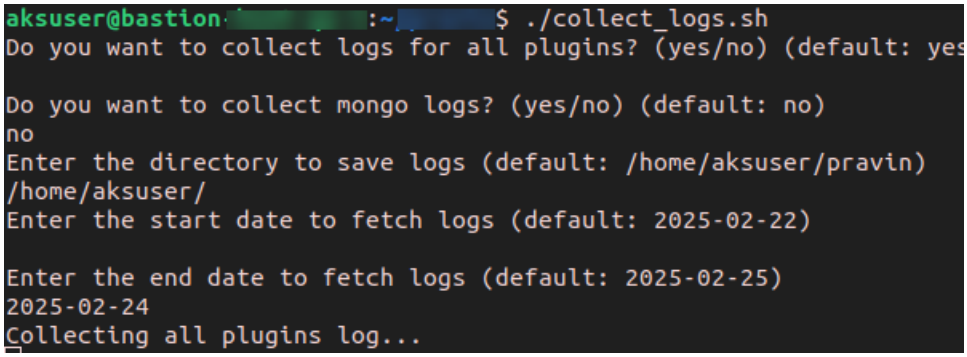
- The collected logs will be saved in the provided directory. If no directory is specified, the default path will be the directory where the script is executed.

d. ***Enter the start date to fetch logs:***

- Enter the date from which you'd like to begin collecting logs.

e. ***Enter the end date to fetch logs:***

- Enter the date until which you want to collect logs.



Alternative options: Passing Parameters via Command Line

Alternatively, you can provide the inputs as parameters instead of entering them interactively. To view the available parameters, run the command below.

```
./mk8s_collect_logs.sh --help
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Uninstall and Cleanup

The process of uninstalling requires one to navigate to the installer directory and execute the following command

```
./appviewxctl uninstall --config .appviewxctl.yaml
```

The following messages are displayed after the uninstall command is executed successfully.

```
./appviewxctl uninstall --config .appviewxctl.yaml

[Init] Using log file at [/avx/appviewxctl-3196327299.log] to dump logs
[Init] Initialise persistent flag config
[Init] Using config file
[Uninstall] Uninstalling appviewx application
[Uninstall] Uninstalling Appviewx application helm chart
[Uninstall] Uninstalling application backup helm chart
[Uninstall] Uninstalling Infra application helm chart
[Uninstall] Uninstalling Third party application helm chart
[Uninstall] Uninstalling IstioOperator from the cluster
[Uninstall] Uninstalling PVCs from the avx namespace
[Uninstall] Uninstalling Pre-requisite helm chart
[Uninstall] Uninstalling Appviewx installed namespaces
[Uninstall] Successfully uninstalled appviewx application and all the related resources
```



Note: In the Managed K8s environments removal of PVCs do not occur at times as it may require patching PVCs first before deletion. This may cause certain error messages to display, indicating that PVC has changed. In case such an error occurs, re-run the above command to solve the issue and uninstall the application.

Sometimes the namespaces take a longer time to be removed. Hence, post installation, check if namespaces are in the terminating state (use the command: **kubectl get namespace**). If any namespace is in the terminating state, manually remove the namespaces by executing the commands below:

```
kubectl get namespace "istio-operator" -o json | tr -d "\n" | sed "s/\"finalizers\": \[[^]]\+\]/\"finalizers\": []/" | kubectl replace
 --raw /api/v1/namespaces/istio-operator/finalize -f - 2>/dev/null
```

```
kubectl get namespace "istio-system" -o json | tr -d "\n" | sed "s/\"finalizers\": \[[^]]\+\]/\"finalizers\": []/" | kubectl replace
 --raw /api/v1/namespaces/istio-system/finalize -f - 2>/dev/null
```

```
kubectl get namespace "avx" -o json | tr -d "\n" | sed "s/\"finalizers\": \[[^]]\+\]/\"finalizers\": []/" | kubectl replace --raw /api/v1/namespaces/avx/finalize -f -
 2>/dev/null
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
kubectl get namespace "avx-kafka" -o json | tr -d "\n" | sed "s/\"finalizers\": \[[^]]\+\]/\"finalizers\": []/" | kubectl replace --raw /api/v1/namespaces/avx-kafka/finalize -f - 2>/dev/null
```

```
kubectl get svc "istio-ingressgateway-proxy" -n istio-system -o json | tr -d "\n" | sed "s/\"finalizers\": \[[^]]\+\]/\"finalizers\": []/" | kubectl $kubeconfig_string replace --raw /api/v1/namespaces/istio-system/services/istio-ingressgateway-proxy -f - 2>/dev/null
```

```
kubectl get svc "istio-ingressgateway" -n istio-system -o json | tr -d "\n" | sed "s/\"finalizers\": \[[^]]\+\]/\"finalizers\": []/" | kubectl $kubeconfig_string replace --raw /api/v1/namespaces/istio-system/services/istio-ingressgateway -f - 2>/dev/null
```

```
kubectl delete ns istio-operator --force 2>/dev/null
```

```
kubectl delete ns istio-system --force 2>/dev/null
```

```
kubectl delete ns avx --force 2>/dev/null
```

```
kubectl delete ns avx-kafka --force 2>/dev/null
```

Troubleshooting

Error	Resolution
403: Access Forbidden - License not available. This error occurs when only the managed kubernetes cluster is restarted every morning.	In case of managed kubernetes, when the cluster is restarted it is recommend to restart all pods.

AppViewX Install and Upgrade for GKE

This guide provides the prerequisites and the procedure for installing, upgrading, and accessing the AppViewX application.

- AppViewX Architecture
- Architecture Overview
- AppViewX Deployment Architecture
- Managed Kubernetes Architecture
- GCP Components
- Prerequisites
- Install AppViewX in Managed Kubernetes
- Upgrade AppViewX in Managed Kubernetes

Copyright © 2025 AppViewX Inc. All Rights Reserved.

• Downloading Images from AppViewX Repository
• Utility Tools and Functions
• Log Collection
• Uninstall and Cleanup
• Troubleshooting

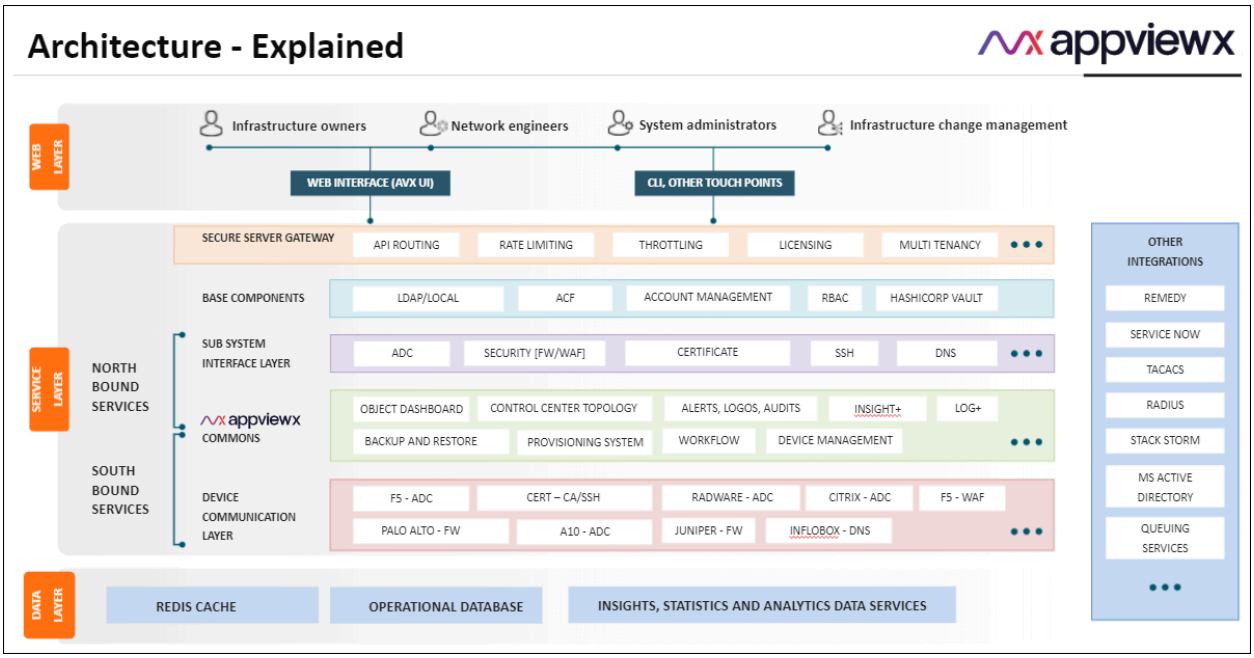
AppViewX Architecture

Architecture Explained

AppViewX is designed based on the microservice architecture and is deployed on Kubernetes—an open-source platform for deploying and managing containers.

The microservice architecture of AppViewX makes it easier to move to containerized workloads and the containers being orchestrated using Kuberneters.

Kubernetes provides container runtime, orchestration, self-healing mechanisms, service discovery and load balancing and it is used for the deployment, scaling, management, and composition of application containers across clusters.



Copyright © 2025 AppViewX Inc. All Rights Reserved.

Benefits of AppViewX Architecture

In order to optimally utilize the resources, AppViewX has adopted Kubernetes to achieve higher security by adopting a zero trust network model. The features of AppViewX coupled with Kubernetes are given below.

- **Auto scaling** - AppViewX services can have a custom throttling capability based on pre-configured memory configuration per API. This will enable AppViewX services to utilize (scale up) resources optimally as the demand surges and scale down when not in use. This will help to horizontally scale the vendor components on demand and optimize the resource usage.
- **Resiliency** - There is no guarantee that AppViewX services may run without any interruptions and they are bound to fail. Kubernetes keeps deployments healthy by restarting containers that have failed, by killing and replacing unresponsive containers based on health checks. This helps to mitigate the common pain point of the application's upkeep process.
- **Security** - AppViewX architecture is designed around the concept of zero trust network model to enforce tighter security within the Kubernetes cluster. This means no one is trusted by default and requires verification to gain access to the services.

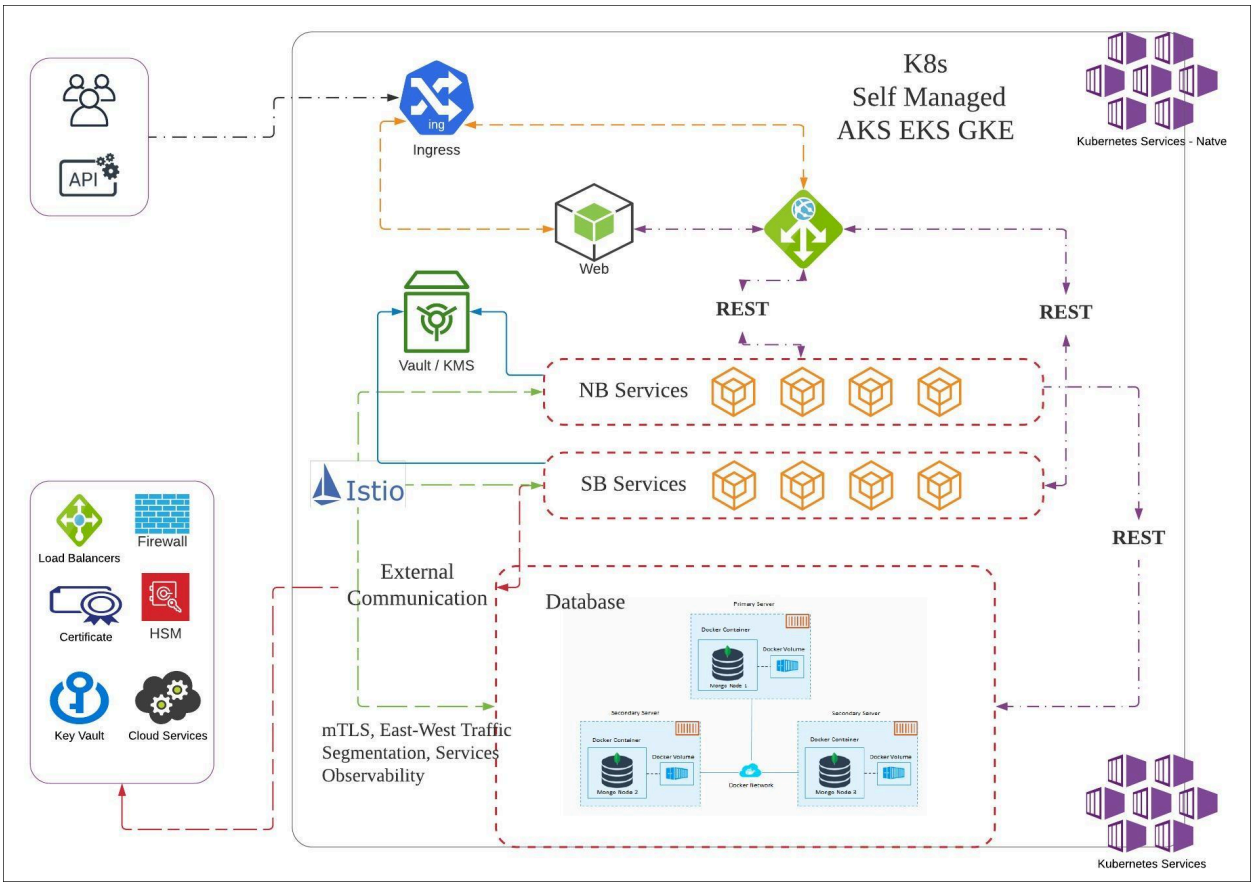
Architecture Overview

AppViewX Kubernetes Architecture

AppViewX workloads are containerized workloads running as microservices and these containers are orchestrated by managed Kubernetes services. Users can prefer the managed k8s platform of their choice.

AppViewX supports deployment on all the three public clouds AWS, Azure and GCP (Google Cloud Platform) using their managed kubernetes engine / services EKS, AKS and GKE specifically.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



Benefits of AppViewX Architecture

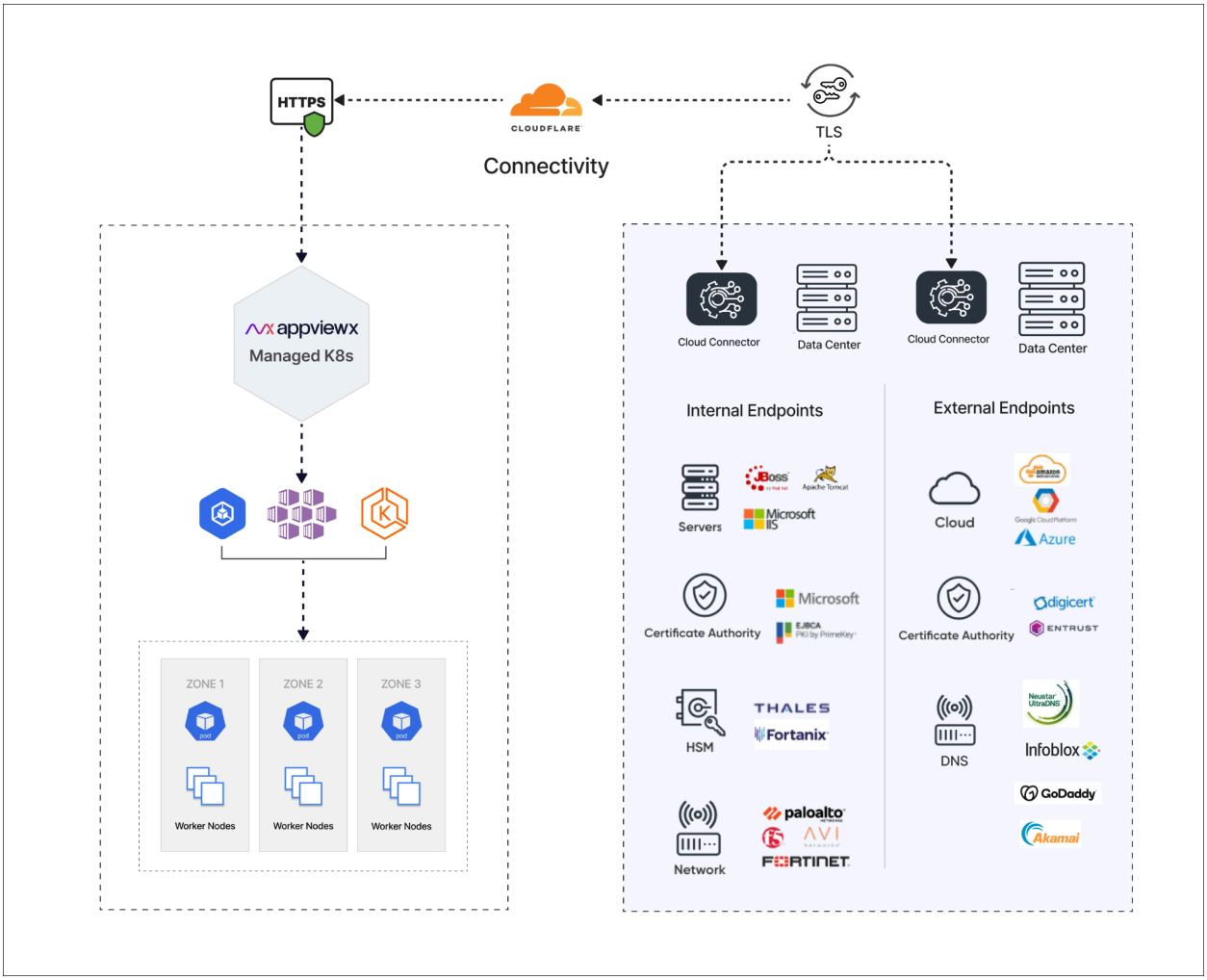
In order to optimally utilize the resources, AppViewX has adopted Kubernetes to achieve higher security by adopting a zero trust network model. The features of AppViewX coupled with Kubernetes are given below.

- **Auto scaling** - AppViewX services can have a custom throttling capability based on pre-configured memory configuration per API. This will enable AppViewX services to utilize (scale up) resources optimally as the demand surges and scale down when not in use. This will help to horizontally scale the vendor components on demand and optimize the resource usage.
- **Resiliency** - There is no guarantee that AppViewX services may run without any interruptions and they are bound to fail. Kubernetes keeps deployments healthy by restarting containers that have failed, by killing and replacing unresponsive containers based on health checks. This helps to mitigate the common pain point of the application's upkeep process.
- **Security** - AppViewX architecture is designed around the concept of zero trust network model to enforce tighter security within the Kubernetes cluster. This means no one is trusted by default and requires verification to gain access to the services.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

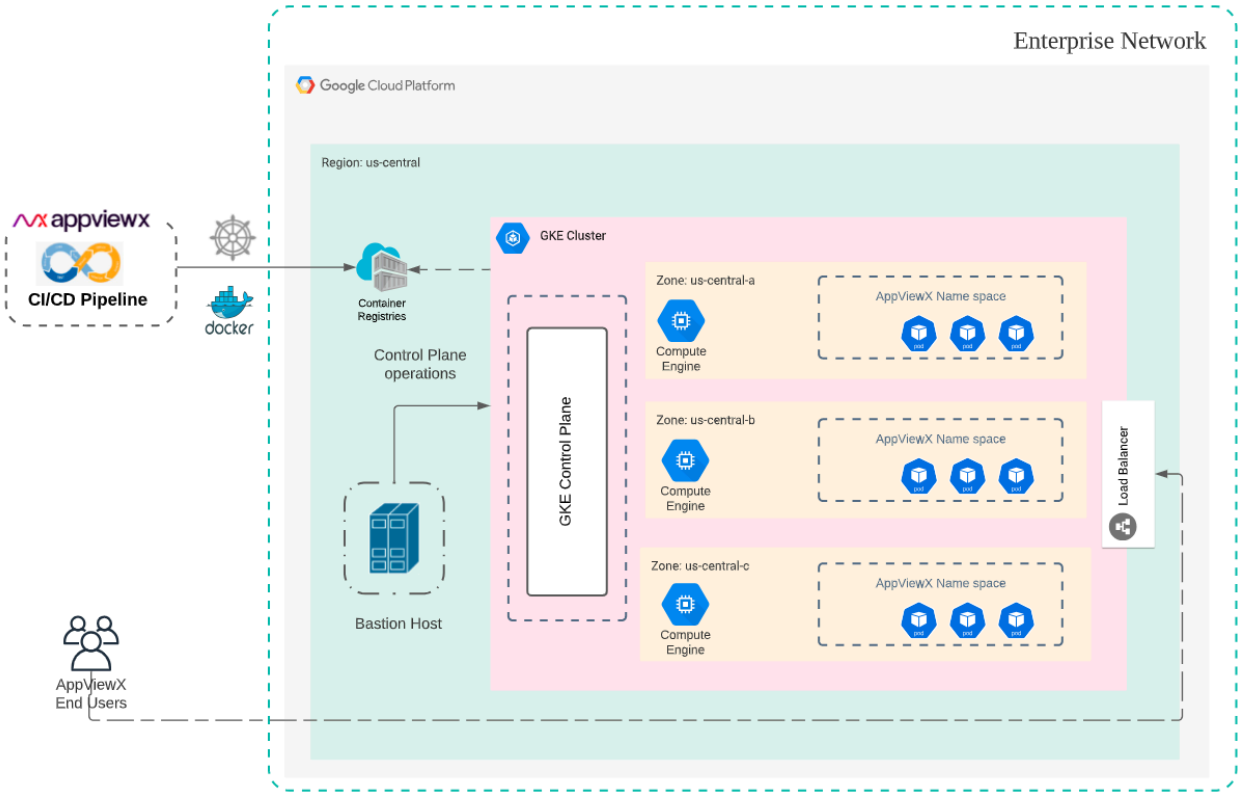
AppViewX Deployment Architecture

The figure below shows a standard AppViewX deployment architecture model via managed Kubernetes service for GKE.



Copyright © 2025 AppViewX Inc. All Rights Reserved.

GKE Deployment Model



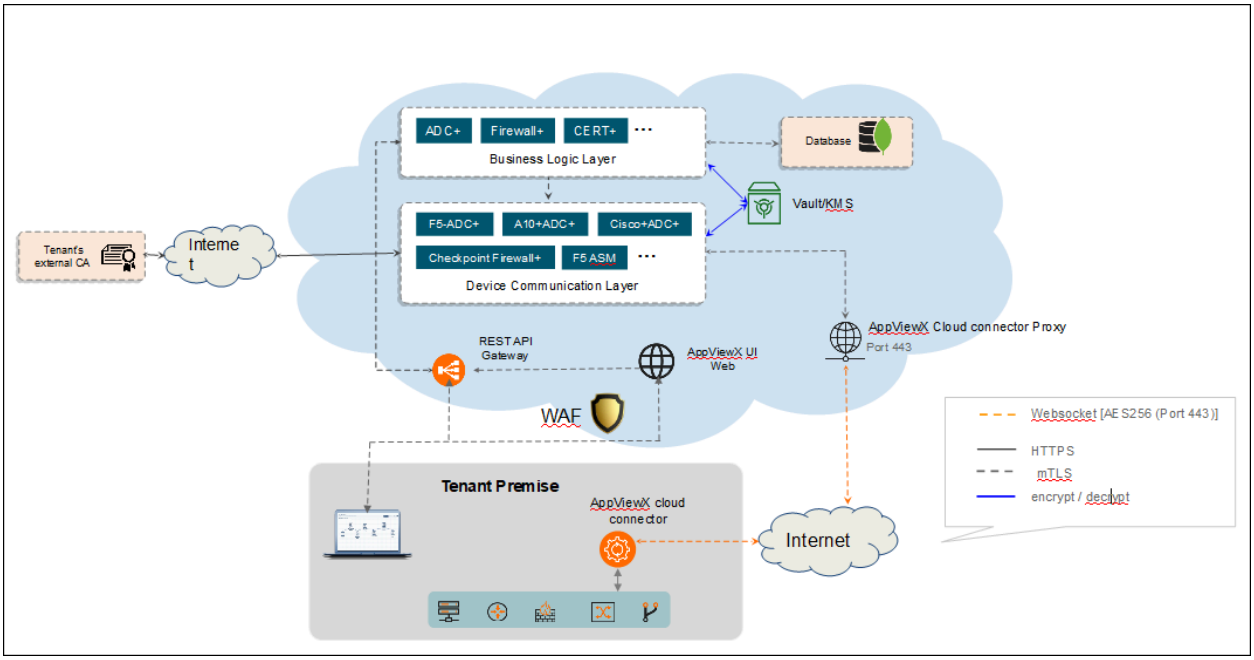
Cloud Connector

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 and your enterprise network without requiring any complex network or infrastructure configuration

Key features of the AppViewX Cloud Connector:

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.



For more details on cloud connectors refer to AppViewX Cloud Connector User Guide.

Note: The below steps have to be performed in all the cloud connector host machines after the 2022.1.0FP2 to FP3 patch upgrade and before the FP3 cloud connector upgrade.

1. Navigate to the installation path in the cloud connector host machine.
2. Execute the following command:

```
./deps/tools/k3s kubectl get deploy avx-mid-server-starter -n cc -o yaml > starter.yaml && sed -i "s/-Xmx2560m/-Xmx4g/g" starter.yaml
&& ./deps/tools/k3s kubectl replace -f starter.yaml
```

Managed Kubernetes Architecture

Managed Kubernetes clusters are composed of the following main components — a control plane and worker nodes. Each cluster runs in its own, fully managed Virtual Private Cloud (VPC).

- The **control plane** is composed of three master nodes, each running in a different Availability Zone to ensure high availability. Incoming traffic directed to the Kubernetes API passes through the respective cloud service load balancer.
- The **worker nodes** run on virtual instances located in a VPC. Managed kubernetes service engine provides managed node groups with automated lifecycle management. This lets users automatically create, update, or shut down nodes with one operation.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Managed Kubernetes service scales the Kubernetes control plane across multiple Availability Zones of the public cloud to ensure high availability and it automatically scales control plane instances based on load, detects and replaces unhealthy control plane instances, and automatically patches the control plane.

Managed Kubernetes workload instances are deployed in multiple availability zones within the region. Each instance has replicas of the services and nodes which exist across all the virtual instances.

Each zone or instance has an active pod listening to other instances. In case of a failure in any instance, the active pod ensures seamless functioning of the application by activating the nodes from any other working cluster.

GCP Components

The following GCP components are utilized by AppViewX:

- Google Kubernetes engine (refer to version support metrics in the next section).
- Storage Bucket for storing MongoDB and Vault backups.
- Service account for accessing storage bucket and GCR registry.

Prerequisites

The following prerequisites must be met before the installation process.

- Managed Kubernetes Version Support Matrix
- Disks Used for AppViewX Installation
- AppViewX Docker Images
- AppViewX Helm Charts
- Bastion Host Setup
- GKE Cluster
- GCP Storage Bucket

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Managed Kubernetes Version Support Matrix

Public Cloud	
Mode of Deployment	**Google**
Release, Vendor, & Product Support **AppViewX v2024.3.0.0**	
Managed K8s Deployment (GKE)	
K8s version 1.33	Yes

Disks Used for AppViewX Installation

Disks Used

Volume	Size	Quantity
logs volume	10Gi	1
avx-kafka	20Gi	3
zookeeper	20Gi	3
openbao	10Gi	3
mongodb	256Gi	3
redis	5Gi	3

If a third party is installed, the values are as follows:

Disks Used (Third Party)

Volume	Size	Quantity
Elasticsearch-ELK	10Gi	1
Elasticsearch-Insight	10Gi	1

Copyright © 2025 AppViewX Inc. All Rights Reserved.

AppViewX Docker Images

AppViewX Docker images are hosted in a private registry https://images.appviewx.com. These images can be pulled using an authentication token (contact AppViewX Support, help@appviewx.com for the authentication token) and can be hosted in the private or public repository at the customer end.

The list of docker images are

- <registry link>/appviewx/pilot:1.26.1
- <registry link>/appviewx/proxyv2:1.26.1
- <registry link>/appviewx/openbao:2.2.2
- <registry link>/appviewx/redis:8.0.2
- <registry link>/appviewx/mongo-init:<tag>
- <registry link>/appviewx/avx-cloud-gateway:<tag>
- <registry link>/appviewx/avx-cloud-web:<tag>
- <registry link>/appviewx/avx-cloud-mongoseed:<tag>
- <registry link>/appviewx/avx-cloud-managedservice-mks:<tag>
- <registry link>/appviewx/avx-platform-report-generator:<tag>
- <registry link>/appviewx/avx-python-sandbox:<tag>
- <registry link>appviewx/avx-mid-server-base:<tag>
- <registry link>/appviewx/kafka:0.46.0-kafka-3.9.0
- <registry link>/appviewx/operator:0.46.0
- <registry link>/appviewx/alpine:3.22.0
- <registry link>/appviewx/kube-metrics-adapter:v0.2.3
- <registry link>/appviewx/kube-state-metrics:v2.15.0
- <registry link>/appviewx/backup-utility-image:v2024.3.0
- <registry link>/appviewx/prometheus:v2.53.4
- <registry link>/appviewx/metrics-server:v0.7.2
- <registry link>/appviewx/elasticsearch:9.0.2
- <registry link>/appviewx/elasticsearch-insight:9.0.2
- <registry link>/appviewx/filebeat:9.0.2
- <registry link>/appviewx/grafana:11.6.2
- <registry link>/appviewx/kibana:9.0.2
- <registry link>/appviewx/logstash:9.0.2
- <registry link>/appviewx/logstash-syslog:9.0.2
- <registry link>/appviewx/alertmanager:v0.28.1
- <registry link>/appviewx/node-exporter:v1.9.1
- <registry link>/appviewx/redis_exporter:v1.73.0

Copyright © 2025 AppViewX Inc. All Rights Reserved.

The steps to download the images from AppViewX repository are as follows:

1. Get the source image repository credentials from AppViewX Support team.
2. Configure the docker using the command

```
docker login -u ${USERNAME} -p ${PASSWORD} ${DOCKER_REPOSITORY}
```

3. Configure the respective cloud provider CLI (Google cloud) and ensure you have access to push docker images to GCR.
4. To push the docker images, use the helper script provided by AppViewX. Follow the steps below.

   a. Download the artifact Managed-Kubernetes_helper_scripts.tar.gz to the bastion host and extract using the command:

   ```
   tar -xf Managed-Kubernetes_helper_scripts.tar.gz
   ```

   b. Navigate to the extracted directory **mk8s_helper_scripts**.

   ```
   cd mk8s_helper_scripts
   ```

   c. Execute the script **avx_image_pull_push.sh** using the command

   ```
   ./avx_image_pull_push.sh <Image tag> <customer registry url>
   ```

   > **Note:** Replace <Image tag> with **24.3.0.0-1505** and <customer registry url> with the actual values.

AppViewX Helm Charts

The helm charts used by AppViewX for installation are released as a part of the installer. The installer consists of helm charts and an AppViewX utility which helps orchestrate the deployment, patch, upgrade and maintenance of AppViewX across managed kubernetes deployment.

Bastion Host Setup

The following packages must be installed on the bastion host or the host/tool (Azure DevOps) from where the installation is triggered

GCP CLI

To set up the GCP CLI refer to Install the gcloud CLI on the Google documentation website.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Kubectl

To set up Kubectl refer to Install and Set Up kubectl on Linux on the Kubernetes documentation website.

Execute the following commands

- `sudo curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"`
- `sudo chmod +x kubectl`
- `sudo mv ./kubectl /usr/bin/#`

Verify installation by executing the command

```
kubectl version
```

Helm

Helm is required only if the deployment is triggered from any other machine instead of the DevOps pipeline. To set up Helm refer to Installing Helm on the Helm documentation website.

Execute the following command:

- `curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3`
- `chmod 700 get_helm.sh`
- `./get_helm.sh#`

Verify installation by executing the command

```
helm version
```

GKE Cluster

To create an GKE cluster refer to Google cloud documentation website - Creating a regional cluster. Although Google cloud manuals are always up-to-date, the recommended choice to make before creating the cluster is as follows:

- Kubernetes version: 1.31
- User nodepool:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- **appnodepool**: Three nodes of type **n2-standard-8** with Auto Scaling disabled
- **mongonodepool**: Three nodes of type **n2-standard-8** with Auto Scaling disabled. Add label **mongo=true** and taint **designatedMongo=true:NoSchedule** to the nodepool (to be performed while creating the cluster).

• Select multi zones for the Nodepools

GCP Storage Bucket

A storage bucket is required to store

• iControlJar: Container name is **icontroljar** and the jar has to be placed here.
• MongoDB backup: Container name should be **mongo-backup**.
• Vault backup: Container name should be **vault-backup**.
• Axisjar, javax.xml.soap-api: Jars needs to be added here - **axisjar**.

A summary of steps for creating the storage bucket is as follows:

> **Note:** After the script is executed, capture the output **Annotation** which is required in the global utility config.

1. Create a storage account with a valid name to indicate the storage account for a specific GKE cluster.
2. Configure Storage buckets and Image Registry access for GKE nodes.
3. The first workload identity should be enabled cluster wide. This operation may be performed from the portal after the cluster creation or at the time of cluster creation. Refer google document Using Workload Identity
4. The above steps can also be performed using a helper script provided by AppViewX. To use this script follow the steps below.
   a. Download the artifact Managed-Kubernetes_helper_scripts.tar.gz to the bastion host and extract using the command:

      ```
      tar -xf Managed-Kubernetes_helper_scripts.tar.gz
      ```

   b. Navigate to the extracted directory **mk8s_helper_scripts**.

      ```
      cd mk8s_helper_scripts
      ```

   c. Edit the file **gcp_sc_config.sh** and replace <PROJECT_ID>, <CLUSTER_NAME>, <NODE_POOL_1,NODE_POOL_2>, <REGION_NAME> with the actual values.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

d. Execute the **gcp_sc_config.sh** file.

```
bash gcp_sc_config.sh
```

5. Store output of the script in step 4d and pass the annotation in the global utility config **serviceAccountAnnotation** (refer the second table in Installation Step 7)

Install AppViewX in Managed Kubernetes

Migration Strategy

Attention: If you are performing a fresh install, then refer the next sub-topic **Installation Steps**.

To migrate from AppViewX on-prem versions (2022.1.0, 2021.1.0, and 2020.3.0) to Managed Kubernetes, it is important to take a backup of the mongodb and vault in the respective on-prem versions. Before you take the backup, execute the script below.

```
db.profile.update({'_id' : 'installationType'}, {$set : {"value" : "Managed_K8s"}})
```

Note: Refer to the specific version of the release documents from the release portal and perform the backups or contact the AppViewX support team.

After performing the backup, follow the installation steps detailed in the section below. At step 11 of the installation process, ensure to restore the data at this stage.

Installation Steps

This section describes the steps to for installing the AppViewX Stack on AKS.

1. Download the installer from the release portal.
2. Create a directory Managedk8s-installer in the bastion host and extract the installer file **tar -xf installer.tar.gz** in the same directory.
3. Verify that the extracted installer must have the following files
   - appviewxctl (binary)
   - helm_charts (directory of helm charts)
4. Generate the configuration files based on the cloud provider. If the cloud provider is **Google**, execute the command below.

```
./appviewxctl config generate --provider gcp
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

5. Verify that the execution of the above command creates the configuration files named **.appviewxctl.yaml** in the same location.
6. The file .appviewxctl will be populated with the fields necessary for installation, in particular cloud provider that was provided in the previous command (**-- provider**).
7. Edit the .appviewxctl.yaml file and populate the values as described below:

appviewxctl.yaml file - Parameters and Description

Parameters	Description of Values
chartPath	The path to the helm_charts which is to be installed. It points to the helm_charts directory extracted in step 3.
configFile	The path to the kube config file to be used by helm and kubectl. If the bastion host is already configured and kube config is under $HOME/.kube directory, then keep this field empty.
install.enableAppBackupCron	Boolean value to enable/disable the backup cronjobs. (True/False). This value is needed for self-managed mongo only. For atlas backup this has to be scheduled in the atlas dashboard.
install.enablePrivateImagePullSecret	Boolean value to enable image pull secret. Set values as **false** if the cluster already has access to the container registry. Otherwise set it to **true** and fill all the details of the access keys described in below sections.
install.enableThirdPartyInstall	Boolean value (True/False) to determine whether third party monitoring components such as ELK, Monitoring, and Insight needs to be installed.
install.thirdPartyApp.elk	Boolean value to add Elk component. Set to True if it needs to be installed.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Parameters	Description of Values
install.thirdPartyApp.monitoring	Boolean value to add Monitoring component. Set to True if it needs to be installed.
install.thirdPartyApp.insight	Boolean value to add Insight component. Set to True if it needs to be installed.
install.imageRegistry	The URL of the container registry where the images are to be pulled from by the pods. *Example*: gcr.io/pe-qa-358108
install.imageTag	The tag of the image that will be used for installation. *Example*: 2023.1.0_FP_750-alpine
install.isSaasEnabled	Boolean value for SaaS enablement. This value should be set to **true** for Managed K8s.
install.kafkaCloudConnector	It is a combination of three values. • enable • password • user Set **enable** to ***true*** and keep the user, password fields empty for Managed K8s. *Example* kafkaCloudConnector: enable: true password: "" user: ""
install.mongo	It is a combination of fields specific to the type of mongodb used.
dbIsolation	Boolean value to indicate whether the database isolation is to be enabled.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Parameters	Description of Values
	In order for database isolation to work, the following prerequisite must be taken care of while creating the cluster node group. • Add label **mongo=true** and taint **designatedMongo=true:NoSchedule** to the nodepool to be used for mongodb.

mongoAtlas	The fields specific to mongodb atlas are as folows:
	• **enable**: Boolean value to decide if mongodb atlas to be used. If set to *false*, a self managed mongodb cluster will be created. If set to *true* mongodb atlas will be used and details of which are to be provided in below mentioned fields.
	• **host**: URL of the mongodb atlas cluster.
	• **password**: password of the mongodb atlas cluster.
	• **user**: username in the mongodb atlas cluster.
	Example:
	`mongo:`
	`  dbIsolation: false`
	`  mongoAtlas:`
	`  enable: true`
	`  host: "managed-k8s.test.mongodb.net"`
	`  password: "samplepassword"`
	`  user: "user1"`

install.useDockerPrivateRegistry	Set this to **true** if the dockerhub private repository is to be used for pulling the necessary images needed. Otherwise set the value **false** and the container registry ACR, ECR, and GCR will be used based on the cloud provider.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Parameters	Description of Values

	If this value is set to *true*, populate the below values, otherwise keep it empty.
	• **dockerhub.pass**: password to be used for authenticating in the dockerhub private repository.
	• **dockerhub.username**: username configured in the dockerhub private repository.
	
	Example:
	`useDockerPrivateRegistry: true`
	`dockerhub:`
	`pass: "testpassword"`
	`username: "appviewx"`

install.size	The size of the installation. Based on the use cases and number of certs to be managed there different sizes (contact AppViewX for sizing recommendations). The sizes supported are (case sensitive values) • xsmall • small • medium • large • xlarge • custom *Example:* `size: small` **Note:** The size provided must be taken into cluster creation and nodegroup sizes must be defined accordingly.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Parameters	Description of Values

install.plugins	The list of plugins that will be installed. Each plugin will have three fields
	• enable
	• imageTag
	• name
	
	Set enable to **true** if the plugin is to be installed. If the same image tag is to be used as defined in the global ImageTag keep it **latest** otherwise override with some other tag of your choice.
	
	Example:
	
	`- enable: true`
	`imageTag: latest`
	`name: avx-config-server`
	
	To enable Cloud DC support in Managed Kubernetes, set plugins as follows:
	
	`- enable: true`
	`imageTag: latest`
	`name: avx-mid-server-platform`
	
	Note: Ensure that **install.isSaasEnabled** and **install.kafkaCloudConnector** are set to **true**.

internalLoadBalancer	If set to **true**, all the Loadbalancers will be private and can only be accessed within the VPC else it will be public.
install.enableSftpStorage	Change to true to use SFTP server for mongodb, vault, and icontrol.jar storage. Boolean (Default: false)
install.sftpServerDetails. dbBackupPath	Provide the location of mongodb backup storage directory. String (Default: “”)

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Parameters	Description of Values
install.sftpServerDetails. vaultBackupPath	Provide the location of vault backup storage directory. String (Default: “”)
install.sftpServerDetails. sftpServerUserName	Provide the username of SFTP server. String (Default: “”)
install.sftpServerDetails. sftpServerIp	Provide the sftp server IP. String (Default: “”)
cloudConnectorEnabled	A boolean value (true/false) to denote the cloud connector usage for southbound communications. If a cloud connector is used set the value to **true**.
socketurl	Customers can now configure their own web-socket URL in the MongoDB by updating the `socketurl` parameter in the **.appviewxctl.yaml** file. After installation or upgrade, the specified socket URL will be recorded in the **saas_agent_meta** collection.

The next fields are to be filled with values that must be collected during the cluster creation and setup process and filled as mentioned below.

Parameters	Description of Values
install.privateImagePullSecret	In this section populate the details of the access keys needed to authenticate and pull the image from the registry. They are not needed if the Dockerhub is used as described above. • **registry**: The registry whose token must be provided below and used to pull images. • **token**: The login token for the registry used. Token can be generated from CLI if authenticated in the CLI from the respective google cloud account. A sample command to generate token of gcr.io registry `gcloud auth print-access-token |docker login -u oauth2accesstoken --password-stdin https://gcr.io` *Example:*

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Parameters	Description of Values
	registry: "gcr.io/pe-qa-358108" token: "sample token"
install.storageAccess	The storage bucket details to be used for setting up backup capability. • **bucketObject**: The name of the bucket object. • **serviceAccountAnnotation**: Annotation of service account that provides access to the storage bucket *Example:* bucketObject: "appviewx-samplebucket" serviceAccountAnnotation: "avx-storage-bucket-access-gsa@sampleproject.iam.gserviceaccount.com"

The following fields must be added to integrate the kubernetes cluster to the external vault.

appviewxctl.yaml file - Parameters and Description (for external vault)

Parameters	Description
install.externalVault.enable	A boolean value (true/false) to denote if the external vault is to be used in the setup. True is to enable the external vault.
install.externalVault.externalVaultAddr	Contains the vault URL and listening port *Example*: https://pm-lxs-node01.lab.appviewx.net:8200
install.externalVault.externalVaultAuthRole	Name of the role created against the access kubernetes auth path
install.externalVault.externalVaultCACertSecret	Name of the secret where **vault-ca.crt** file is mounted.
install.externalVault.externalVaultDBRole	Static role created to access the database cred.
install.externalVault.externalVaultEnginePath	Enter the value “/database”

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Parameters	Description
install.externalVault.externalVaultKubeAuthPath	The Kuberenetes access path created with cluster information for service account authentication.
install.externalVault.externalVaultSAName	The Service account used to create externalVaultAuthRole.
install.externalVault.mongoPasswordVaultEngine	Enter the value DATABASE

8. Once the values are filled in *.appviewxctl* as described in the step above, proceed with the installation. Before doing so, check if the the preconditions are met by executing the command

```
./appviewxctl preflight --config .appviewxctl.yaml
```

   This will prompt if the necessary prerequisites are met.

9. The metrics server in the GCP clusters comes pre-installed with the cluster, hence they must be disabled from the **avx_pre_req** chart.
   a. Navigate to helm_charts/avx_pre_req.
   b. Edit the **values.yaml** file by setting the following parameters.

```
avx-metrics-server:

enable: false
```

   The metrics server installation is disabled.

10. To proceed with installation, execute the command

```
./appviewxctl install --config .appviewxctl.yaml
```

> **Note:** The installation will take several minutes to complete. Upon completion you see the following message:
>
> [Install] Successfully installed Appviewx infra stack
>
> This would imply the completion of infra component setup.

11. This step involves restoring the existing data from the previous AppViewX version's cluster in case there is a need to migrate from the older versions to the Managed K8s version. **Ignore this step if it's a fresh setup with no migration necessary.**

    To restore mongodb and vault fetch the backup files and place them in the bastion in a directory such as /home/user/backup execute the mongo_restore and vault_restore scripts as follows:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Note: Before you trigger the `mongo_restore.sh` command, ensure that the mongo restore file has the file extension as **.tar.gz**.

```
./mongo_restore.sh <mongo backup tar filepath>
```

```
./vault_restore.sh -p <vault backup filepath> --removedek
```

Attention: If the data is being restored from an older version (2020.3.0 - 2022.1) then use the command

```
./vault_restore.sh -p <vault backup filepath> --removedek
```

Note:

- The backup files must have extension as **.tar.gz**
- The above commands work for a self-managed mongodb setup. Setting up the mongodb atlas requires the installation of mongodb tools in the bastion host as described below.

For an rpm based OS:

```
echo -e "[mongodb-org-4.2] \nname=MongoDB
 Repository\nbaseurl=https://repo.mongodb.org/yum/redhat/\$releasever/mongodb-org/4.2/x86_64/\ngpgcheck=1\nenabled=1\ngpgkey=https://
www.mongodb.org/static/pgp/server-4.2.asc" > /etc/yum.repos.d/mongodb-org-4.2.repo
yum install mongodb-org-shell-4.2.0
yum install mongodb-org-tools-4.2.0
```

For a debian based OS:

```
wget -qO - https://www.mongodb.org/static/pgp/server-6.0.asc | sudo apt-key add -
sudo apt-get install gnupg
wget -qO - https://www.mongodb.org/static/pgp/server-6.0.asc | sudo apt-key add -
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0 multiverse" | sudo
 tee /etc/apt/sources.list.d/mongodb-org-6.0.list
sudo apt-get update
sudo apt-get install -y mongodb-mongosh
sudo apt-get install -y mongodb-org-tools
```

Verify if the mongodb restore commands have executed successfully using the command

```
mongorestore -- version
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

12. To proceed with the AppViewX application installation, execute the command:

```
./appviewxctl installapp --config .appviewxctl.yaml
```

Once installation is complete the following messages are displayed:

```
[Install] Appviewx infrastructure chart [avx-app] installed successfully
[Install] Successfully installed Appviewx application stack
[Install] Fetching login URL for app
[Install] Waiting for Public IP allotment for istio service
[Install] AppViewX Web URL: https://34.100.197.159/appviewx/
[Install] AppViewX Gateway URL: https://34.100.197.159/avxmgr/
[Install] Grafana URL: https://34.100.197.159/grafana/
[Install] Kibana URL: https://34.100.197.159/kibana/login
[Install] Run below commands to get mongo user credentials
 export MONGO_USER=$(kubectl get secret -n avx mongo-key -o=jsonpath='{.data.mongo-init-user}' | base64 -d)
 export MONGO_PASS=$(kubectl get secret -n avx mongo-key -o=jsonpath='{.data.mongo-init-pass}' | base64 -d)
[Install] Run below commands to get Elasticsearch and Kibana credentials
 export ES_PASS=$(kubectl get secret -n avx elasticsearch-pw-elasticsearch -o=jsonpath='{.data.password}' | base64 -d)
 export KIBANA_PASS=$(kubectl get secret -n avx elasticsearch-pw-kibana -o=jsonpath='{.data.password}' | base64 -d
[Install] Application Installation completed successfully
```

Note: Follow the URLs and commands given in the output message to get the credentials and access the application.

13. If installation of the third party monitoring components was not enabled during the entire process, they can be installed later by the following steps:

    a. While installing the third party components (helm_charts/avx_third_party/values.yaml), the only that values are set to 'true' by default are - *prometheus, nodeexporter, kube-state metrics*. The other components are set as 'false' by default and must be to set to true if they are to be enabled, they are - *elk-elasticsearch, elk-filebeat, elk-kibana, elk-logstash, grafana, elasticsearch-insight, logstash-syslog.*

    b. Edit the **.appviewxctl.yaml** file and set **install.enableThirdPartyInstall** to ‘true’

    c. Configure the following **thirdPartyApp** parameters as true as per the requirements:
    - **install.thirdPartyApp.elk**
    - **install.thirdPartyApp.monitoring**
    - **install.thirdPartyApp.insight**

Copyright © 2025 AppViewX Inc. All Rights Reserved.

d. Now, edit the file **values.yaml** present at location helm_charts/appviewx_monitoring/prometheus/chart/values.yaml and append the below values at the end of the file (only if that are not present).

```
limits:
 cpu_limit: 80
 memory_limit: 80
 disk_limit: 80
 timelimit_cpu_memory: 5
 timelimit_disk: 1
 timelimit_pod: 1
 timelimit_node: 1
```

e. Run the command below

```
./appviewxctl installtpt --config .appviewxctl.yaml
```

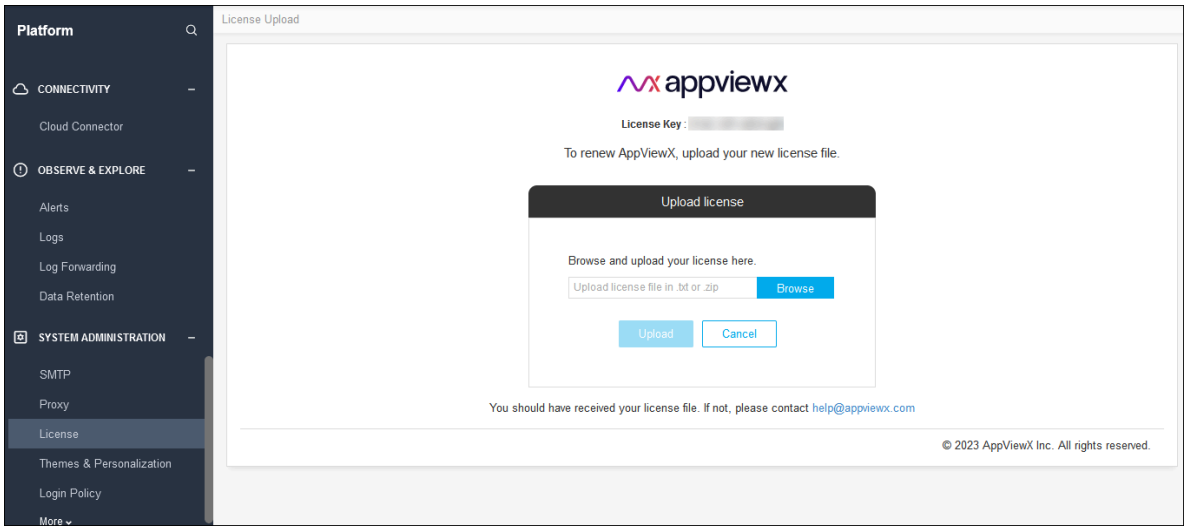
What to do next

1. If you have to install the Helm based CC, then parch the internal tenant secret, using the command below:

   ```
   kubectl patch secret internal-tenant -n avx --type='json' -p='[{"op": "add", "path": "/data/REPO_AUTH_TOKEN", "value": "<repoauthtokenvalue>"}]'
   ```

   Once the command is executed, restart the saas-proxy.
2. Customers migrating from AppViewX version 2020.3.0 to Managed Kubernetes, it is mandatory to upgrade the license.
   a. Login to the AppViewX with valid credentials.
   b. Navigate to Platform >> System Administration >> License page.
   c. Click **Upgrade License**.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



d. Click **Browse** to find the latest license key file.

e. Click **Upload**.

> **Note:** For the licenses contact AppViewX Support at help@appviewx.com or customerlicences@appviewx.com.

Customers migrating from AppViewX version 2020.3.0 to Managed Kubernetes FP3, it is mandatory to upgrade the license.

Upgrade AppViewX in Managed Kubernetes

Before you begin: Before performing the infra upgrade, upgrade the kubernetes to the latest version **1.31**. Refer to the manually upgrading a GKE cluster or node pool webpage.

> **Attention:**
>
> - If you are using the self managed private docker registry instead of AppViewX's docker registry, then before proceeding with the upgrade, ensure you have copied the latest images to your registry. The list of images can be found in the Prerequisite section - AppViewX Docker Images.

To upgrade AppViewX with a new image version, follow the steps below:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

1. To take the backups, execute the commands below.

   For self-managed mongodb:

   ```
   kubectl create job --from=cronjob/mongo-backup -n avx mongo-backup-<unique-identifier>
   ```

   ```
   kubectl create job --from=cronjob/vault-backup -n avx vault-backup-<unique-identifier>
   ```

   Replace <unique-identifier> in above commands with some random string and run. Monitor the pods until completion and verify the backups are placed in the storage bucket.

   Note: Atlas backup must be taken in the atlas dashboard. Refer to the atlas snapshots section in the page Backup and Restore.

2. Before performing the Infra Upgrade, update the following parameters.
3. Download the upgrade tar file (**upgrade.tar.gz**) from the release portal and extract it to a suitable location. (The extracted files contain the binary and helm charts tar.)
4. Navigate to the folder where the upgrade tar is extracted.
5. Copy the **appviewxctl** binary from the current folder (extracted folder location) to the installer location.

   ```
   cp appviewxctl <absolute path of the installer directory>
   ```

6. Navigate to the installer directory.
7. To merge the old data of .appviewxctl.yaml to the new file, execute the command below.

   ```
   ./appviewxctl mergeConfForUpgrade --config .appviewxctl.yaml
   ```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
eksuser@bastion-hsot-qa-1:~/mar6$ ./appviewxctl mergeConfForUpgrade --config .appviewxctl.yaml
[Init] Using log file at [/home/eksuser/mar6/appviewxctl-554518342.log] to dump logs
[Init] Initialise persistent flag config
[Init] Using config file
Created backup file: .appviewxctl-old-20250310-134744.yaml
[Config] Generating config file with provider gcp - .appviewxctl.yaml
[Config] Successfully generated sample config file.
Waiting for config generation to complete...
Created backup file: .appviewxctl-tmp-bkp.yaml
Plugin 'avx-vendors-sync' has been added under install key. Do you want to enable it? (y/n): y
Field 'socketurl' has been added under root key. Please enter the new field value (default: ): web.socket.com
Plugin 'avx-vendors-sync' has been added under upgrade key. Do you want to enable it? (y/n): n

Summary of New Fields:
-------------------------------------------------------------
Field Path                          | Value
-------------------------------------------------------------
avx-vendors-sync                    | enable: true
socketurl                           | web.socket.com
avx-vendors-sync                    | enable: false
-------------------------------------------------------------

Do you want to proceed with these changes? (y/n): y
Merged old YAML values into new YAML.

Please enter the output file name for the new YAML. Hit enter to take the default value. (default: .appviewxctl.yaml):
Successfully saved YAML to .appviewxctl.yaml
```

During the execution of the above command, the fields of the old conf file are compared with that of the newly generated conf file and you will be prompted to enter values.

- If it's a plugin, you'll be prompted to enable or disable it during both installation and upgrade.
- If a new field has been added to the `.appviewxctl.yaml` file, you'll be prompted to provide a value for it. Pressing Enter without input will apply the default value.

Once all the values are entered, the process continues as follows:

- A summary table is displayed to confirm the provided inputs.
- If the user declines, the prompts will repeat.
- If the user confirms, the system proceeds to merge the old YAML with the new one.
- The values provided by the user for the newly added fields will be retained, while existing fields will retain values from the old configuration file.
- Users can specify a custom output filename for the merged YAML; if not specified, the default will be `.appviewxctl.yaml`.
- The merged YAML is successfully saved to the output file.

8. Refer to the tables below for the essential upgrade parameters and their description.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Upgrade Parameters in the appviewxctl.yaml file

Parameters	Description of Values
upgrade.imageRegistry	The URL of the container registry where the images are to be pulled from by the pods. *Example*: gcr.io/pe-qa-358108
upgrade.imageTag	The tag of the image that will be used for upgrade. *Example*: 2023.1.0_FP_750-alpine
install.imageTag	The tag of the image that will be used for installation. *Example*: 2022.1.0_FP_750-alpine
upgrade.isSaasEnabled	Boolean value for SaaS enablement. This value should be set to **true** for Managed K8s.

upgrade.plugins	The list of plugins that will be installed. Each plugin will have three fields
	• enable
	• imageTag
	• name
	
	Set enable to **true** if the plugin is to be upgraded. If the same image tag is to be used as defined in the global ImageTag keep it **latest** otherwise override with some other tag of your choice.
	
	Note: The list of plugins to be enabled should match the ones in the install section.
	
	Example:
	
	`- enable: true`
	`imageTag: latest`

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Parameters	Description of Values
	`name: avx-config-server`
upgrade.upgradeInfra	Boolean value to upgrade infra component. Set to True for upgrade.
upgrade.upgradeThirdParty	Boolean value to upgrade the monitoring (ELK, insight, and monitoring) components. Set to True for upgrade.
socketurl	Customers can now configure their own web-socket URL in the MongoDB by updating the `socketurl` parameter in the **.appviewxctl.yaml** file. After installation or upgrade, the specified socket URL will be recorded in the **saas_agent_meta** collection.

Component Parameters in the appviewxctl.yaml file

Parameters	Description of Values
install.thirdPartyApp.elk	Boolean value to add Elk component. Set to True for upgrade.
install.thirdPartyApp.monitoring	Boolean value to add Monitoring component. Set to True for upgrade.
install.thirdPartyApp.insight	Boolean value to add Insight component. Set to True for upgrade.

Install parameters in the appviewxctl.yaml file required to integrate the kubernetes cluster to the external vault.

Parameters	Description of Values
cloudConnectorEnabled	A boolean value (true/false) to denote the cloud connector usage for southbound communications. If a cloud connector is used set the value to **true**.
install.externalVault.enable	A boolean value (true/false) to denote if the external vault is to be used in the setup. True is to enable the external vault.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Parameters	Description of Values
install.externalVault.externalVaultAddr	Contains the vault URL and listening port *Example*: https://pm-lxs-node01.lab.appviewx.net:8200
install.externalVault.externalVaultAuthRole	Name of the role created against the access kubernetes auth path
install.externalVault.externalVaultCACertSecret	Name of the secret where **vault-ca.crt** file is mounted.
install.externalVault.externalVaultDBRole	Static role created to access the database cred.
install.externalVault.externalVaultEnginePath	Enter the value “/database”
install.externalVault.externalVaultKubeAuthPath	The Kuberenetes access path created with cluster information for service account authentication.
install.externalVault.externalVaultSAName	The Service account used to create externalVaultAuthRole.
install.externalVault.mongoPasswordVaultEngine	Enter the value DATABASE

Note: Path parameters should have a leading forward slash '/'.

9. Copy the **vault_restore_openbao_migration.sh** file from the current folder (extracted folder location) to installer location

Note: If this step was completed during the previous upgrade (2024.3.0.0), you can skip it. However, if you are upgrading from an older version (2023.1.3 or earlier), this step is required.

```
cp vault_restore_openbao_migration.sh <absolute path of the installer directory>
```

10. Copy the **mongo_restore.sh** file from the current folder (extracted folder location) to installer location

```
cp mongo_restore.sh <absolute path of the installer directory>
```

11. To upgrade AppViewX infra, execute the command

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Note: If you plan on enabling additional 3pt monitoring components as part of the infra upgrade do the following:

a. Navigate to <installer>/helm_charts/avx_thrid_party/.

b. Edit the **values.yaml** file.

c. Set "enable" to true for the components you wish to enable as part of the upgrade.

```
./appviewxctl infraUpgrade --config .appviewxctl.yaml
```

```
^C
eksuser@bastion-hsot-qa-1:~/mar6$ ./appviewxctl infraUpgrade --config .appviewxctl.yaml
[Init] Using log file at [/home/eksuser/mar6/appviewxctl-4041704234.log] to dump logs
[Init] Initialise persistent flag config
[Init] Using config file
Ensure that you have taken vault backup before proceeding with infra upgrade. If backup taken enter 'yes' and proceed.
Note: Vault backup command
kubectl create job --from=cronjob/vault-backup -n avx vault-backup-<unique-identifier>
Vault Backup taken (Yes/No): Yes
Ensure that you have taken mongo backup before proceeding with infra upgrade. If backup taken enter 'yes' and proceed.
Note: Mongo backup command
kubectl create job --from=cronjob/mongo-backup -n avx mongo-backup-<unique-identifier>
mongo Backup taken (Yes/No): Yes
Please provide the path of updated helm charts tar:
```

This will prompt the following message

```
Vault Backup taken (Yes/No):
```

If you haven't taken the backup, enter **No**, and execute the vault backup command in step 2. If you have already taken the backup, enter **Yes**. You will be prompted with the message below.

This will prompt the following message

```
Mongo Backup taken (Yes/No):
```

If you haven't taken the backup, enter **No**, and execute the vault backup command in step 2. If you have already taken the backup, enter **Yes**. You will be prompted with the message below.

```
Please provide the path of updated helm charts tar. :
```

Enter the absolute path of the extracted Helm charts tar artifact to proceed with the infrastructure upgrade. Ensure you provide the full file path, not just the directory location, for example: /home/aksuser/mar06/upgrade_avx/helm_charts.tar.gz. Appropriate error messages will be displayed if: only the directory (without file name), wrong file type, or incorrect tar.gz values are entered.

Note: After completing the infrastructure upgrade and before triggering the **installMongoRS**, ensure the `install.imageTag` in the `.appviewxctl.yml` file is updated with the latest imageTag.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

12. Install **Mongo Replicaset** using the command below

> **Note:** If this step was completed during the previous upgrade (2024.3.0.0), you can skip it. However, if you are upgrading from an older version (2023.1.3 or earlier), this step is required.

```
./appviewxctl --config .appviewxctl.yaml installMongoRS
```

13. Install **OpenBao** using the command below.

> **Note:** If this step was completed during the previous upgrade (2024.3.0.0), you can skip it. However, if you are upgrading from an older version (2023.1.3 or earlier), this step is required.

```
./appviewxctl --config .appviewxctl.yaml installOpenbao
```



> **Note:** The plugin pods must not be restarted after installing OpenBao until the plugin upgrade is complete. Restarting the pods prematurely may disrupt functionality, as Vault could lock the user if it detects multiple login attempts.

14. To restore the vault, fetch the backup files and place them in the bastion in a directory such as /home/user/backup execute the restore scripts below.

> **Note:** If this step was completed during the previous upgrade (2024.3.0.0), you can skip it. However, if you are upgrading from an older version (2023.1.3 or earlier), this step is required.

```
./vault_restore_openbao_migration.sh -p <absolute path of the vault_bkp_file_location>
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

15. To restore the mongo, fetch the backup files and place them in the bastion in a directory such as /home/user/backup and execute the restore scripts below.

> **Note:**
> a. If this step was completed during the previous upgrade (2024.3.0.0), you can skip it. However, if you are upgrading from an older version (2023.1.3 or earlier), this step is required.
> b. Before you trigger the `mongo_restore.sh` command, ensure that the mongo restore file has the file extension as **.tar.gz**.

```
./mongo_restore.sh <absolute path of the mongo_bkp_file_location>
```

16. Run the plugin upgrade command below.

```
./appviewxctl upgrade --config .appviewxctl.yaml
```

Rollback Steps

a. Restore the DB using the restore scripts (step 11 in the Installation Steps section) for self-managed DB or in atlas using snapshot restore in the dashboard.

b. Update the **appviewxctl.yaml** upgrade section's values to the previous image tag and re-run the upgrade command.

- Troubleshooting for Upgrade Errors
- MongoShardDB Cleanup
- Vault/Consul Cleanup
- OpenBAO Cleanup
- Mongo Replicaset Cleanup

Post-Upgrade Actions

The following steps have to be performed after you completed the upgrade successfully.

1. Retrieve the updated **web login IP** by executing:

```
kubectl get svc -n istio-system
```

From the output, locate the `ExternalIP` associated with the `istio-ingressgateway` and use that IP to access the web login page.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

2. Update the **RSocket URL** in the properties file and restart the `avx-mid-server-starter` and `avx-mid-server-platform` pods.

   a. In the UI, go to **Platform** > **Cloud Connector**.

   b. Copy the **Cloud Connector URL**.

   c. On the Cloud Connector (CC) node CLI, go to the properties directory:

   ```
   cd <Installer_Folder>/deps/properties
   ```

   d. Open the `appviewx.properties` file using a text editor:

   ```
   vi appviewx.properties
   ```

   e. Update the value of `RSOCKET_URL` with the copied Cloud Connector URL (e.g., `https://98.70.232.0:443`) and save the file.

   f. Restart the following pods to apply the changes:
   - avx-mid-server-starter
   - avx-mid-server-platform

   ```
   kubectl delete pods -n cc avx-mid-server-starter,avx-mid-server-platform --force
   ```

3. If GUI login fails:

   a. Restart all pods:

   ```
   kubectl delete pods -A --all --force
   ```

   b. Allow the pods to restart and verify that they are running.

   c. When all pods are running, log in again.

What to do next:

Complete the following steps after the pluging upgrade ONLY if they have not been performed previously.

- Remove/delete all traces MongoShardDB setup, refer to the section MongoShardDB Cleanup.
- Remove/delete all traces of vault setup, refer to the section Vault/Consul Cleanup.

Troubleshooting for Upgrade Errors

Copyright © 2025 AppViewX Inc. All Rights Reserved.

UPGRADE FAILED: another operation (install/upgrade/rollback) is in progress

While performing the upgrade (`./appviewxctl upgrade --config .appviewxctl.yaml`), if the system encounters the error of UPGRADE FAILED: another operation (install/upgrade/rollback) is in progress as shown below.

```
[appviewx@ip-10-66-86-132 Hudson_FP2_Feb21_167]$ ./appviewxctl upgrade --config .appviewxctl.yaml
[Init] Using log file at [/home/appviewx/Hudson_FP2_Feb21_167/appviewxctl-4556929.log] to dump logs
[Init] Initialise persistent flag config
[Init] Using config file
[Upgrade] Upgrade of the AVX application started...
[Upgrade] Updating enabled plugins dependency and configurations
[Upgrade] updated dependency [avx_app]
[Upgrade] Starting Appviewx application chart [avx-app] upgrade
[Upgrade] failed to run upgrade with error: exit status 1: Error: UPGRADE FAILED: another operation
(install/upgrade/rollback) is in progress
```

Perform the following steps:

1. Fetch the deployment history of **avx-app helm** deployment using command mentioned below.

   ```
   helm history avx-app -n avx
   ```

   ```
   [appviewx@ip-10-66-86-132 Hudson_FP2_Feb21_167]$ helm ls -n avx
   NAME                        NAMESPACE       REVISION        UPDATED                                 STATUS          CHART                                   APP VERSION
   avx-backup                  avx             1               2024-02-21 06:25:44.072256062 +0000 UTC deployed        mongo-backup-1.0.0                      1.0.0
   avx-mongo                   avx             1               2024-02-21 06:15:24.525684044 +0000 UTC deployed        mongodb-1.0.0                           1.0.0
   avx-mongo-seed              avx             1               2024-02-21 06:24:05.867497488 +0000 UTC deployed        mongodbseed-1.0.0                       1.0.0
   avx-pre-req                 avx             1               2024-02-21 06:15:12.370829011 +0000 UTC deployed        avx-pre-req-1.0.0                       1.0.0
   avx-third-party-monitoring  avx             1               2024-02-21 06:25:41.292879916 +0000 UTC deployed        avx-third-party-monitoring-1.0.0        1.0.0
   consul                      avx             1               2024-02-21 06:18:31.951654843 +0000 UTC deployed        consul-1.1.0                            1.16.3
   redis                       avx             1               2024-02-21 06:22:28.447488345 +0000 UTC deployed        redis-1.0.0                             1.0.0
   vault                       avx             1               2024-02-21 06:21:14.012364906 +0000 UTC deployed        vault-0.11.0                            1.14.7
   [appviewx@ip-10-66-86-132 Hudson_FP2_Feb21_167]$ helm history avx-app -n avx
   REVISION        UPDATED                         STATUS          CHART           APP VERSION     DESCRIPTION
   1               Wed Feb 21 06:41:55 2024        superseded      avx-1.0.0       1.0.0           Install complete
   2               Mon Feb 26 06:20:30 2024        deployed        avx-1.0.0       1.0.0           Upgrade complete
   3               Mon Feb 26 09:47:59 2024        failed          avx-1.0.0       1.0.0           Upgrade "avx-app" failed: context canceled
   4               Mon Feb 26 09:48:15 2024        pending-upgrade avx-1.0.0       1.0.0           Preparing upgrade
   ```

2. Find the most recent revision of avx-app which is in deployed state.

   In this case Revision 2 is the most recent upgrade.

3. Run the rollback command to retrieve the deployment from failed state.

   ```
   helm rollback avx-app <REVISION NUMBER> -n avx
   ```

   Example:

   ```
   helm rollback avx-app 2 -n avx
   ```

   ```
   [appviewx@ip-10-66-86-132 Hudson_FP2_Feb21_167]$ helm rollback avx-app 2 -n avx
   W0227 06:33:06.194029    8645 warnings.go:70] outlier detection consecutive errors is deprecated, use consecutiveGatewayErrors or consecutive5xxErrors instead
   Rollback was a success! Happy Helming!
   [appviewx@ip-10-66-86-132 Hudson_FP2_Feb21_167]$ helm history avx-app -n avx
   REVISION        UPDATED                         STATUS          CHART           APP VERSION     DESCRIPTION
   1               Wed Feb 21 06:41:55 2024        superseded      avx-1.0.0       1.0.0           Install complete
   2               Mon Feb 26 06:20:30 2024        superseded      avx-1.0.0       1.0.0           Upgrade complete
   3               Mon Feb 26 09:47:59 2024        failed          avx-1.0.0       1.0.0           Upgrade "avx-app" failed: context canceled
   4               Mon Feb 26 09:48:15 2024        pending-upgrade avx-1.0.0       1.0.0           Preparing upgrade
   5               Tue Feb 27 06:33:01 2024        deployed        avx-1.0.0       1.0.0           Rollback to 2
   ```

4. Re-run the `helm history avx-app -n avx` command to verify the rollback is complete.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Failure Issue during Upgrade

1. Go to the section Cleanup of OpenBao and follow the steps provided.
2. When the cleanup is complete, install in the following order:

```
#Install OpenBao
./appviewxctl --config .appviewxctl.yaml installOpenbao

# Restore Vault
./vault_restore_openbao_migration.sh -p <vault backup file full path>

# Restore Mongo
./mongo_restore.sh <mongo backup file full path>

# Upgrade Plugins
./appviewxctl upgrade --config .appviewxctl.yaml
```

Login Issue after Upgrade

If a login issue occurs after the upgrade:

1. Perform he vault cleanup, refer the section Vault/Consul Cleanup.
2. Reinstall OpenBao.
3. Restore the Vault data.
4. Now, repeat the application upgrade process.

MongoShardDB Cleanup

After completing the plugins upgrade and validating the functionality, it is necessary to clean up (delete) all traces of the MongoDB shard database setup. Follow the steps outlined below:

1. Remove the Mongo ShardDB using the command below.

```
helm uninstall avx-mongo -n avx
```

2. Delete the StatefulSet resource by executing the command below.

```
kubectl delete sts mongo-shareddb mongo-configdb mongo-routerdb -n avx --force
```

3. Delete the specified Kubernetes services (service objects) in the cluster by executing the command below.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
kubectl delete svc mongo-shareddb-service mongo-configdb-service mongo-routerdb-service -n avx --force
```

4. Delete the MongoShardDB related **PVCs** by executing the commands below.

```
kubectl delete pvc mongo-configdb-persistent-storage-claim-mongo-configdb-0 mongo-configdb-persistent-storage-claim-mongo-configdb-1
 mongo-configdb-persistent-storage-claim-mongo-configdb-2 -n avx --force
```

```
kubectl delete pvc mongo-shareddb-persistent-storage-claim-mongo-shareddb-0 mongo-shareddb-persistent-storage-claim-mongo-shareddb-1
 mongo-shareddb-persistent-storage-claim-mongo-shareddb-2 -n avx --force
```

5. Delete the MongoShardDB related kubernetes secrets using the commands below.

```
kubectl delete secret mongo-key -n avx --force
```

Vault/Consul Cleanup

After completing the plugins upgrade or if you have recently migrated/installed OpenBao, and validating the functionality, it is necessary to clean up (delete) all traces of the Vault/Consul setup. Follow the steps outlined below:

> ⚠ **Warning:** Do not perform these steps if the migration from Vault to OpenBao has not been completed

1. Remove the vault/consul and related pods using the commands below.

```
helm uninstall vault -n avx
```

```
helm uninstall consul -n avx
```

```
kubectl delete pods vault-0 vault-1 vault-2 -n avx --force
```

```
kubectl delete pods consul-consul-server-0 consul-consul-server-1 consul-consul-server-2 -n avx --force
```

2. Delete the specified Kubernetes services (service objects) in the cluster by executing the command below.

```
kubectl delete secret vault-key-0 vault-key-1 vault-key-2 -n avx --force
```

3. Delete the vault/consul related **PVCs** by executing the commands below.

```
kubectl patch pvc data-avx-consul-consul-server-0 -n avx -p '{"metadata":{"finalizers":null}}'
kubectl patch pvc data-avx-consul-consul-server-1 -n avx -p '{"metadata":{"finalizers":null}}'
kubectl patch pvc data-avx-consul-consul-server-2 -n avx -p '{"metadata":{"finalizers":null}}'
kubectl delete pvc data-avx-consul-consul-server-0 data-avx-consul-consul-server-1 data-avx-consul-consul-server-2 -n avx --force
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

4. Delete the configmap using the commands below.

```
kubectl delete cm avx-vault-configmap -n avx
```

OpenBAO Cleanup

If any issues are encountered while running the `installOpenbao` command, ensure that all traces of previous attempts are cleaned up before retrying by following these steps:

1. Remove OpenBao and releated pods using the commands below.

```
helm uninstall openbao -n avx
```

```
kubectl delete pods openbao-0 openbao-1 openbao-2 -n avx --force
```

2. Delete kubernetes secrets using the commands below.

```
kubectl delete secret openbao-key-0 openbao-key-1 openbao-key-2 -n avx --force
```

```
kubectl delete secret bao-key-secretid -n avx --force
```

```
kubectl delete secret bao-key-secretid -n avx-kafka --force
```

```
kubectl delete secret bao-key-secretid --force
```

3. Delete **PVCs** by executing the command:

```
kubectl delete pvc data-openbao-0 data-openbao-1 data-openbao-2 -n avx --force
```

4. Delete **configmap** by executing the command:

```
kubectl delete cm avx-openbao-configmap -n avx --force
```

Mongo Replicaset Cleanup

If any issues are encountered while running the `installMongoRS` command, ensure that all traces of previous attempts are cleaned up before retrying by following these steps:

1. Remove the Mongo Replicaset using the command below.

```
helm uninstall avx-mongo-rs -n avx
```

2. Delete the StatefulSet resource by executing the command below.

```
kubectl delete sts avx-mongo-mongodb -n avx --force
```

3. Delete the specified Kubernetes services (service objects) in the cluster by executing the command below.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
kubectl delete svc avx-mongo-mongodb-headless -n avx --force
```

4. Delete the Mongo Replicaset related **PVCs** by executing the commands below.

```
kubectl delete pvc avx-mongo-mongodb-persistent-storage-claim-avx-mongo-mongodb-0
 avx-mongo-mongodb-persistent-storage-claim-avx-mongo-mongodb-1 avx-mongo-mongodb-persistent-storage-claim-avx-mongo-mongodb-2 -n avx --force
```

5. Delete the Mongo Replicaset related kubernetes secrets using the commands below.

```
kubectl delete secret mongo-key-rs -n avx --force
```

Downloading Images from AppViewX Repository

Prerequisites

1. Get the source image repository credentials from AppViewX.
2. Configure the docker using the command

```
docker login -u ${USERNAME} -p ${PASSWORD} ${DOCKER_REPOSITORY}
```

3. Configure the respective cloud provider CLI (Google cloud) and ensure you have access to push docker images to GCR.

The script for image push and pull is as follows:

```
#!/bin/bash
# Pre requisite
# Get the source image repository credentials from Appviewx and then configure docker using `docker login -u ${USERNAME} -p ${PASSWORD}
 ${DOCKER_REPOSITORY}`
# Configure the respective cloud provider cli (aws/az/gcloud) and make sure that the user has access to push docker images to ACR/ECR/GCR.


appVersion=$1 # App image version. E.g: 24.3.0.0
targetImageRegistry=$2 # Image resgistry name


# Validate required inputs
if [ -z "$appVersion" ] || [ -z "$targetImageRegistry" ];then
{
    echo "Please provide script parametes as ./script.sh <appVersion> <targetImageRegistry>"
    exit
}
fi
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
# Set the registry login
if echo $targetImageRegistry | grep -iq "amazonaws";then
{
    registryProvider="ecr"
    region=$(echo $targetImageRegistry | cut -d "." -f4)
    aws ecr get-login-password --region $region | docker login --username AWS --password-stdin $targetImageRegistry
}
elif echo $targetImageRegistry | grep -iq "azurecr";then
{
    registryProvider="acr"
    az acr login -n $targetImageRegistry
}
elif echo $targetImageRegistry | grep -iq "gcr";then
{
    registryProvider="gcr"
    gcloud auth print-access-token | docker login -u oauth2accesstoken \
    --password-stdin $(echo $targetImageRegistry | cut -d '/' -f2)
}
else
{
    echo "Unknown registry provider"
    exit 2
}
fi

# Image tag mappings
imageTags='[
   {
      "imageName": "avx-cloud-managedservice-mks",
      "tagVersion": "appVersion",
      "upload": true
   },
   {
      "imageName": "avx-cloud-web",
      "tagVersion": "appVersion",
      "upload": true
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
},
{
    "imageName": "avx-cloud-gateway",
    "tagVersion": "appVersion",
    "upload": true
},
{
    "imageName": "avx-platform-report-generator",
    "tagVersion": "appVersion",
    "upload": true
},
{
    "imageName": "mongo-init",
    "tagVersion": "appVersion",
    "upload": true
},
{
    "imageName": "avx-cloud-mongoseed",
    "tagVersion": "appVersion",
    "upload": true
},
{
    "imageName": "avx-python-sandbox",
    "tagVersion": "appVersion",
    "upload": true
},
{
    "imageName": "avx-mid-server-base",
    "tagVersion": "appVersion",
    "upload": true
},
{
    "imageName": "alpine",
    "tagVersion": "3.22.0",
    "upload": true
},
```


Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
{
    "imageName": "pilot",
    "tagVersion": "1.26.1",
    "upload": true
},
{
    "imageName": "proxyv2",
    "tagVersion": "1.26.1",
    "upload": true
},
{
    "imageName": "openbao",
    "tagVersion": "2.2.2",
    "upload": true
},
{
    "imageName": "redis",
    "tagVersion": "8.0.2",
    "upload": true
},
{
    "imageName": "kafka",
    "tagVersion": "0.45.0-kafka-3.9.0",
    "upload": true
},
{
    "imageName": "operator",
    "tagVersion": "0.45.0",
    "upload": true
},
{
    "imageName": "kube-metrics-adapter",
    "tagVersion": "v0.2.3",
    "upload": true
},
{
    "imageName": "kibana",
```


Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
    "tagVersion": "9.0.2",
    "upload": true
},
{
    "imageName": "grafana",
    "tagVersion": "11.6.2",
    "upload": true
},
{
    "imageName": "filebeat",
    "tagVersion": "9.0.2",
    "upload": true
},
{
    "imageName": "logstash",
    "tagVersion": "9.0.2",
    "upload": true
},
{
    "imageName": "logstash-syslog",
    "tagVersion": "9.0.2",
    "upload": true
},
{
    "imageName": "elasticsearch",
    "tagVersion": "9.0.2",
    "upload": true
},
{
    "imageName": "elasticsearch-insight",
    "tagVersion": "9.0.2",
    "upload": true
},
{
    "imageName": "prometheus",
    "tagVersion": "v3.5.0",
    "upload": true
```


Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
  },
  {
   "imageName" : "backup-utility-image",
   "tagVersion": "v2024.3.0",
      "upload": true
  },
  {
   "imageName" : "alertmanager",
   "tagVersion": "v0.28.1",
      "upload": true
  },
  {
   "imageName" : "node-exporter",
   "tagVersion": "v1.9.1",
      "upload": true
  },
  {
   "imageName" : "metrics-server",
   "tagVersion": "v0.7.2",
      "upload": true
  },
  {
   "imageName" : "kube-state-metrics",
   "tagVersion": "v2.15.0",
      "upload": true
  },
  {
   "imageName" : "redis_exporter",
   "tagVersion": "v1.73.0",
      "upload": true
  }
]'

for row in $(echo "${imageTags}" | jq -r '.[] | @base64'); do
  _jq() {
   echo ${row} | base64 --decode | jq -r ${1}
  }
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
  imageUpload=$(_jq '.upload')
  tagVersion=$(_jq '.tagVersion')
  if [ $imageUpload == "true" ];then
  {
     if [ "${tagVersion}" == "appVersion" ];then
     {
        docker pull images.appviewx.com/appviewx/$(_jq '.imageName'):$appVersion
        docker tag images.appviewx.com/appviewx/$(_jq '.imageName'):$appVersion $targetImageRegistry/appviewx/$(_jq '.imageName'):$appVersion
        docker push $targetImageRegistry/appviewx/$(_jq '.imageName'):$appVersion
     }
     else
     {
        docker pull images.appviewx.com/appviewx/$(_jq '.imageName'):$(_jq '.tagVersion')
        docker tag images.appviewx.com/appviewx/$(_jq '.imageName'):$(_jq '.tagVersion') $targetImageRegistry/appviewx/$(_jq '.imageName'):$(_jq
'.tagVersion')
        docker push $targetImageRegistry/appviewx/$(_jq '.imageName'):$(_jq '.tagVersion')
     }
     fi
  }
  fi
done
```

Execute the Image Push-Pull Script

To execute the above image push-pull script, run the command

```
./avx_image_pull_push.sh <image-tag> <targetImageRegistry>
```

Note: Contact AppViewX Support for the <**image-tag**>.

Utility Tools and Functions

Updating Ingress Host in the Config Map

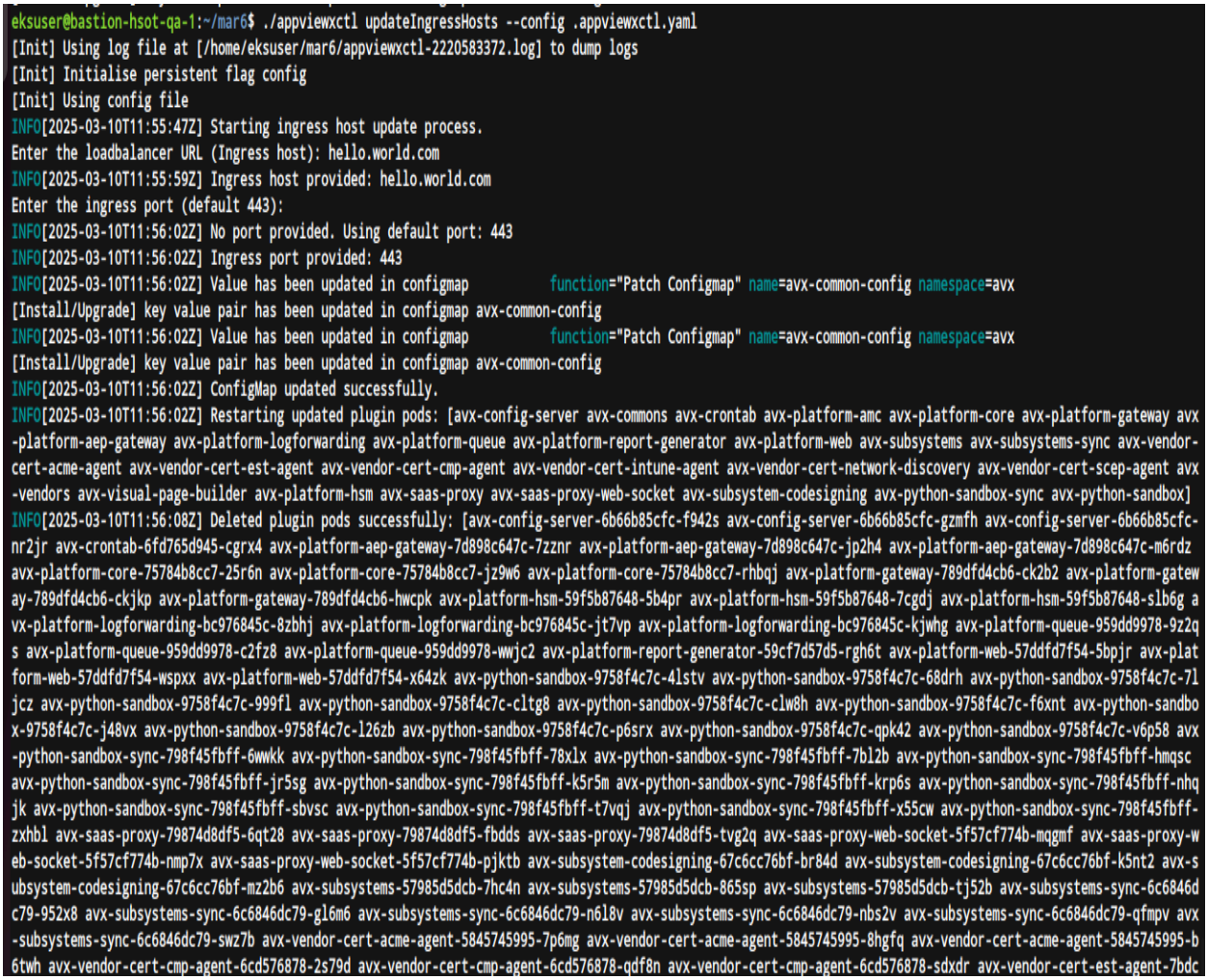
Customers can now update their own ingress host directly in the config map using the updateIngressHosts utility below. If they are using an external load balancer and wish to modify the ingress, it can be done using the following command:

```
./appviewxctl updateIngressHosts --config .appviewxctl.yaml
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

On executing the above utility function, you will be prompted with the following:

- *Enter the loadbalancer URL (Ingress Host):*
- *Enter the ingress port (default 443):* - If you're using the default port 443, simply press Enter. Otherwise, enter the desired port number and then press Enter.



Note: After config map is updated, all the plug-in pods will get restarted automatically.

Updating Istio Certificates in the tls-credential secret

Customers can now upload/update their own istio certificates using the updateIngressCerts utility provided below:

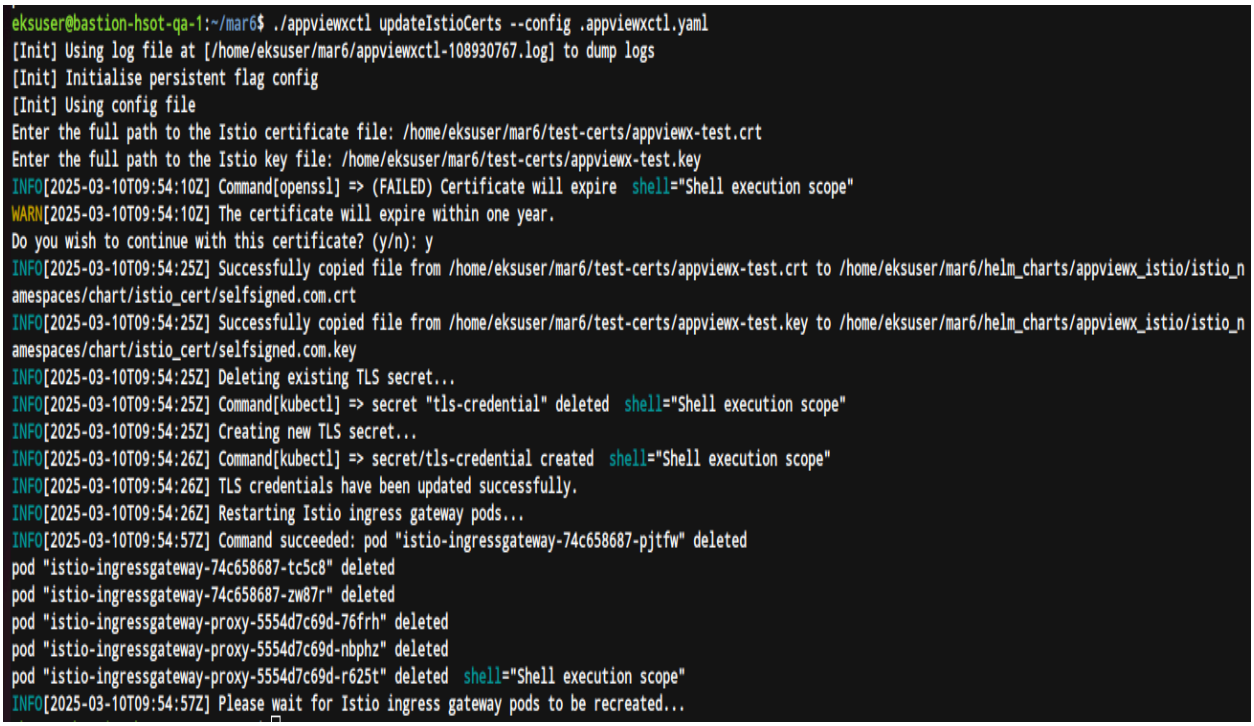
```
./appviewxctl updateIngressCerts --config .appviewxctl.yaml
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

On executing the above utility function, you will be prompted with the following:

- *Enter the full path to the Istio certificate file:* - Enter the directory path including the certificate file.
- *Enter the full path to the Istio key file:* - Enter the directory path including the certificate key.

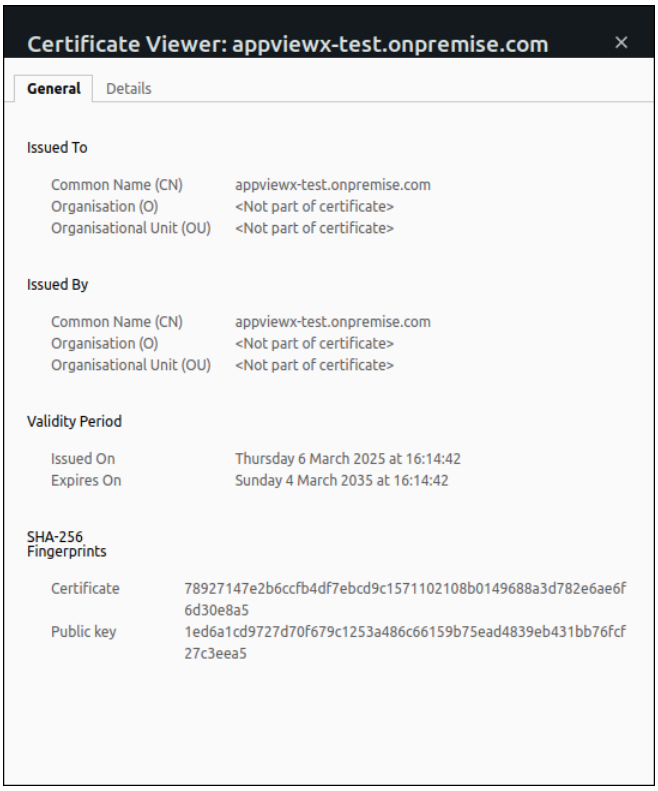
- If the entered certificate is due to expire within one year then you will prompted with *Do you wish to continue with this certificate? (y/n):*



The entered certificate file and key are placed in the **istio_certs** folder. The existing tls-credential secret is deleted and a new one is created using the entered certificate and key. The exiting istio ingress gateway and the istio ingress gateway pods are deleted. The istio ingress gateway pods are then recreated and once they are up and running the utility program ends.

The certificates can be verified on the browser as shown below.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

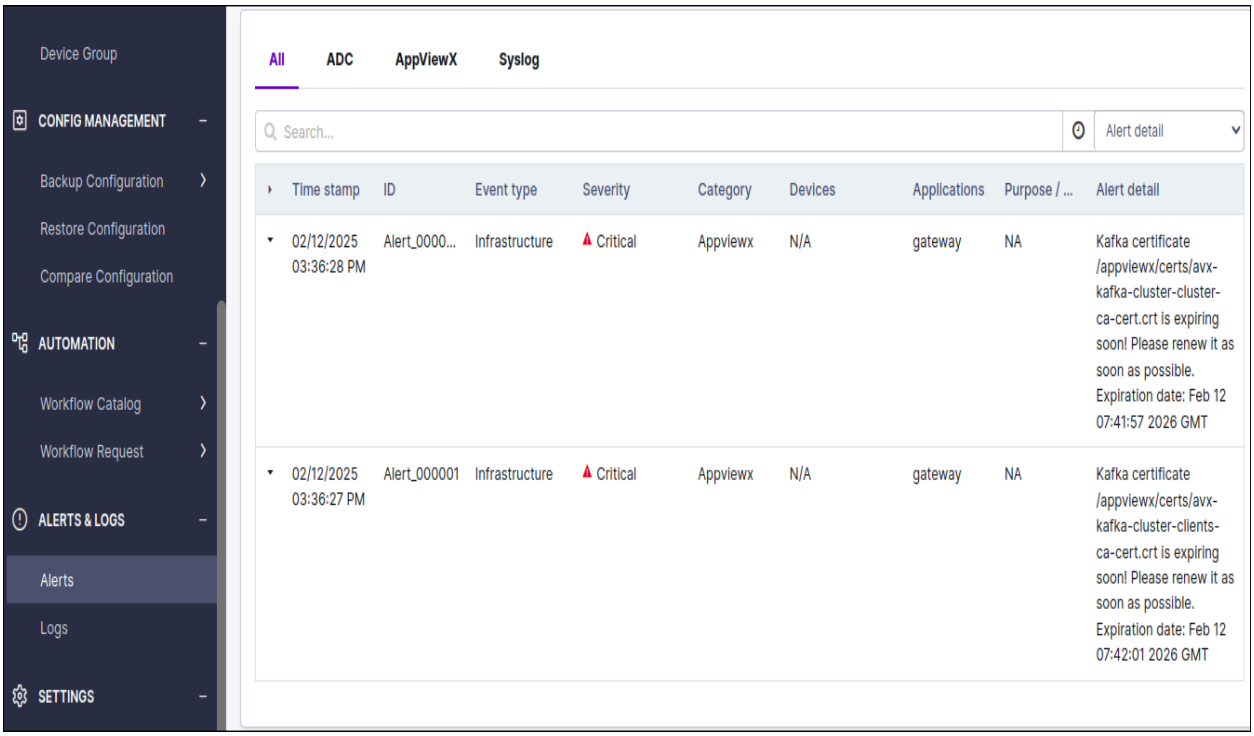


Kafka Cert Utility

If the Kafka certificates are nearing expiry, a weekly cron job is available that sends alerts to the gateway if they expire within the next 3 months. The cron job alert details can be viewed in the product by accessing the product menu, **ALERTS & LOGS** > **Alerts**.

Note: To make the alerts section visible on the gateway, first enable the following checkbox: **Identity → Role → Select appropriate role → Authorized Functions → Platform → Observe and Explore - Alerts (Checkbox).**

Copyright © 2025 AppViewX Inc. All Rights Reserved.



Utility function to renew kafka certs: Users will be able to renew kafka certs with a new utility, by executing the single command below.

```
./appviewxctl renewKafkaCerts --config .appviewxctl.yaml
```

Log Collection

The log collection script is designed to collect logs in a managed kubernetes setup. It is compatible with all mk8s environments, regardless of the cloud provider, and also retrieves logs from the backup folder.

Prerequisites: The bastion host should have kubectl and have access to the compute cluster to collect the logs.

1. Login to bastion host.
2. Download the helper tarball from the release portal.
3. Execute the following command:

```
./mk8s_collect_logs.sh
```

> **Note:** The script can be run anywhere in the bastion host.

4. Fill the interactive questions as follows:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

a. ***Do you want to collect logs for all plugins?***

- Answer **no** to collect logs for specific plugins. You will then be prompted to enter the names of the desired plugins.
- Answer **yes** to collect logs for all enabled plugins.

> **Note:** To view all the enabled plugins in the setup, run:
>
> ```
> ./mk8s_collect_logs.sh -operation get-enabled-plugins
> ```

b. ***Do you want to collect mongo logs?***

- Answer **yes** to include MongoDB logs in the collection.
- Answer **no** to exclude MongoDB logs.

c. ***Enter the directory to save logs:***

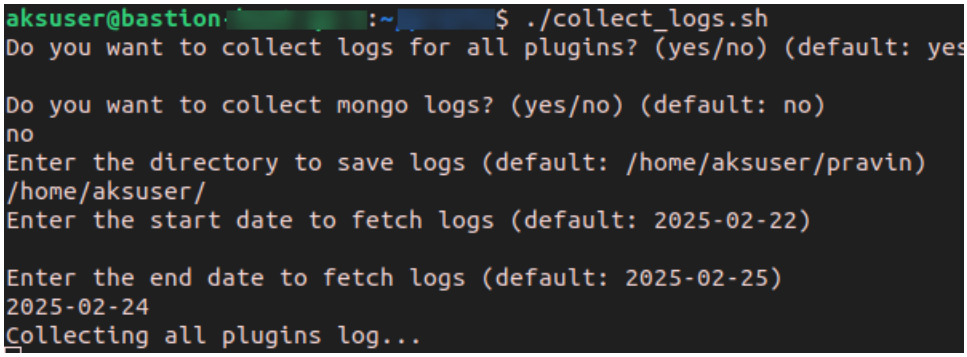
- The collected logs will be saved in the provided directory. If no directory is specified, the default path will be the directory where the script is executed.

d. ***Enter the start date to fetch logs:***

- Enter the date from which you'd like to begin collecting logs.

e. ***Enter the end date to fetch logs:***

- Enter the date until which you want to collect logs.



Alternative options: Passing Parameters via Command Line

Alternatively, you can provide the inputs as parameters instead of entering them interactively. To view the available parameters, run the command below.

```
./mk8s_collect_logs.sh --help
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Uninstall and Cleanup

The process of uninstalling requires one to navigate to the installer directory and execute the following command

```
./appviewxctl uninstall --config .appviewxctl.yaml
```

The following messages are displayed after the uninstall command is executed successfully.

```
./appviewxctl uninstall --config .appviewxctl.yaml

[Init] Using log file at [/avx/appviewxctl-3196327299.log] to dump logs
[Init] Initialise persistent flag config
[Init] Using config file
[Uninstall] Uninstalling appviewx application
[Uninstall] Uninstalling Appviewx application helm chart
[Uninstall] Uninstalling application backup helm chart
[Uninstall] Uninstalling Infra application helm chart
[Uninstall] Uninstalling Third party application helm chart
[Uninstall] Uninstalling IstioOperator from the cluster
[Uninstall] Uninstalling PVCs from the avx namespace
[Uninstall] Uninstalling Pre-requisite helm chart
[Uninstall] Uninstalling Appviewx installed namespaces
[Uninstall] Successfully uninstalled appviewx application and all the related resources
```



Note: In the Managed K8s environments removal of PVCs do not occur at times as it may require patching PVCs first before deletion. This may cause certain error messages to display, indicating that PVC has changed. In case of such an error occurs re-run the above command to solve the issue and uninstall the application.

Sometimes the namespaces take a longer time to be removed. Hence, post installation, check if namespaces are in the terminating state (use the command: **kubectl get namespace**). If any namespace is in the terminating state, manually remove the namespaces by executing the commands below:

```
kubectl get namespace "istio-operator" -o json | tr -d "\n" | sed "s/\"finalizers\": \[[^]]\+\]/\"finalizers\": []/" | kubectl replace
 --raw /api/v1/namespaces/istio-operator/finalize -f - 2>/dev/null
```

```
kubectl get namespace "istio-system" -o json | tr -d "\n" | sed "s/\"finalizers\": \[[^]]\+\]/\"finalizers\": []/" | kubectl replace
 --raw /api/v1/namespaces/istio-system/finalize -f - 2>/dev/null
```

```
kubectl get namespace "avx" -o json | tr -d "\n" | sed "s/\"finalizers\": \[[^]]\+\]/\"finalizers\": []/" | kubectl replace --raw /api/v1/namespaces/avx/finalize -f -
 2>/dev/null
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
kubectl get namespace "avx-kafka" -o json | tr -d "\n" | sed "s/\"finalizers\": \[[^]]\+\]/\"finalizers\": []/" | kubectl replace --raw /api/v1/namespaces/avx-kafka/finalize -f - 2>/dev/null
```

```
kubectl delete ns istio-operator --force 2>/dev/null
```

```
kubectl delete ns istio-system --force 2>/dev/null
```

```
kubectl delete ns avx --force 2>/dev/null
```

```
kubectl delete ns avx-kafka --force 2>/dev/null
```

Troubleshooting

Error	Resolution
403: Access Forbidden - License not available. This error occurs when only the managed kubernetes cluster is restarted every morning.	In case of managed kubernetes, when the cluster is restarted it is recommend to restart all pods.

ManagedK8s Kubernetes Upgrade: Guidelines for Cluster Admins

This section provides guidelines for cluster administrators to understand the Managed Kubernetes (managedk8s) upgrade process, identify potential points of failure, and follow recommended mitigation steps.

- Understanding the Managedk8s Kubernets Upgrade Process
- Potential Failure Scenarios and their Mitigations

Understanding the Managedk8s Kubernets Upgrade Process

The cluster architecture comprises a control plane and associated node pools.

1. The control plane upgrade is managed entirely by the cloud provider and performed in the backend services. The control pane is upgraded first followed by the node pools.
2. The node pool upgrade follows a rollout approach. For each node in the pool, a new node is provisioned with the new Kubernetes version.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

3. The existing node is then drained, allowing all active pods to migrate seamlessly to the newly provisioned node.
4. This process is repeated one node at a time until all nodes in the node pool are upgraded.
5. Once all node pools have been upgraded, the cluster upgrade is considered complete.

Potential Failure Scenarios and their Mitigations

In Step 3 outlined above, when the nodes undergo a draining process, if any deployed entities prevent the node from being drained successfully, it may result in timeout failures.

AppViewX deployed components that can cause these failures and their mitigations are as follows:

PDB (Pod Disruption Budget)

Failure Cause: AppViewX infrastructure components are deployed with Pod Disruption Budgets (PDBs) to ensure that a minimum number of replicas for critical components remain available, thereby maintaining application availability. During node draining process, these PDBs may block the process if the required number of replicas are not yet online, as they are configured with a strict policy and have “allowed disruption” of 1.. The default PDBs are as follows:

Namespace	Name	Min Available	Max Unavailable	Allowed Disruption
avx-kafka	avx-kafka-cluster-kafka	2	NA	1
avx-kafka	avx-kafka-cluster-zookeeper	2	NA	1
avx	consul-consul-server	NA	1	1
avx	vault / openbao	NA	1	1

Note: In the table above, based on the AppViewX version, some PDBs may not be available (e.g.: consul). Ignore if not available.

Mitigation: To mitigate this potential upgrade issue, we can temporarily set the **Min Available to 1** or **Max Unavailable to 2**. This will allow disruption of 2, allowing the draining process to have some extra room to reach completion and not get timed out waiting for enough replicas to be online. Execute the command:

```
kubectl edit pdb <pdb-name> -n <namespace>
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Post this, edit the Min Available or Max Unavailable with the prescribed values.

Note: Revert the values once the cluster is upgraded.

Pods with istio Sidecar Termination Seconds Configured

Failure Cause: AppViewX deploys specific pods configured to keep the sidecar container alive for up to 1.5 hours after pod termination. This setup ensures that certain jobs complete successfully, even when pods are terminated due to external events like scaling. However, this behavior can delay the node draining process and may lead to upgrade timeouts.

Mitigation: To mitigate this, keep track of pods that are in a terminating state and forcefully kill it.

- To check the terminating pods, execute the command

```
kubectl get pods -n avx | grep -i terminating
```

- To delete the terminating pods, execute the command

```
kubectl delete pod $(kubectl get pods -n avx | grep -i terminating | awk '{print $1}') -n avx --force
```

Monitor the upgrade process, when a node is being drained i.e. the node is in scheduling disabled state, run the above commands to find and kill the terminating pods forcefully.

Alternatively, run the following snippet in a bash script while the upgrade is in process and this will periodically take care of the terminating pods. Stop this script after upgrade completion.

```
#!/bin/bash


while true; do
 pods=$(kubectl get pods -n avx | grep -i Terminating | awk '{print $1}')
 if [ -n "$pods" ]; then
  for pod in $pods; do
   echo "Force deleting pod: $pod"
   kubectl delete pod "$pod" -n avx --force --grace-period=0
  done
 fi
 sleep 30
done
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Chapter 4: OpenShift Deployment with ArgoCD

- Overview
- Prerequisites
- Creating Image Pull Secret
- Installing avx-infra
- Installing avx-app
- Deleting the avx-app
- Upgrading the avx-app
- Migrating Data from On-Premise to OCP

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Chapter 5: Overview

ArgoCD is a GitOps-based tool designed to facilitate the deployment and management of applications through Git. It requires the following inputs:

1. **Cluster Details**: Information about the target Kubernetes cluster.
2. **Git Repository**: A repository containing Helm charts for the applications to be deployed.

ArgoCD continuously monitors the Helm charts in the Git repository and detects any changes. Upon identifying modifications, it synchronizes the application with the Kubernetes cluster to apply the updates automatically.

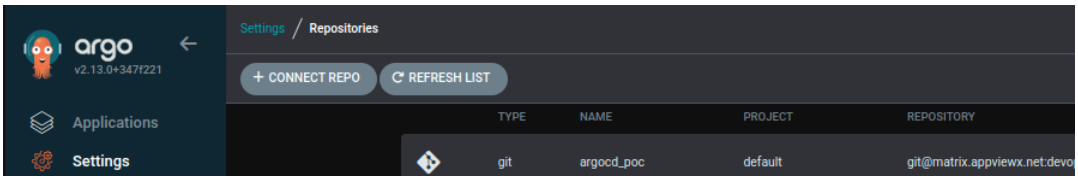
This document serves as a guide for deploying the AppViewX Managed Kubernetes application using ArgoCD.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

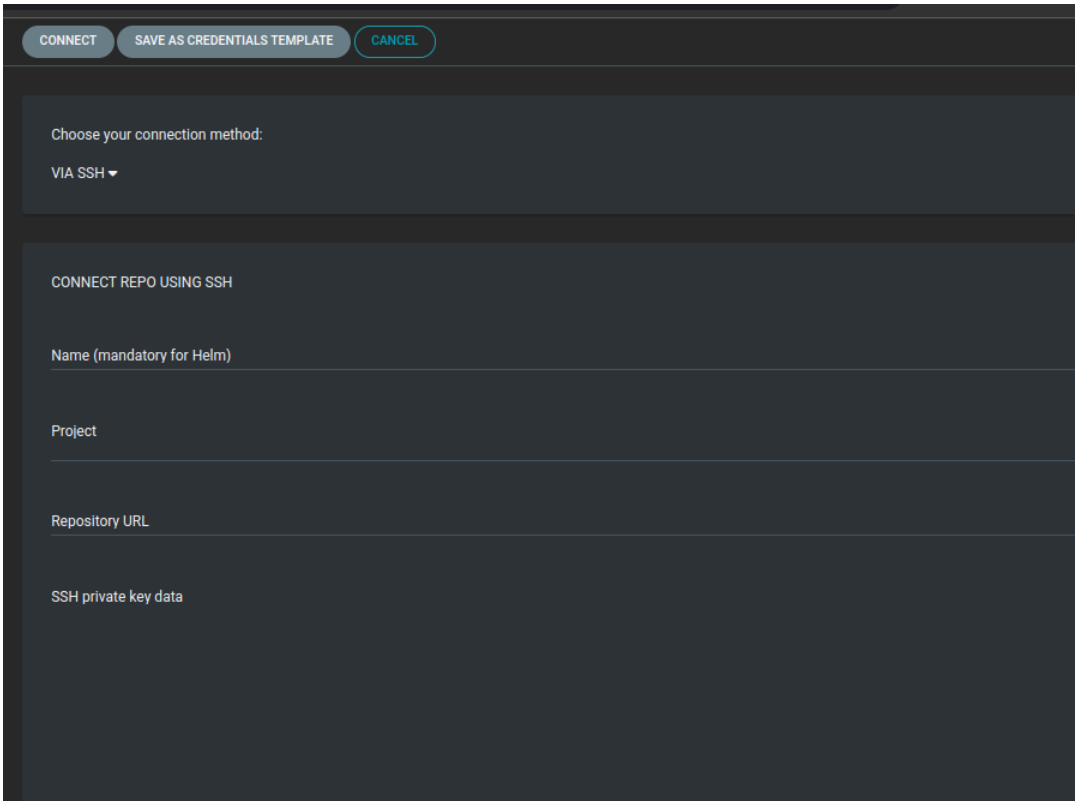
Chapter 6: Prerequisites

Adding the Git Repository to ArgoCD

1. Log in to ArgoCD UI, go to **Settings** > **Repositories**, and then click **Connect Repo**.



2. Fill in the required details; there are multiple ways to connect:
   - Via SSH (Private Key needed)
   - Via HTTPS
   - Via Github App

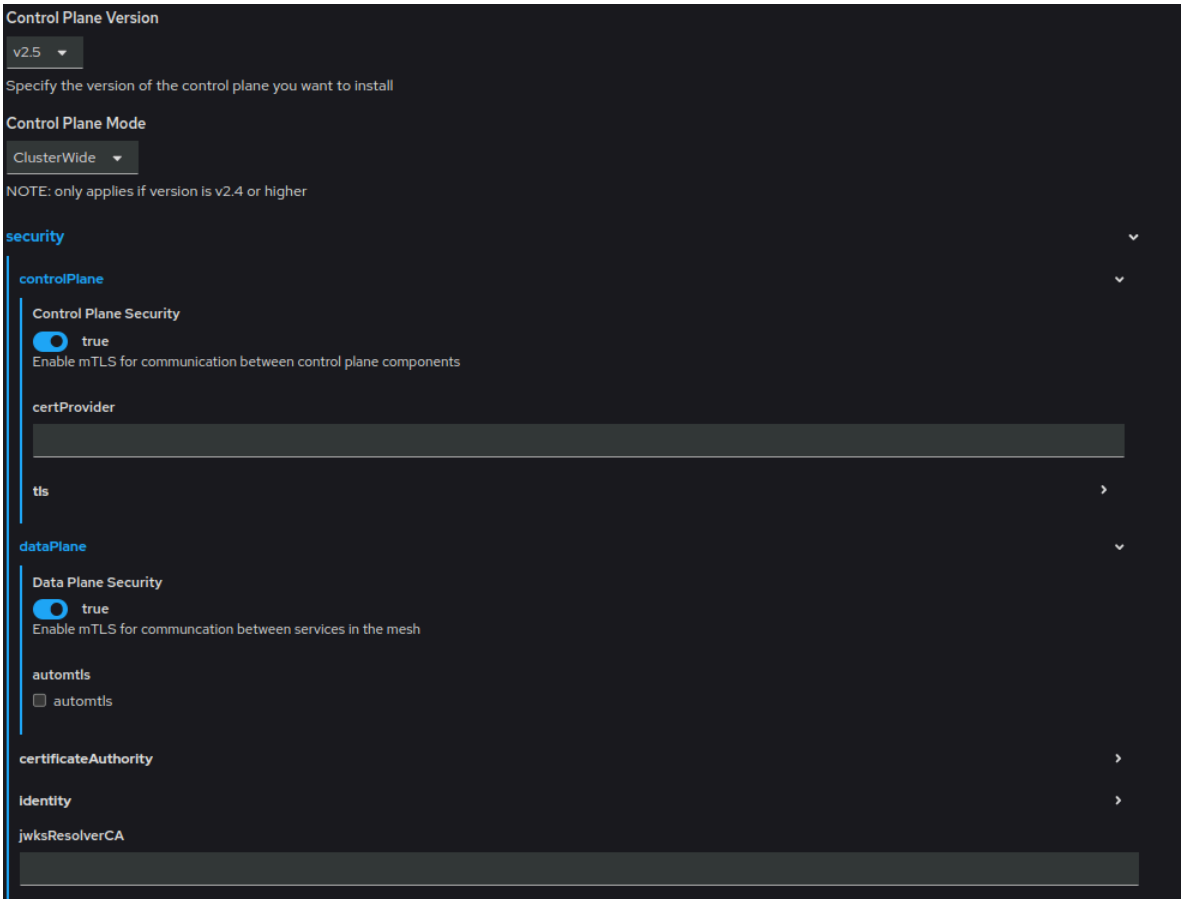


Configuring Istio-Ingress Gateway (Prerequisite)

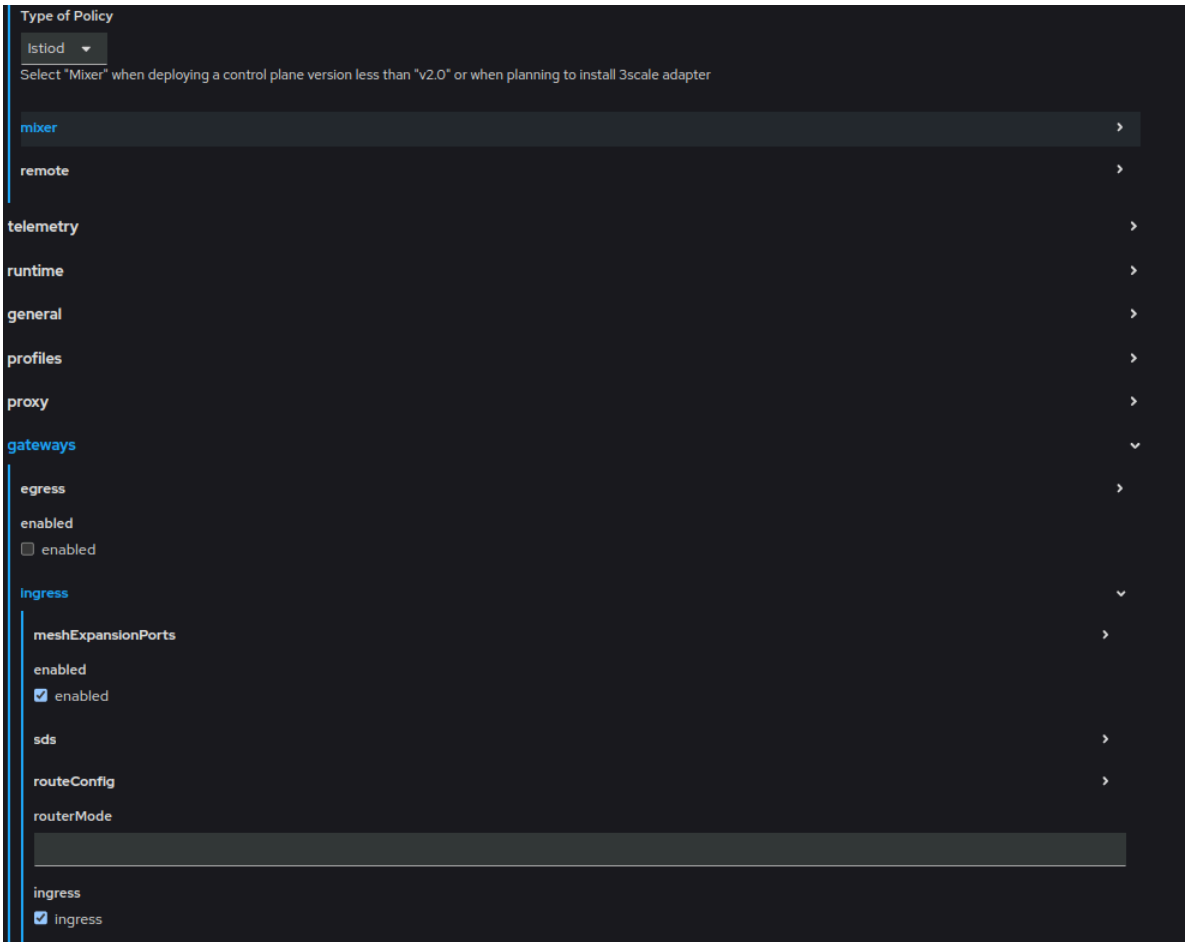
The following section of the document outlines the steps to install the OpenShift Service Mesh in an OCP cluster. It also provides an overview of concepts such as routes and the implementation of the Istio Ingress Gateway Proxy in OpenShift.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

1. Login into OCP cluster UI.
2. Navigate to the **Operator** section.
3. As a prerequisite install **Jaeger** and **Kiali** operators.
4. Install **OpenShift Service Mesh** operator.
5. Once the Red Hat Openshift Service Mesh operator is up and running, go to **Operator Details**.
6. The operator will have custom resource definitions (CRDs).
   - ServiceMeshControlPlane
   - Istio Service Mesh Member
   - Istio Service Mesh Member Roll
7. Create the instance of ServiceMeshControlPlane using the configuration mentioned in the screenshot below.

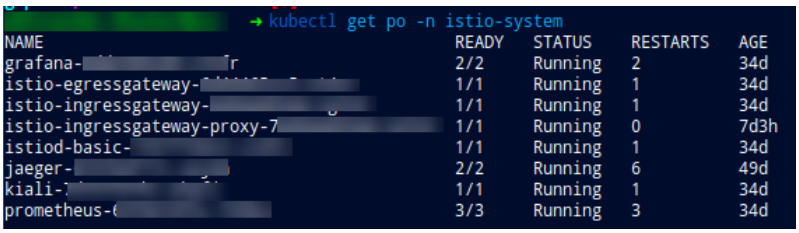


Copyright © 2025 AppViewX Inc. All Rights Reserved.



8. This configuration creates an Istio control plane with the name **basic**, and deploys **istiod** and **istio-ingress-gateway** in the **istio-system** namespace.
   Execute the below command to view the istio pod status:

```
kubectl get po -n istio-system
```



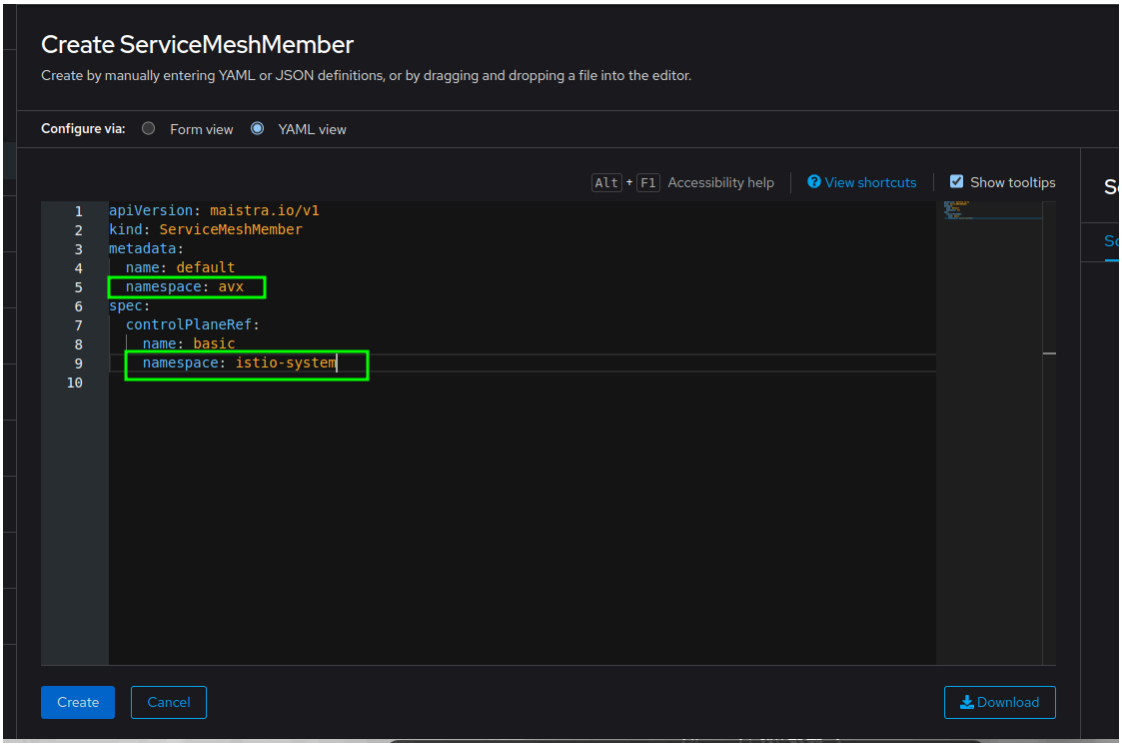
9. To install the **istio-ingressgateway-proxy**, go to **ServiceMeshControlPlane** and edit the existing instance of ServiceMeshControlPlane.
10. In the **spec.gateway** section add the **istio-ingressgateway-proxy** configuration.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
additionalIngress:
 istio-ingressgateway-proxy:
  enabled: true
  service:
   metadata:
    labels:
     app: istio-ingressgateway-proxy
     istio: istio-ingressgateway-proxy
   type: ClusterIP
```

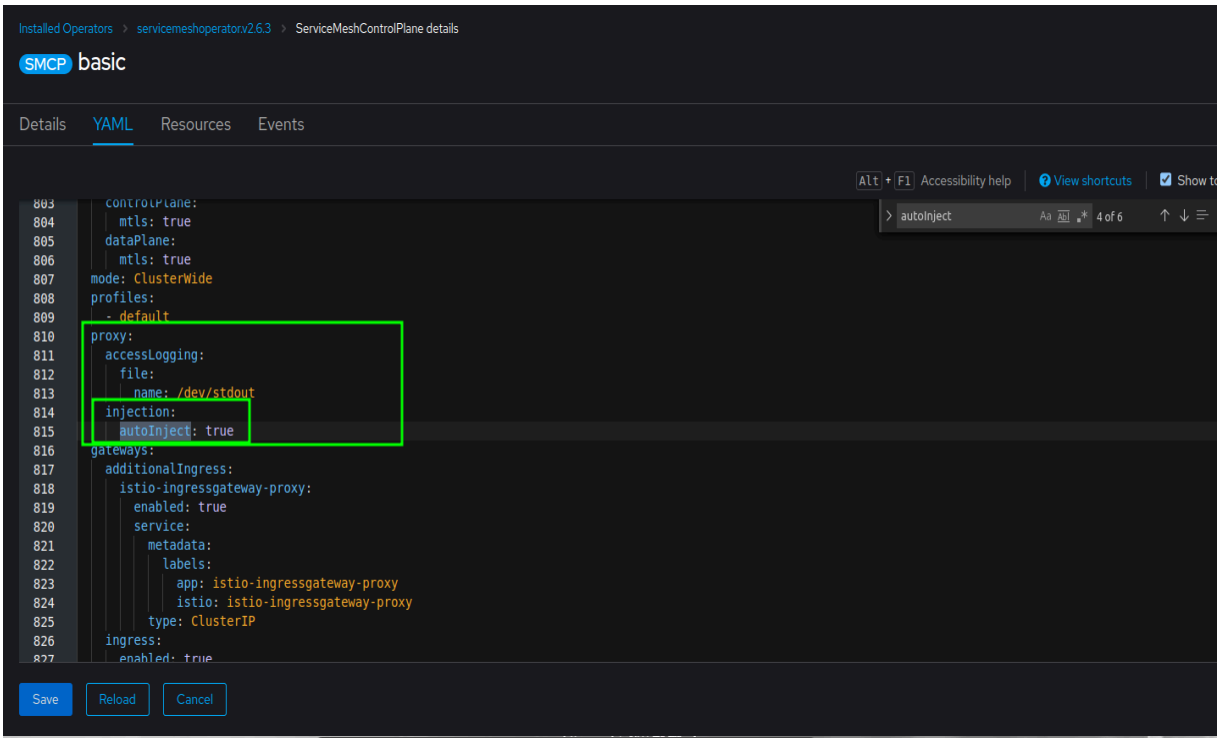
Once the configuration is saved Operator will spin up istio-ingressgateway-proxy of type gateway.

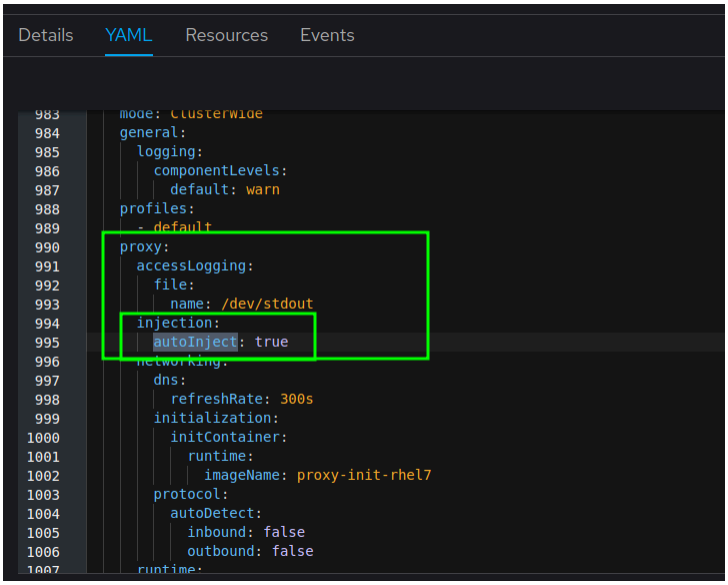
11. To enable the istio injection in avx namespace, create the **Istio Service Mesh Member** and configuration mentioned in the screenshot below.



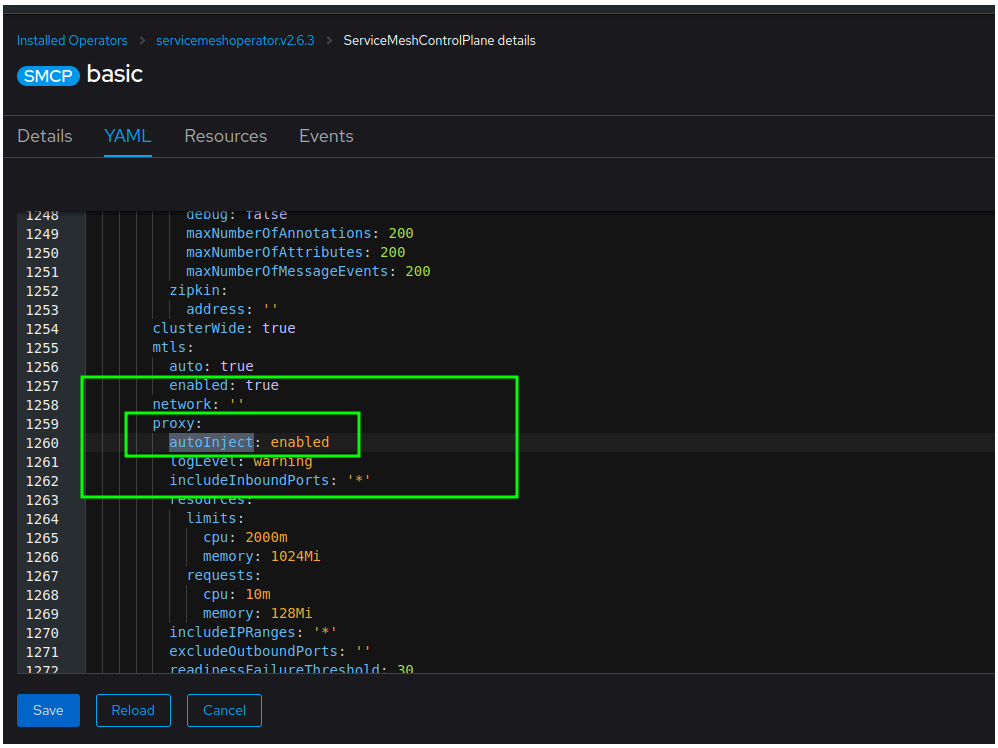
12. In the servicemesh controller yaml, ensure that **autoinject** is enabled as configured in below screenshots.

Copyright © 2025 AppViewX Inc. All Rights Reserved.





Copyright © 2025 AppViewX Inc. All Rights Reserved.



To expose istio-ingressgateway and istio-ingressgateway-proxy routes have to be created. Route for istio-ingressgateway will be created automatically at the time of instance creation of ServiceMeshControlPlane.

To create a route of **istio-ingressgateway-proxy** follow the steps below:

a. Login to OCP dashboard.

b. Navigate to the **Networking** section.

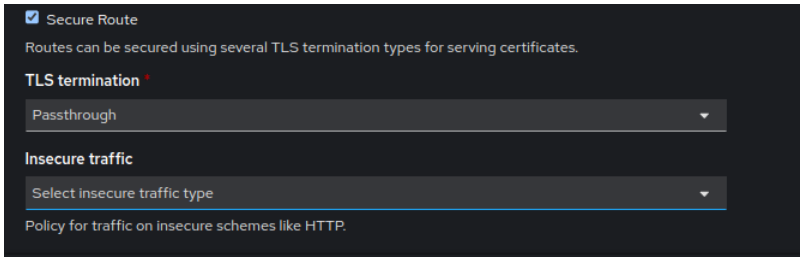
c. In the Networking section, click **Routes**.

d. Change the **Project Name** to *istio-system*.

e. Click the **Create Route** button.

Copyright © 2025 AppViewX Inc. All Rights Reserved.





f. Fill in the mandatory fields.
   i. The service field lists all the services available in the **istio-system** field.
   ii. Select **istio-ingressgateway-proxy** service
   iii. Select target port **443**.
   iv. Enable **Secure Route** and configure TLS termination as pass through.
   v. Once configured, a route will be created for istio-ingressgateway-proxy.

   Reference: OpenShift Documentation: Installing OSSM

Note:

- By default Istio-ingressgateway route will be exposed on 8080, Configure it to use port 443 and use the TLS termination option Passthrough.
- Ensure the above steps are followed before AppViewX installation.

Creating the Ingress Gateway's Certificates

Create the ingress gateway's certificates and place it in the kubernetes secret name **tls-credential** in **istio-system** namespace.

Execute the command:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
kubectl create -n istio-system secret tls tls-credential --key=<absolute path of .key file> --cert=<absolute path of .crt file>
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Chapter 7: Creating Image Pull Secret

This is a mandatory step to be performed before installing the avx-infra.

- Create a pull secret in the **avx**, **avx-kafka**, and **kube-system** namespace that would permit the pods to pull images from the intended image registry.
- The default name of the secret is **avx-image-pull-secret**. If a different name is used then the image pull secrets' name should match the name given in the values.yaml in both avx-infra and avx-app (imagePullSecrets) during installation
- For HA (High Avaliability), 3 zones/datacenters are required. The pods will be distributed across those zones. Ensure that the nodes in the zones have the following label to identify which zone a particular node belongs to:

  ```
  topology.kubernetes.io/zone
  ```

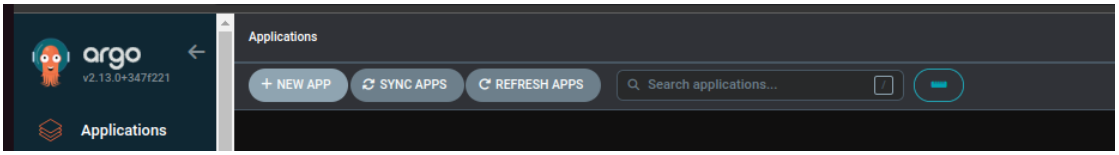
  Example:

  ```
  topology.kubernetes.io/zone=datcenter1
  ```

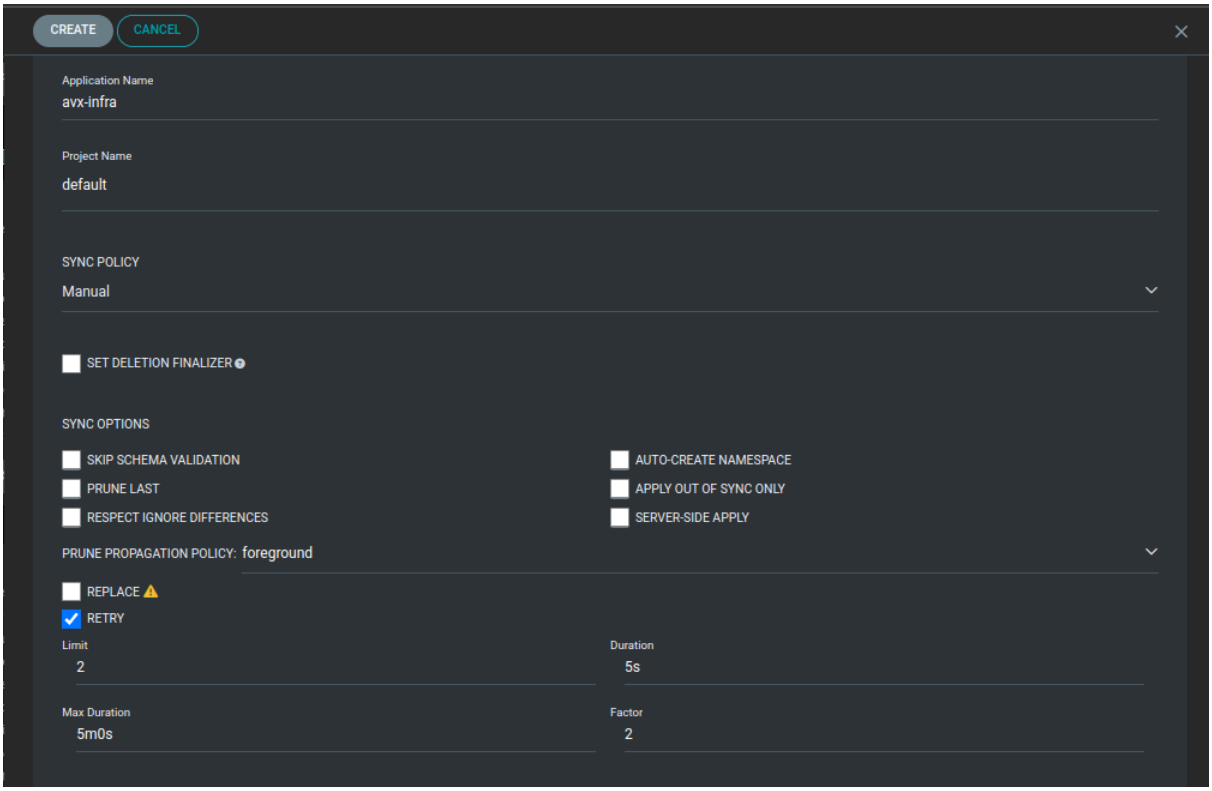
Copyright © 2025 AppViewX Inc. All Rights Reserved.

Chapter 8: Installing avx-infra

1. Once you have logged in to the ArgoCD UI, and after registering the cluster, go to the **Applications** page, click **+ NEW APP** button.



2. Enter the required details (set the **retry option** with **5mins** timeout).



3. Enter the configurations in the ArgoCD that are necessary for avx-infra, as follows:
   - **Application Name:** Can be any unique value
   - **Project Name** (Default value: default)
   - **Sync Policy** (Manual/Automatic - Set it to Manual)
   - **Retry** (Limit - 2, Duration - 5s, Max Duration - 5m , Factor - 2)

   SOURCE

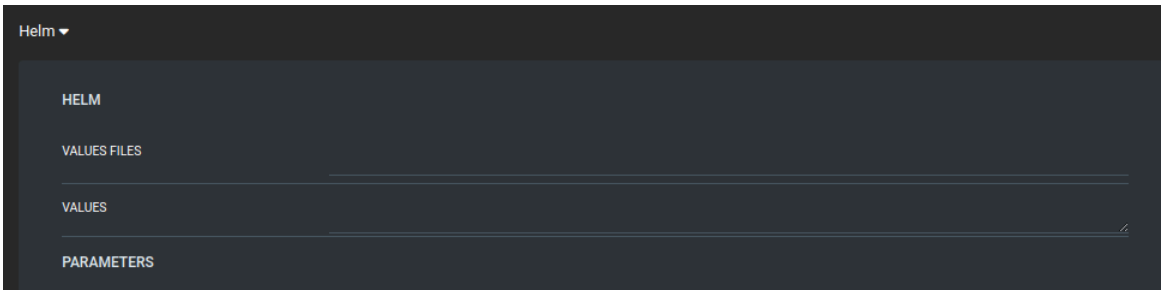
Copyright © 2025 AppViewX Inc. All Rights Reserved.

- **Repository URL:** Git repository in which helm charts are present. ArgoCD will track this repository and deploy the changes while applying sync.
- **Revision:** Branch name of the above git repository to be tracked
- **Path:** Path of the helm chart to be tracked. In case of composite helm charts (parent and child helm chart), provide the path to the parent helm chart. (In case of **avx-infra**, select the **values.yaml** in the **managed_k8s/helm charts** folder in the git repository.)

DESTINATION

- **Cluster URL:** After successfully adding the cluster to ArgoCD during the setup process, the added cluster can be selected from the dropdown.
- **Namespace:** This is the release namespace. (*Leave it empty for avx-infra*)

After entering details in the **Source**, you will find all the values that are in the path of the helm chart displayed in the UI itself. You may add the additional values files or values here.



Copyright © 2025 AppViewX Inc. All Rights Reserved.

Compulsory values to be filled for avx-infra

bitnami-mongodb.auth.rootPassword	
global.minio.accessKey	
global.minio.bucketName	
global.minio.secretKey	
global.minio.server	
global.toolsImage.tag	24.0.1.1-3866
mongo-backup.image.tag	24.0.1.1-3866
global.imageRegistry	registry.appviewx.net
global.imageRepository	appviewx
global.redis.password	
global.imagePullSecrets.name	

Fill the storage class fields created for nutanix volumes and log storage (Default values are present in the helm chart)

global.logStorageClass	<Storage class for log storage>
global.storageClass	<Storage class used for nutanix volume>
openbao.storageClass.openshift	<Storage class used for nutanix volume>

Fill the below field values for strimzi-kakfa-operator and mongoDB

PARAMETERS	
strimzi-kafka-operator.imageRepositoryOverride	appviewx
strimzi-kafka-operator.imageRegistryOverride	registry.appviewx.net
strimzi-kafka-operator.image.imagePullSecrets	<image pull secrets' name created in pre-requisites>
mongo-backup.image.registry	registry.appviewx.net
mongo-backup.image.tag	24.0.1.1-3866
mongo-backup.imagePullSecrets.name	<image pull secrets' name created in pre-requisites>

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Note:

- To enable/disable any particular plugin, update the boolean field 'enable' for that plugin.
- There are limitations to the special characters used in the Redis password. Only use ***** (aserix) and @ (at the rate). This is not a password which is exposed outside of AppViewX and is for internal use only.
- *(Optional step)* If you have your own docker private registry, then enable the **global.useDockerPrivateRegistry** to *true* and enter the username and password in the **global.dockerusername** and **global.dockerpass** fields respectively.

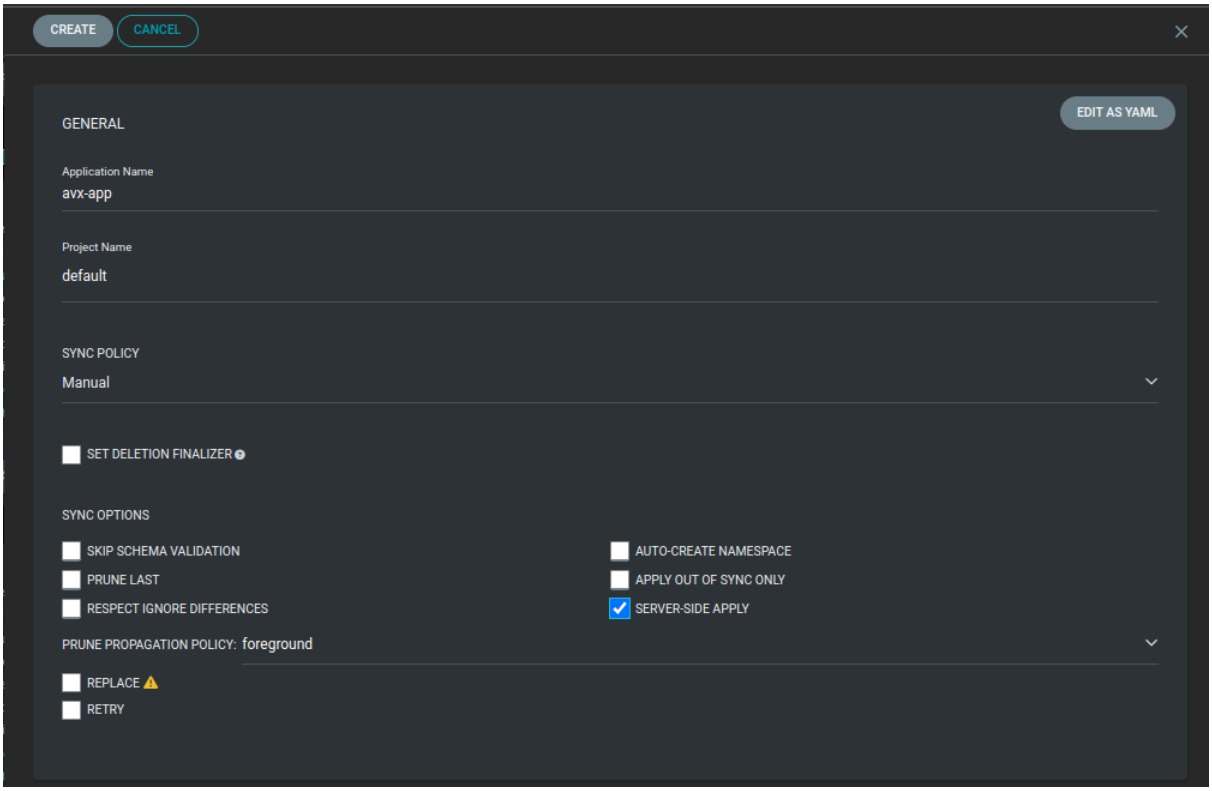


Copyright © 2025 AppViewX Inc. All Rights Reserved.

Chapter 9: Installing avx-app

Note: Perform the steps below only after installing the avx-infra.

1. Once you have logged in to the ArgoCD UI, and after registering the cluster, go to the **Applications** page, click **+ NEW APP** button.
2. Enter the required details (set the **server-side apply** as **true**).



3. Enter the configurations in the ArgoCD that are necessary for avx-app, as follows:
   - **Application Name:** Can be any unique value
   - **Project Name** (Default value: default)
   - **Sync Policy** (Manual/Automatic - Set it to Manual)
   - **Server-side apply** should be **true**.

   SOURCE

   - **Repository URL:** Git repository in which helm charts are present. ArgoCD will track this repository and deploy the changes while applying sync.
   - **Revision:** Branch name of the above git repository to be tracked
   - **Path:** Path of the helm chart to be tracked. In case of composite helm charts (parent and child helm chart), provide the path to the parent helm chart. (In case of **avx-app**, select the **values.yaml** in the **managed_k8s/helm charts/avx_app** folder in the git repository.)

Copyright © 2025 AppViewX Inc. All Rights Reserved.

DESTINATION

- **Cluster URL:** After successfully adding the cluster to ArgoCD during the setup process, the added cluster can be selected from the dropdown.
- **Namespace:** This is the release namespace. (**avx** in case of **avx-app**)

After entering all the required details, similar to avx-infra, all necessary values can be completed directly through the UI.

Compulsory values to be filled for avx-app

Updating Image Tags, Registry, Image pull secrets' name in ArgoCD

- **Code-based Updates (helm_charts/avx_app/values.yaml)**: When a new image tag needs to be updated, the tag must be modified in a single location in the code. This is possible because both the image tag and the registry are defined in a central location within the repository and are referenced elsewhere in the codebase. This approach ensures consistency and reduces the risk of errors when managing updates.

```
images:
  tag: &tag 24.0.1.1-3827
  registry: &registry "registry.appviewx.net"
```

```
# When imagePullSecret is not required, then pass it as name: &name ""
imagePullSecrets:
  name: &name "dummy-secret"
```

- **UI-based Updates**: If updates are made directly through the ArgoCD UI, the tag and registry values must be updated individually for each plugin. This is because ArgoCD resolves and replaces referenced values during synchronization, making it necessary to edit the resolved values separately for every plugin.

Recommendations

- To reduce manual effort and ensure consistency, it is recommended to update image tags and registry details via the Git repository instead of using the ArgoCD UI.
- Enter in the required information for specific plugin details. For example, in avx_**vendors**, if you prefer to use **minio** instead of **aws**, enter the details as shown below.

```
avx-vendors:

 saas_enabled: *saas_enabled

 cloudConnectorEnabled: *cloudConnectorEnabled
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

```
serviceName: avx-vendors.avx.svc.cluster.local:6701
image:
  repository: appviewx/avx-cloud-managedservice-mks
  registry: *registry
  tag: *tag

imagePullSecrets:
  name: *name

externaljar:
  f5JarRequired: "true"
  icontroljar: "iControl-13.1.0.jar"
 minio:
  enabled: true
  server: "<minio server url"
  accessKey: "<access key of minio>"
  secretKey: "<secret key of minio>"
  bucketName: "<minio bucket name"
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Chapter 10: Deleting the avx-app

Note:

- Perform the delete operation only where needed.
- Whenever you delete the avx_app, the avx_python_sandbox pods will be in a terminating state. Always force delete these pods either by logging into the cluster or via the ArgoCD UI. Only then will the avx_app be deleted.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Chapter 11: Upgrading the avx-app

Note the following limitations when upgrading **avx-app** to a higher version of AppViewX:

- Update the helm charts as provided by AppViewX in the repository (github/gitlab).
- In helm_charts/avx_app/values.yaml update the image tag with the new upgrade image version.
- Sync **avx-app** in ArgoCD.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Chapter 12: Migrating Data from On-Premise to OCP

To migrate on-premise data to OCP, the following backups must be available for restoration in the OCP environment:

- Mongo Database Backup
- Vault/Openbao Backup

After completing the restoration, data migration needs to be triggered, which is managed as part of the `avx-app` installation, as described in earlier chapters.

Prerequisites for Restoring Data

The scripts provided by AppViewX for restoring data from on-premise to OCP must be executed from an environment or VM with `kubectl` admin access to the OCP cluster. This access is essential to perform the required operations for copying data and initiating the restore process.

Restoring Mongo Backup

To restore the mongodb backup

1. Fetch the backup tar file and place it in a directory such as /home/user/backup.
2. Execute the mongo restore script as follows:

```
./mongo_restore_ocp.sh <mongo backup tar filepath>
```

Example:

```
./mongo_restore_ocp.sh /home/user/backup/mongo_backup_XX-XX-XX.tar.gz
```

Restoring Vault/Openbao Backup

To restore the vault/openbao backup

1. Fetch the backup tar file and place it in a directory such as /home/user/backup.
2. Execute the vault restore script as follows:

```
../vault_restore_ocp.sh -p <vault backup tar filepath>
```

Example:

```
./vault_restore_ocp.sh -p /home/user/backup/vault_backup_XX-XX-XX.tar.gz
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Post Backup Actions

After the data restoration is complete, the data migration process must be initiated. This step is automatically handled during the installation of the `avx-app`.

Note:

- If the **avx-app** is already installed, it can be upgraded/re-synced, so that the migrations run on the newly restored data.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Chapter 13: MSP Portal User Guide

AppViewX is a leader in automated certificate lifecycle management (CLM) and PKI, providing turnkey solutions for Machine Identity Management and PKI across hybrid infrastructures. The AVX One platform offers a comprehensive multi-tenancy SaaS solution tailored for fast-growing MSPs.

Designed to address complex cybersecurity challenges, the AppViewX MSP portal is a central manager console that features an intuitive interface, providing a unified single-pane-of-glass view and access across customer environments. Built for MSPs serving diverse clients, the console offers end-to-end provisioning, governance, flexible licensing models, tenant insights, reporting capabilities supported by efficient monitoring of multiple customers across various AppViewX cloud environments.

Benefits of AppViewX for MSSPs

- **Comprehensive Machine Identity Management**

  AppViewX provides MSSPs with a comprehensive Machine Identity Management platform, enabling them to manage, secure, and automate the lifecycle of digital certificates, keys, and other machine identities across diverse environments.

- **Enhanced Security Posture**

  By centralizing visibility and control over machine identities, AppViewX empowers MSSPs to proactively identify and mitigate security risks, ensuring continuous compliance with industry regulations.

- **Streamlined Operations and Automation**

  AppViewX's intuitive interface and automation capabilities streamline operations for MSSPs, minimizing manual intervention and reducing the risk of human error.

- **Scalable Solutions for Growth**

  AppViewX offers scalable solutions that cater to the evolving needs of MSSPs and their clients, supporting growth and expansion while maintaining high levels of performance and reliability.

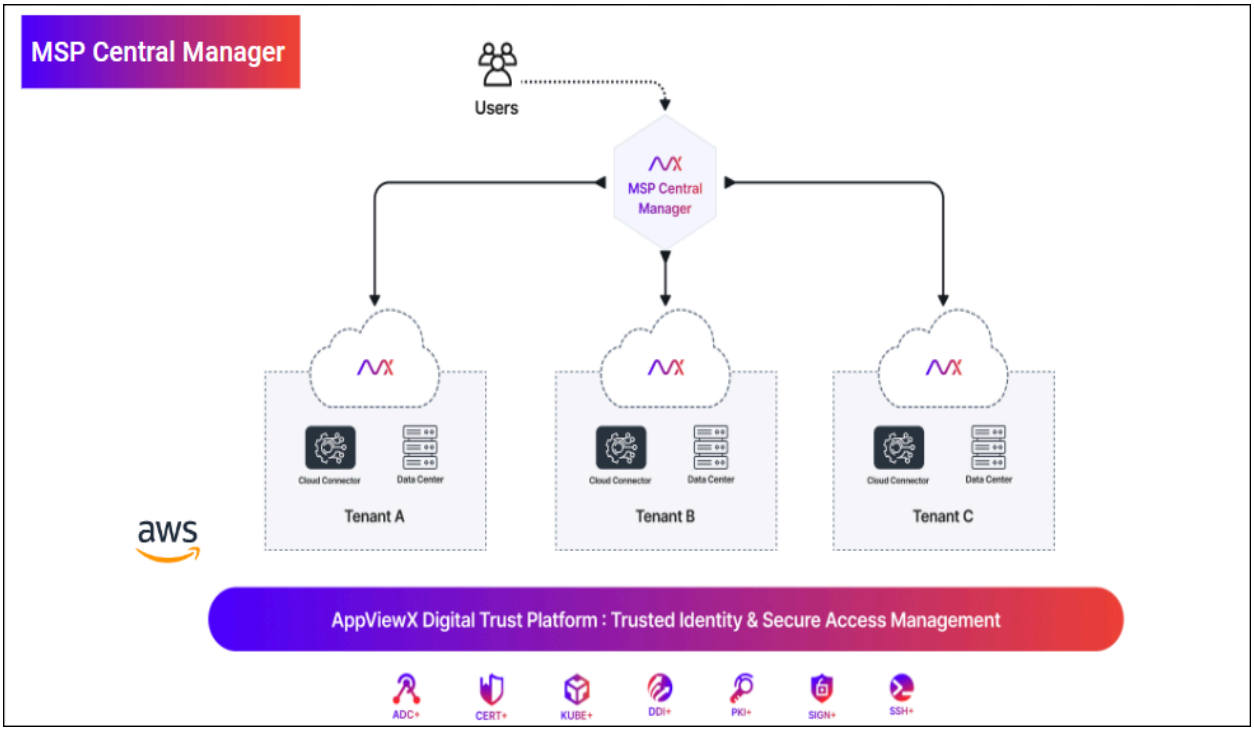
- What is the MSP Portal?
- Key Benefits of the MSP Portal for Partners
- Accessing the MSP Portal
- User Preferences
- Viewing the MSP Portal Dashboards
- Managing White Labels

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- Managing License
- Managing Plans in the MSP Portal
- Viewing Clusters in the MSP Portal
- Managing Tenants in the MSP Portal

What is the MSP Portal?

The MSP portal The MSP Portal is a central manager console that helps a managed service provider (MSP) to deliver services, such as network, application, infrastructure, and security, via ongoing and regular support and active administration on customers' premises. With the help of this centralized console, MSPs can provision various customer accounts, govern, gather useful insights, and act on them.



The MSPs can:

- Create and manage tenants.
- Upgrade tenant licenses.
- Create and manage custom plans for tenants.
- White label MSP portal and the tenant(s).
- Gain holistic insights into tenant application and license metrics.
- Login to tenants and take action.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Key Benefits of the MSP Portal for Partners

The MSP Portal provides a unified, single view of the tenant data across all instances, allowing MSPs to efficiently manage and monitor all their AppViewX Cloud setups from one portal, streamlining oversight and operations.

The key benefits of the MSP central manager include:

- A dedicated white-labeled portal for seamless tenant management from a single console.
- License-to-Service value through flexible license plan - Prepaid and Pay-as-you-go/Subscription model.
- Scale across multiple customers, up-sell/cross-sell across product lines.
- Unified tenant insights in a single view.

Accessing the MSP Portal

After AppViewX's Site Reliability Engineering (SRE) team has set up the MSP Portal for your requirements, you will receive an email with your MSP portal URL, credentials for logging in to the portal, and your license details.

To access the MSP portal:

1. Login to AppViewX with your valid credentials.
2. Go to (**Menu**) > **MSP Portal**.
   The MSP portal dashboard is displayed.

User Preferences

The **User Preference** option on a webpage allows individual users to customize their interface and behavior settings based on their personal needs and usage patterns.

1. Login to AppViewX with valid credentials.
2. From the top right corner of your screen, click **(Profile)** icon.
3. Click **User Preferences**.
   The User Preferences page is displayed.
4. To set the UI preferences for the user, refer to the table below for the fields.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Fields	Description
Language	Select the language for the text to be displayed across the UI from the dropdown options.
Font display	Select the font size for the text to be displayed across the UI from the size options provided.
Font type	Select the font type for the text to be displayed across the UI from the options provided. • OpenSans • Ubuntu • Inter
Date display format	Select the date display format to be displayed across the UI from the options provided. • MM/DD/YYYY • DD/MM/YYYY • YYYY/MM/DD **Note:** The date format selected here will be used in all the inventory pages, audit logs, and the form (date) fields across the application.
Landing page	Select the preference for the landing page to be displayed upon logon from the **Preference** dropdown.

5. Click **Save**.

Viewing the MSP Portal Dashboards

1. Go to (**Menu**) > **MSP Portal** > **Governance**.
2. From the following options, select the dashboard you want to view:
   - **Usage and Adoption**
   - **Tenants & License**

   By default, on login, the **Tenants & License** dashboard is displayed.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

• Usage and Adoption
• Tenants Insights

Usage and Adoption

The **Usage and Adoption** dashboard offers the MSPs holistic insights into application usage and license information across all tenants, as well as into the up-sell and cross-sell opportunities based on usage. This dashboard presents the following tenant-related data across product lines for which the license is enabled:

- Tenant summary
- Tenants by product line
- Platform adoption
- CERT-SSL/TLS adoption
- CERT lifecycle management operations
- ADC app and device adoption
- ADC app actions and device operations
- DDI adoption
- KUBE adoption

Tenants Insights

The Tenant Insights dashboard offers MSPs a single pane of glass view into the tenants by product lines and a seamless way to access individual tenants to perform remedial actions. Data under tenant insights is categorized as:

- **Product Insights**
  - This dashboard lists insights for all product lines for which tenants have been onboarded.
  - To view the insights for an individual product line:
    - Under **Dashboard Name**, click the insight name.
    - For the selected product line, this insights dashboard shows certificate category-wise reports for CAs and certificate status.
    - You can use the **Filter by tenant** dropdown list to filter the reports for individual tenants and login to the selected tenant's environment. For instructions on how to login/impersonate a tenant, click here.
- **License Insights**

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- This dashboard shows the license usage for each product line for which tenants have been onboarded.
- To view the license usage details for a product line, click the accordion for the product line to expand (and collapse) the details view.
- You can use the **Filter by tenant** dropdown list to filter the license usage details for individual tenants.

Managing White Labels

White labeling is essential for MSPs' tenants because it allows the MSPs to customize the platform with their own branding, providing a seamless experience for their tenants.

To upload the logos and favicons that will be displayed on the tenants' applications and emails, go to (**Menu**) > **MSP Portal** > **Management** > **White Label** Refer to the section Customizing the AppViewX User Interface for the detailed steps.

Note: It is crucial to customize the email templates sent to customers to prevent AppViewX logos being displayed for the MSP’s customers.

Managing License

This section provides guidance on managing the AppViewX license effectively. It covers processes such as accessing the License page, upgrading the license, and monitoring license product usage, ensuring proper license management.

Note:

- AppViewX blocks the usage/upload of the old license. If an organization uploads an old license, an error message is displayed. **LICENSE_ERROR_037:** Unsupported version of the license file. Please contact AppViewX support. The AppViewX support team will help with the new and latest supported versions of the license.
- AppViewX blocks the usage of the product if the license has expired. If an organization exceeds the threshold of license (maximum number of ADC+ objects, CERT+ certificates, and/or PKI+ Certificates allowed), a warning is issued as a reminder. A pop-up warning is issued for both scenarios.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

To access the licenses page, go to (**Menu**) > **MSP Portal** > **Management** > **License**.

For detailed information on managing licenses refer to the platform guides - Managing Licenses - Licenses for SaaS.

Managing Plans in the MSP Portal

The **Plan Management** inventory lists the predefined plans as well as the custom plans (created when the predefined plans don't align with the tenants' requirements) for each product type offered by AppViewX. This inventory is the source for the **Plan & Product Details** section that is displayed at the time of creating and upgrading a tenant.

- Creating Custom Plans

Accessing the Plan Management Inventory

Go to (**Menu**) > **MSP Portal** > **Management** > **Plans**.
The **Plan Management** inventory is displayed.

Understanding the Plan Management Inventory

Fields in the Plan Management Inventory

Field	Description
Search	Enter a keyword/list of keywords to search for records that match the search value.
Add Plan	Button to create custom plans.
Plan Name	AppViewX-assigned name of the plan
Product Type	Product type that is part of the plan

Plan Category	• For ADC+, plans belong to the following categories:
	  • ADC nodes
	• For CERT+, plans belong to the following categories:
	  • Server certificate
	  • Device certificate
	  • Managed

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Fields in the Plan Management Inventory (continued)

Field	Description

	• For PKI+, plans belong to the following categories:
	  • Certificate authority
	  • Issuance certificate
	• For DDI+, plans belong to the following categories:
	  • DDI Objects
	• For KUBE+, plans belong to the following categories:
	  • Kube Cluster
	  • Kube Node
	• For SIGN+, plans belong to the following categories:
	  • Signatures
	• For SSH+, plans belong to the following categories:
	  • SSH Host
	  • SSH Key

Plan Type	Specifies if the plan is a licensed one or for a free trial
Source Type	Tenants can be onboarded from the AppViewX. This field will have the value: • Internal (plan defined by AppViewX's PM team)
Plan Description	Additional details related to the plan
Action	 : View plan details.  : Delete a plan  **Note:** Only custom plans can be deleted.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Fields in the Plan Management Inventory (continued)

Field	Description
	To delete a plan: 1. From the **Plan Management** inventory, for the required plan, from the Action column, click  . The **Delete Plan** confirmation dialog box is displayed. 2. Click **Yes**. The selected plan is deleted.

Creating Custom Plans

Custom plans can be created when any of the predefined product plans do not meet the tenants' requirements. Your tenant success team or the sales team will raise a request for a custom plan, as required by a tenant. The AppViewX SRE team then creates a custom plan based on the requirement. These custom plans can then be used for all tenants.

Note:

- Custom plans can be created only for licensed versions and not for free trials.
- If yours is a pool-based license, the MSP's license will be validated for the product limits before a custom plan is created.

To create a custom plan:

1. Go to (**Menu**) > **MSP Portal** > **MSP Management** > **Plans**.
   The **Plan Management** inventory is displayed.
2. From the **Plan Management** inventory, click **Add Plan**.
   The **Add Plan** page is displayed.
3. Enter/Select the custom plan details.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Field descriptions for the custom plan details

Field	Description
***Plan Source Type**	Tenants can be onboarded from the AppViewX website. Plans are, therefore, created in accordance with the source of onboarding. • If this new plan has been created by AppViewX's PM team, from the dropdown list, select **Internal**.
***Product Type**	From the following options, select the product type for which this new plan will be applicable: • ADC+ • DDI+ • CERT+ • SIGN+ • KUBE+ • PKI+ • SSH+

***Plan Category**	• For ADC+, plans belong to the following categories:
	  • ADC nodes
	• For CERT+, plans belong to the following categories:
	  • Server certificate
	  • Device certificate
	  • Managed
	• For PKI+, plans belong to the following categories:
	  • Certificate authority
	  • Issuance certificate
	• For DDI+, plans belong to the following categories:
	  • DDI Objects
	• For KUBE+, plans belong to the following categories:
	  • Kube Cluster
	  • Kube Node
	• For SIGN+, plans belong to the following categories:
	  • Signatures
	• For SSH+, plans belong to the following categories:
	  • SSH Host
	  • SSH Key

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Field	Description
***Number of Managed Devices**	This field is displayed only **Product Type** = **ADC+** Enter the number of devices that will be managed by this plan.
***Number of Managed Certificates**	This field is displayed only **Product Type** = **CERT+**.and **Plan Category** = **Managed**. Enter the number of device/server certificates that will be managed by this plan.
***Number of Server Certificates**	This field is displayed only **Product Type** = **CERT+**.and **Plan Category** = **Server**. Enter the number of server certificates that will be managed by this plan.
***Number of Device Certificates**	This field is displayed only **Product Type** = **CERT+**.and **Plan Category** = **Device**. Enter the number of device certificates that will be managed by this plan.
***Number of Issuance Certificates**	This field is displayed only **Product Type** = **PKI+** and **Plan Category** = **Issuance Certificate**. Enter the number of issuance certificates that will be managed by this plan.
***Number of CA**	This field is displayed only **Product Type** = **PKI** and **Plan Category** = **Certificate Authority**. Enter the number of certificate authorities that will be managed by this plan.
***Number of DDI Objects**	This field is displayed only **Product Type** = **DDI+** Enter the number of DDI objects that will be managed by this plan.
***Number of Managed Clusters**	This field is displayed only **Product Type** = **KUBE+** and **Plan Category** = **Kube Clusters**. Enter the number of Kube clusters that will be managed by this plan.
***Number of Managed Nodes**	This field is displayed only **Product Type** = **KUBE+** and **Plan Category** = **Kube Nodes**. Enter the number of Kube clusters that will be managed by this plan.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Field	Description
***Number of Managed Signatures**	This field is displayed only **Product Type** = **SIGN+**. Enter the number of signatures that will be managed by this plan.
***Number of Managed SSH Hosts**	This field is displayed only **Product Type** = **SSH+** and **Plan Category** = **SSH Hosts**. Enter the number of SSH hosts that will be managed by this plan.
***Number of Managed SSH Keys**	This field is displayed only **Product Type** = **SSH+** and **Plan Category** = **SSH Keys**. Enter the number of SSH keys that will be managed by this plan.
***Plan Name**	By default, based on the above selections, a plan name is auto-populated in this field, in adherence to AppViewX's recommended naming convention (which is used for the predefined plans). If required, you can edit the plan name.
***Plan Description**	Enter any additional details related to the plan.

4. Click **Create**.
   The new plan is listed in the **Plan Management** inventory.



Note: If a plan has to be modified, the recommended course of action is to delete that plan and create a new one.

Viewing Clusters in the MSP Portal

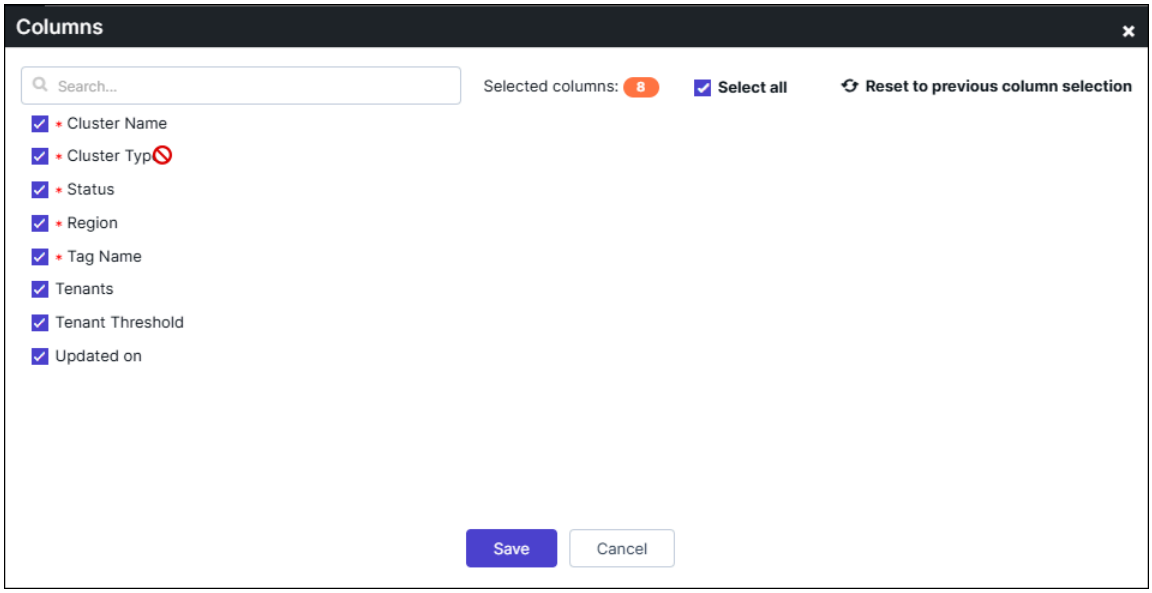
Accessing the Cluster Management Inventory

Go to  (**Menu**) > **MSP Portal** > **Management** > **Clusters**.
The **Cluster Management** inventory is displayed.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Understanding the Cluster Management Inventory

Fields in the Cluster Management Inventory

Field	
Filter	Lets you filter the cluster inventory entries for the following parameters: • Cluster type • Region • Tag name • Version • Status
Columns	Button to customize the cluster inventory columns to be displayed. On clicking the Columns button, a Columns  **Note:** Column selection is user-specific and remains saved across logout and subsequent login sessior
Cluster Name	Displays the name of the cluster
Cluster Type	Displays the type of the cluster. The valid types are: • Compute • Database

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Fields in the Cluster Management Inventory (continued)

Field	
Status	Displays if the cluster is active.
Region	Displays the region in which the cluster is created.
Tag Name	Displays the name of the associated tag.
Tenants	Displays the number of tenants associated with the respective cluster.
Updated on	Displays the date of the last cluster update.
Action	Displays the possible actions that can be taken on the cluster. (Currently MSPs don't have the option to modify

Managing Tenants in the MSP Portal

Tenant management in the MSP Portal includes the provision to create, delete, and move tenants across clusters, extend the free trial of a tenant as well as move tenants from a free trial to a licensed account, and so on.

- Creating a New Tenant
- Extending Free Trial
- Activating a Tenant License
- Deleting Tenants
- Upgrading License Details for a Tenant
- Repropagating Tenant Details
- Enable mTLS
- Disable mTLS
- Viewing MSP License Usage Details
- Accessing the Tenant Environment

Accessing the Tenant Management Inventory

- Go to (**Menu**) > **MSP Portal** > **Management** > **Tenants**.

  The **Tenant Management** inventory is displayed.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

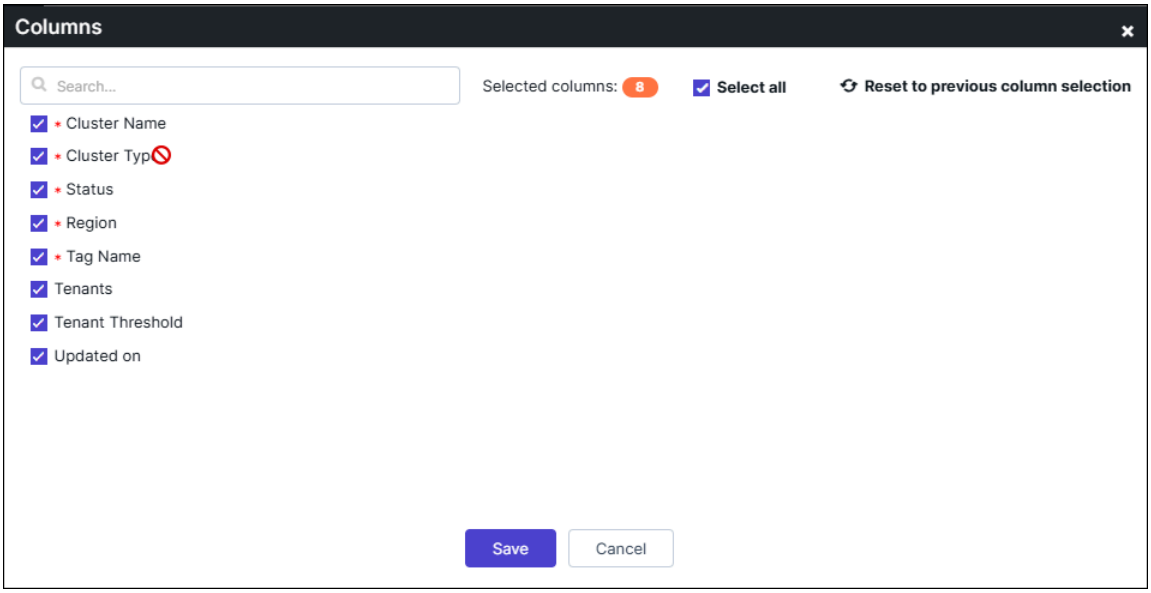
Understanding the Tenant Management Inventory

Fields in the Tenant Management Inventory

Field	Description
Filter	Allows you to apply filters to the tenant inventory to search for specific tenant records. The available filters are: • Plan • Tenant type • Request type • Cluster • Region • Product type • Status
Add Tenant	Allows you to onboard a tenant manually
Action	Dropdown for the possible actions that can be taken on the tenant. The possible actions are: •  (Tenant Login) •  (Fetch IDP Service URL) •  (Delete Tenant) •  (Extend Trial) •  (Tenant Cluster Movement) •  (Activate License) •  (Repropagate tenant details) •  (Enable mTLS)
Columns	Button to customize the cluster inventory columns to be displayed. On clicking the Columns button, a Columns pop-up is displayed as shown below. By default, all the columns will be

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Fields in the Tenant Management Inventory (continued)

Field	Description
	selected. Choose the required columns to be displayed and save the changes. There are options to **Select all** and **Reset to previous column selection**.  **Note:** Column selection is user-specific and remains saved across logout and subsequent login sessions.
Tenant Domain	Displays the domain used by the tenant to access the product. Clicking on the domain name (for Active tenants) will display a Tenant Details pop-up with all the processed as inputs values in the view only mode.
Company Name	Displays the name of the tenant company
Email	Displays the tenant's registered email ID
Products and Plans	Displays the AppViewX product name(s) that the tenant has a trial/licensed version of and the AppViewX plan that the user has subscribed to. The plan name gives the details of the product configuration/capability made available to the tenant. On hovering over the value in the column, this field displays the product and plan details.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Fields in the Tenant Management Inventory (continued)

Field	Description
	The plan name gives the details of the product configuration/capability made available to the tenant. It also displays the status of the tenant if they have opted for the trial plan
Status	Displays the status of the tenant as: • Active • Inprogress • Expired • Failed
Request Info	Displays the information of the workflow request processed for tenant related actions such as onboarding, License upgrade and so on. **Note:** To view the request, click the request ID. A stage-wise view of the workflow request is displayed.
Request Status	Displays the status of the tenant addition request

Creating a New Tenant

Note: All tenant creation and management activities will be validated against the plans in the MSP license. MSPs can create and manage tenants only within the limits specified in the plans and for the specified products. Otherwise, the application will throw an error.

1. Go to (**Menu**) > **MSP Portal** > **Management** > **Tenants**.
   The **Tenant Management** inventory is displayed.
2. From the **Tenant Management** inventory, click **Add Tenant**.
   The **Add Tenant** page is displayed.
3. Enter/Select the **General** tenant details.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Field descriptions for the General tenant details

Field	Description
***Company Name**	Enter the name of the tenant's company. For example: newco.
***Email**	Enter a valid email address of the account owner. For example: xxx@newco.com
***Customer Type**	From the dropdown list, select **Customer**.
**: Mandatory fields*	

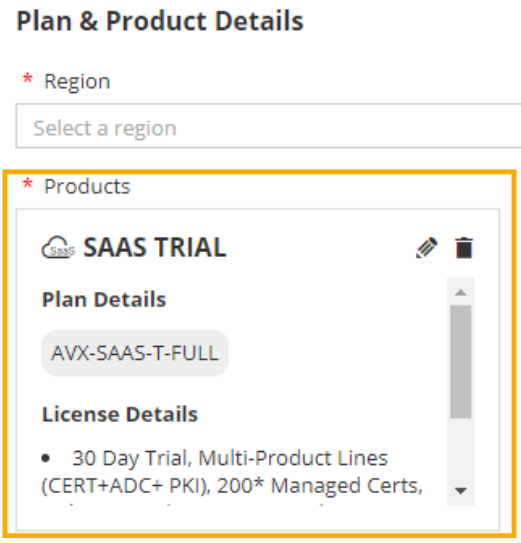
4. In the **Tenant Details** section, in the **Domain** field (mandatory), enter a valid domain name.
5. Select the **Plan & Product Details** for this tenant.

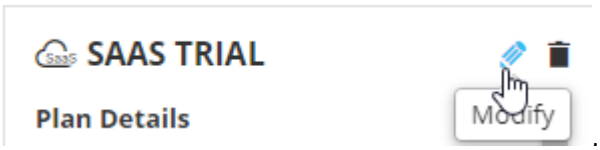
Field	Description
***Region**	Select the region in which the tenant's SaaS account will be set up.
***Products**	a. Click **Add Product**. The **Modify Product** window is displayed. b. From the **Product Type** dropdown list, select the product for which the tenant is evaluating the trial version/has purchased the license. c. If the**Product Type** = **CERT+**, select the **License Subscription Model** for CERT+: • **Instance based**: Certificate count is derived from the number of connectors associated with a certificate. • **Certificate based**: Certificate count for the license is based on the number of certificates in the inventory. For more details on the CERT+ licensing models, refer to the CERT+ User Guide. d. From the **Plan Name** dropdown list, select the plan(s) chosen by the tenant. This list is populated based on the selected product type. **License Details of the selected plan(s)** are displayed in the **Modify Product** window. This banner displays all the details of the license plan selected, such as the number of certificates managed under each plan, issuing CAs, trial duration, and so on. e. In the **End Date** field, click  to specify the expiration date according to the selected product and plan. **Note:** This field is not applicable for a trial tenant.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

f. Click **Add**.

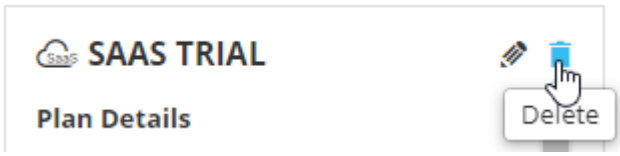
The product details for the selected product(s) are displayed as shown in the image below:



g. To modify the plan details, click 

.

h. To delete the plan details, click



.

i. If the tenant is using a licensed version of multiple product types and wants to have a common license end date for all product types and plans selected, enable the **Enable Common License End Date** toggle key.

j. In the **End Date** field, click  to select a common license end date for all product types. Refer to the tenant's purchase order for the latest license end date for this tenant.

Note: If you need to modify this end date after a tenant has been created, you will be required to click **Activate License/Update License** from the **Action** column for the required tenant and update the **Plan date**.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

6. Select the **Cluster & Other Details**.

Field descriptions for the Cluster & Other Details

Field	Description
***Tag Name**	Select the applicable tag from the dropdown menu. **Note**: Tags are loaded on the basis of the region and plan associated.
**: Mandatory fields*	

7. Click **Add**.
   The tenant is added to the inventory.

   The tenant's URL, username, and temporary password are automatically emailed on the email address specified in the **General** tenant details.

   The MSP header, logo, and favicon, as configured in the MSP Portal, is also applied to the end tenant's environment.

Extending Free Trial

You have the provision to extend the duration of a tenant's free trial. This is subject to approvals from the relevant stakeholders based on discussion with the prospect requesting trial extension. As a standard practice, a maximum extension of 60 days will be allowed, based on approval.

Process flow for trial extension

1. Prospect requests for **X** days of trial extension.
2. MSP gets notified on the account and trial extension duration.
3. MSP extends the free trial period.

To extend the free trial duration of a prospect/tenant:

1. Go to (**Menu**) > **MSP Portal** > **Management** > **Tenants**.
   The **Tenant Management** inventory is displayed.
2. From the **Tenant Management** inventory, for the required trial tenant, from the **Action** column, click .

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Tip: You can also hover your mouse over the selected tenant to view the  icon.

The **Tenant Trial Extension** page is displayed. The tenant details are auto-populated and non-editable.

Field	Description
***Tenant Id**	Tenant ID
***Region**	Region in which the tenant is located
***Product Type**	Product type for which the tenant is evaluating the trial version
***Current Plan**	Current plan details of the tenant.
***Current expiry date**	Current expiry date of the trial period
Remaining Trial Extension Period	Once the default 30-day trial duration expires, AppViewX lets you extend the trial duration for an additional 60 days. This field is auto populated with the number of days left out of the 60-day additional extension duration.

3. To extend a tenant's trial period, in the **Number of days to extend** field, enter the number of days for which the trial is to be extended.
   Once the default 30-day trial duration expires, AppViewX lets you extend the trial duration for an additional 60 days (which makes the total trial duration to be 90 days). The trial period can be extended by a minimum of 15 days and a maximum of 30 days at one time. So, you can extend the trial:
   • twice for 30 days each time
   • four times for 15 days each time
   • twice by 15 days and once by 30 days

Note:
If the value entered in this field is more than the **Remaining Trial Extension Period**, an error message is displayed.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

4. Click **Extend Trial**.
   The trial period is extended and all the columns in the inventory are replaced with the current values for the tenant.

Activating a Tenant License

> **Note:** All tenant creation and management activities will be validated against the plans in the MSP license. MSPs can create and manage tenants only within the limits specified in the plans and for the specified products. Otherwise, the application will throw an error.

1. Go to (**Menu**) > **MSP Portal** > **Management** > **Tenants**.
   The **Tenant Management** inventory is displayed.
2. From the **Tenant Management** inventory, from the **Action** column, click .

   > **Tip:** You can also hover your mouse over the selected tenant to display the icon.

   The **Upgrade Tenant** page is displayed.

   > **Note:** Fields in the **Tenant Details** section are auto-populated and cannot be edited.

3. In the **Upgrade Plan & Product Details** section:
   a. From the **License Action** dropdown list, select the reason for the license upgrade.
   b. From **Products**, click **Add Products** to modify the product and plan details for the tenant.
      For descriptions of the fields in this section, refer to the Creating a New Tenant section.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Important: The **Upgrade Plan & Product Details** section includes the following warning: **Only the below selected product license will be pushed to tenant. Make sure to select the necessary products and plans.** This means that all upgrades done to the license here will overwrite the existing license details for the tenant. Users are, therefore, advised to ensure that the upgraded license includes all products and plans that a tenant requires, and not just the modifications.

Note: In the event that the upgrade includes additional products added to the tenant's license, AppViewX includes the **Enable Common License End Date** to set a common license end date for all of a tenant's new and existing products, so that all products can then be renewed on the same date.

4. In the **Cluster & Other Details** section, Select a **Tag Name** from the dropdown field.

   Note: Tag(s) are loaded depending upon the region and plan name selected.

5. Click **Upgrade**.

   The tenant license is upgraded as specified.

   Note: You can check the workflow execution of this request by clicking the Request ID in the **Request Info** column in the tenant inventory.

Deleting Tenants

MSP Tenants that are in *Active, Expired,* or *Failed* states and *Trial* tenants will have the 'delete' action button option on the tenant inventory page. For tenant deletion in **MSP** (Managed Service Provider), it is crucial to designate **Level-1** (L1) and **Level-2** (L2) approvers. These approvers are users or user groups authorized to approve tenant deletion requests.

- **L1 Approvers:** The first set of users or user groups responsible for validating and approving the initial deletion request.
- **L2 Approvers:** The second set of approvers who review and approve the request after it has been validated by the L1 approvers.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

The **Delete Tenant Approval** Settings page allows you to assign L1 and L2 approvers. In the MSP provisioning cluster, the *MSP admin* user has default authority to select approvers. Although any MSP sub-user can initiate a tenant deletion request, the request will only succeed if it is approved by the designated L1 and L2 approvers configured in the deletion settings menu.

- Delete Tenant Approval Settings
- Deleting Tenants On-Demand
- Auto-Deletion of Trial/Production Tenants

Delete Tenant Approval Settings

1. Go to (**Menu**) > **Settings** > **Delete Tenant Approval**.
   The Delete Tenant Approval page is displayed.
2. In the Delete Tenant Approval page enter the field information as described in the table below:

Field description for delete tenant approval.

Field	Description
Level 1 Approver	(Mandatory) Configure either **User**(s) or **User group**(s) as level 1 approver for tenant deletion action. Select one or many users/user groups from the dropdown. **Note:** If the user is associated with the *MSP User Group* or *MSP Access Resource*, they can approve the deletion request without being added as L1 or L2 approvers.
Level 2 Approver	(Mandatory) Configure either **User**(s) or **User group**(s) as level 2 approver for tenant deletion action. Select one or many users/user groups from the dropdown. **Note:** • The L1 approver (user) must be different from the L2 approver (user). • If the user is associated with the *MSP User Group* or *MSP Access Resource*, they can approve the deletion request without being added as L1 or L2 approvers.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Field	Description
***Licensed Tenant Retention Period (Retention Period Time(in days))**	Enter the retention period value between 30 and 366 days. **Note:** Trial tenants have a fixed retention period of 30 days. For production tenants, the retention period can be configured between 30 and 366 days.
**: Mandatory fields*	

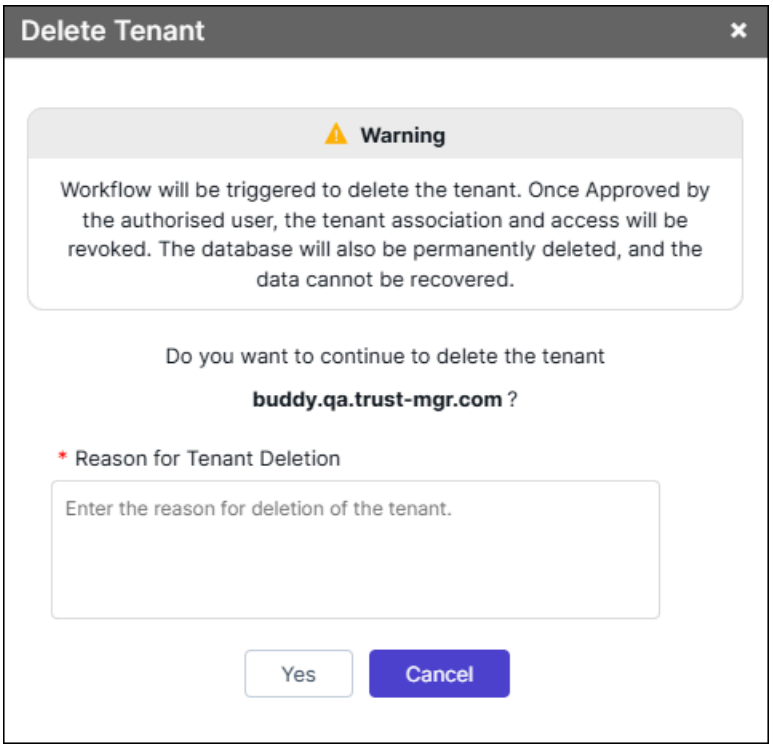
3. Click **Save.**

Deleting Tenants On-Demand

To delete a tenants/trial customers on-demand:

1. Go to (**Menu**) > **MSP Portal** > **Management** > **Tenants**.
   The **Tenant Management** inventory is displayed.
2. From the **Tenant Management Inventory**,select the checkboxfor the required tenant, and from the enabled **Action** dropdown, select **Delete Tenant**.
   The **Delete Tenant** pop-up window is displayed.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



3. Enter the **Reason for Tenant Deletion**.
4. Click **Yes**.

   The offboarding workflow will be triggered and an email is sent to the L1 approver, with a link to approve/reject the request.
5. In the email, click on the link that directs to the request page for L1 approver. Verify the tenant details to be deleted and click **Approve**.

   An email is sent to the L2 approver, with a link to approve/reject the request.
6. In the email, click on the link that directs to the request page for L2 approver. Verify the tenant details to be deleted and click **Approve**.

   The Offboarding process continues with the following steps.
   a. All tenant mappings and kafka topics are removed
   b. The DNS records are deleted.
   c. The tenant database is deleted.
   d. An offboarding email is sent to the user who has initiated the request.
   e. The status of the tenant in the inventory is displayed as Deleted.

Note: After a tenant is deleted from the system, the MongoDB backup for the deleted tenant remains stored in AWS S3 for 30 days. After this period, the backup is permanently deleted.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Auto-Deletion of Trial/Production Tenants

As the trial period comes to an end, the customer can opt for the following:

- *Extension of the trial period*: The trial period can be extended for up to a maximum of 60 days. For more information on extending the trial period, click here.
- *Upgrading to a licensed version*: At the end of the trial period, the tenant can be upgraded to a licensed version. For more information on this, click here.
- *Not upgrade*: If a trial tenant does not wish to upgrade to a licensed account after the retention period is over, the tenant database will be deleted automatically at the end of the day through a cron job and will be marked with status as "deleted".

> **Note:**
> - After a tenant is deleted from the system, the MongoDB backup for the deleted tenant remains stored in AWS S3 for 30 days. After this period, the backup is permanently deleted.
> - Trial tenants have a fixed retention period of 30 days. For production tenants, the retention period can be configured between 30 and 366 days from the Delete Tenant Approval Settings page.

Upgrading License Details for a Tenant

To upgrade a tenant's license configuration:

1. Go to (**Menu**) > **MSP Portal** > **Management** > **Tenants**.
   The **Tenant Management** inventory is displayed.
2. From the **Tenant Management** inventory, from the **Action** column, click .

   > **Tip:** You can also hover your mouse over the selected tenant to display the icon.

   The **Upgrade Tenant** page is displayed.

   > **Note:** Fields in the **Tenant Details** section are auto-populated and cannot be edited.

3. In the **Upgrade Plan & Product Details** section:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

a. From the **License Action** dropdown list, select the reason for the license upgrade.

b. From **Products**, click **Add Products** to modify the product and plan details for the tenant.
For descriptions of the fields in this section, refer to the Creating a New Tenant section.

> **Important:** The **Upgrade Plan & Product Details** section includes the following warning: **Only the below selected product license will be pushed to tenant. Make sure to select the necessary products and plans.** This means that all upgrades done to the license here will overwrite the existing license details for the tenant. Users are, therefore, advised to ensure that the upgraded license includes all products and plans that a tenant requires, and not just the modifications.

> **Note:** In the event that the upgrade includes additional products added to the tenant's license, AppViewX includes the **Enable Common License End Date** to set a common license end date for all of a tenant's new and existing products, so that all products can then be renewed on the same date.

4. In the **Cluster & Other Details** section, Select a **Tag Name** from the dropdown field.

> **Note:** Tag(s) are loaded depending upon the region and plan name selected.

5. Click **Upgrade**.
The tenant license is upgraded as specified.

> **Note:** You can check the workflow execution of this request by clicking the Request ID in the **Request Info** column in the tenant inventory.

Repropagating Tenant Details

To repropagate tenant details:

1. Go to (**Menu**) > **MSP Portal** > **Management** > **Tenants**.
The **Tenant Management** inventory is displayed.
2. From the **Tenant Management Inventory**,for the required tenant, from the **Action** column, click .

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Tip: You can also hover your mouse over the selected tenant to view display the icon.

3. In the **Re-propagate Tenant Details** confirmation dialog box, click **Yes**.

Enable mTLS

To secure communication between the tenant and the Cloud Connector, mTLS must be enabled during installation. Previously, this required the AppViewX team to :

- Generate root and client certificates.
- Upload the certificates to the Cloud Connector.
- Manually enable the relevant WAF rule in Cloudflare.

To streamline this process, the **Enable mTLS** action in the tenant inventory is used to trigger the workflow that automates or guides these steps, ensuring secure and consistent configuration during Cloud Connector deployment.

To enable mTLS for cloud connectors,

1. From the **Tenant Management Inventory**, select the checkboxfor the required tenant, and from the enabled **Action** dropdown, select **Enable mTLS**.
   The **RequestEnable mTLS Certificate in Cloud Connectors :: FormBuilder** page is displayed.
2. From the **mTLS Details** section, enter/select from the following fields:

Field	Description
***Select the action to perform?**	Select from the following options: • Enable MTLS • Upload Client cert to CC
***CC Agents**	Select the CC agents from the dropdown.
**: Mandatory fields*	

3. From the **Tenant Details** section, enter/select from the following fields:

Field	Description
***Tenant Name**	Enter the tenant name.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Field	Description
***Tenant Id**	Enter the tenant id.
***Tenant Username**	Enter the tenant username.
***Tenant Password**	Enter the tenant password.
***Provisioning Gateway URL**	Enter the tenant's provisioning gateway URL.
***MSP Mode**	Enter the MSP's mode.
**: Mandatory fields*	

4. In the Notification Details section, enter the ***Email Id** to whom the workflow success or failure email is sent.
5. Click **Submit**.

   The workflow executes in the following stages:

   a. The root certificate is created via API and uploaded to Cloudflare.
   b. The active CCs are fetched for the tenant and installed.
   c. For each CC the client certificates are generated.
   d. The enable WAF workflow rule is executed to add the WAF rule in the Cloudflare.

Note:

Disable mTLS: mTLS can be disabled by the AppViewX SRE team upon request. Please reach out to the SRE team to initiate the mTLS disablement process.

Disable mTLS

Disabling mTLS refers to switching the cloud connector communication mode from mTLS to Standard TLS. It uses the same workflow as the Enable mTLS. The Disable mTLS feature can be performed by the SREs only, upon request.

1. From the **Tenant Management Inventory**,select the checkboxfor the required tenant, and from the enabled **Action** dropdown, select **Disable mTLS**.

   The **Tenant Management Screen** page is displayed with the **Tenant Details** section and **mTLS Details** section autofilled and read only.
2. In the **Cloud Connector Details** section, select the respective CC (multiple CCs can be selected, as required).
3. Click **Disable mTLS**.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

The workflow request is generated for disabling the mTLS.

Note: To view the workflow stages, from the **Request Info** column, click the specific request id.

In case of any failures the workflow is retried three times. If unsuccessful, the SREs will have to manually resolve the issues and reinitiate the disable mTLS workflow.

Viewing MSP License Usage Details

Go to (**Menu**) > **Platform** > **System Administration** > **License**.
The **Settings :: License** page is displayed.

Multiple metrics to track your license usage are displayed on this page.

For a pool based license, the metrics will show the permitted and used count of resources.

For a pay as you go license, the metrics will show the number of resources used; there is no maximum limit for a pay as you go license.

Accessing the Tenant Environment

Tenant login allows MSP admins to act on behalf of their clients with the help of MSP credentials. This feature enables MSP admins to provide support, manage resources, and perform administrative tasks directly within the clients' environments.

MSPs have two options for logging into tenant environments:

1. When logged in as local users, MSPs can use the same credentials to access the tenant.
2. If logged in as SSO users, MSPs can seamlessly log into tenant environments without re-entering SSO credentials.

Note: MSP admins can login to a tenant only if the tenant, at the time of their first login, has selected the Allow MSP Users to access this tenant in the EULA/terms and conditions page.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

• Setting ACL Permissions for Tenants
• Tenant Environment Login Methods
• SRE Access to MSP Portal

Accessing Tenants via Dashboard

1. Go to (**Menu**) > **MSP Portal** > **Governance** > **Tenant Insights** > **Product Insights**.
   The **Tenant Insights** page is displayed with a list of the product-wise insight dashboards.
2. From the **Dashboard Name** column, click to open a product insight dashboard.
3. From the **Filter by tenant** dropdown list, select and tenant and click **Login**.
   You will be redirected to the tenant's environment.
4. Login to the tenant's environment using your MSP credentials.

Accessing Tenants via Tenant Inventory

1. Go to (**Menu**) > **MSP Portal** > **Management** > **Tenants**.
   The Tenant Management inventory is displayed.

2. For the tenant you want to login to, from the tenant inventory, under the **Action** column for this tenant, click (**Tenant Login**).
   The **Tenant Login** pop-up will be displayed with the tenant URL and **Continue** and **Cancel** buttons.
3. Click **Continue**.
   You will be redirected to the tenant's environment.
4. Login to the tenant's environment using your MSP credentials.

Setting ACL Permissions for Tenants

To define a tenant's visibility into the application, you can assign ACL permissions as follows:

1. Go to (**Menu**) > **Platform** > **Identity** > **Resource**.
   The **Resource** page is displayed.
2. From the list displayed on this page, click a resource name.
   The **Resource > Modify :: <resource name>** page is displayed.
3. Under **Access Control** > **List**, select **Tenants**.
   The records section is updated to list all the tenants that have been onboarded.
4. To map this tenant to the resource selected in step 2, select the checkbox against the tenant name.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

5. Click **Save**.
   The tenant is mapped to the resource, which in turn is mapped to a user group

Tenant Environment Login Methods

Logging in as Local Users

MSP users logged in with their credentials will need to re-enter the same credentials to access their tenants, either through the dashboard or the tenant inventory.

Important: By default, the MSP admin has login access for all tenants.

Logging in as SSO Users

To login as SSO users, MSPs must first configure SSO, configure IDP with the service URL and then login. The steps are as follows:

Configure SSO

Configure the SSO feature within AppViewX's Platform module. Refer to the Single Sign On Configuration in the Platform guides.

Fetching the IDP Service URL

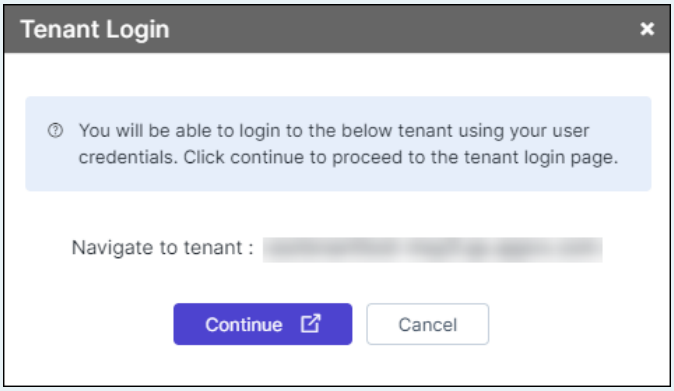
The **Fetch IDP Service URL** option is used to retrieve the Identity Provider (IDP) service URL. This URL is essential for integrating the tenant's authentication system with an external identity provider, enabling Single Sign-On (SSO).

To obtain the IDP service URL:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

1. From the **Tenant Management** inventory, for the required tenant, from the **Action** column, click  (**Fetch IDP Service URL**).

Note: You can also hover your mouse over the selected tenant to display the  icon. The **Fetch IDP Service URL** pop-up is displayed as shown below.



2. Click the 'Copy' icon next to the Service URL field and save it onto a notepad.
3. Add the copied URL in the IDP to utilize the SSO login features.

Important: AppViewX will match the user group details present in the ID token or SAML response with the existing user groups in AppViewX. Therefore, the IDP user group must be configured in both the MSP and the tenant to ensure that only authorized MSP users can access the MSP and its tenants.

Note:

- MSPs must fetch the IDP service URLs for each tenant separately and configure it in the IDP to use the SSO feature.
- SSO with Local authorization is not supported for tenant login.

Login to the Tenant

MSP users logged in with their SSO credentials can access their tenants, either through the dashboard or the tenant inventory without having to enter the credentials.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

SRE Access to MSP Portal

The SRE users will also have separate credentials to login to the MSP portal solely for the purpose of troubleshooting or verifying the issues if any and other functions such as:

- Create and manage clusters.
- Create and manage tags.
- Create and manage internal tenants.
- Configure internal tenant settings.
- Configure agent repo.
- Configure notification mail id.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Chapter 14: AppViewX Windows Gateway Setup

This guide outlines the steps for installing the AppViewX Windows Gateway for enabling communication between AppViewX and Windows. It also includes the steps for installing and using the AppViewX validator to validate the accessibility of the target machine on which the AppViewX Windows Gateway will be installed.

• AppViewX Windows Gateway
• Deployment Modes
• Communication Modes
• Setting up the AppViewX Windows Gateway
• Configuration Parameters
• Uninstalling the AppViewX Windows Gateway
• Updating AppViewX Windows Gateway
• Appendix A
• Troubleshooting
• FAQ: Installation and Configurations
• Quick References

AppViewX Windows Gateway

The AppViewX Windows Gateway Service is a Windows Communication Foundation (WCF) service designed to enable secure, seamless communication between AppViewX and the Windows server infrastructure. This service supports a variety of critical features tailored for managing and securing Windows-based environments. The key capabilities offered by AppViewX for Windows Server Infrastructure are as follows

1. **Certificate Life Cycle Management (CLM)** - AppViewX facilitates the complete lifecycle management of digital certificates across various environments, including:
   • Microsoft Servers
   • Microsoft PC’S
   • Microsoft IIS
   • Microsoft SQL Servers
   • Microsoft Certificate Authority (CA) Servers
   • IBM WebSphere
   • Windows WebLogic

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- JBoss
- Windows Apache Servers
- Windows Tomcat Servers

This includes certificate issuance, renewal, revocation, and validation, ensuring comprehensive certificate management and compliance.

> **Note:**
> - **Microsoft Server** - Both certificate store and file system certificates can be managed.
> - **Microsoft PC** - Only certificate store certificates can be managed.

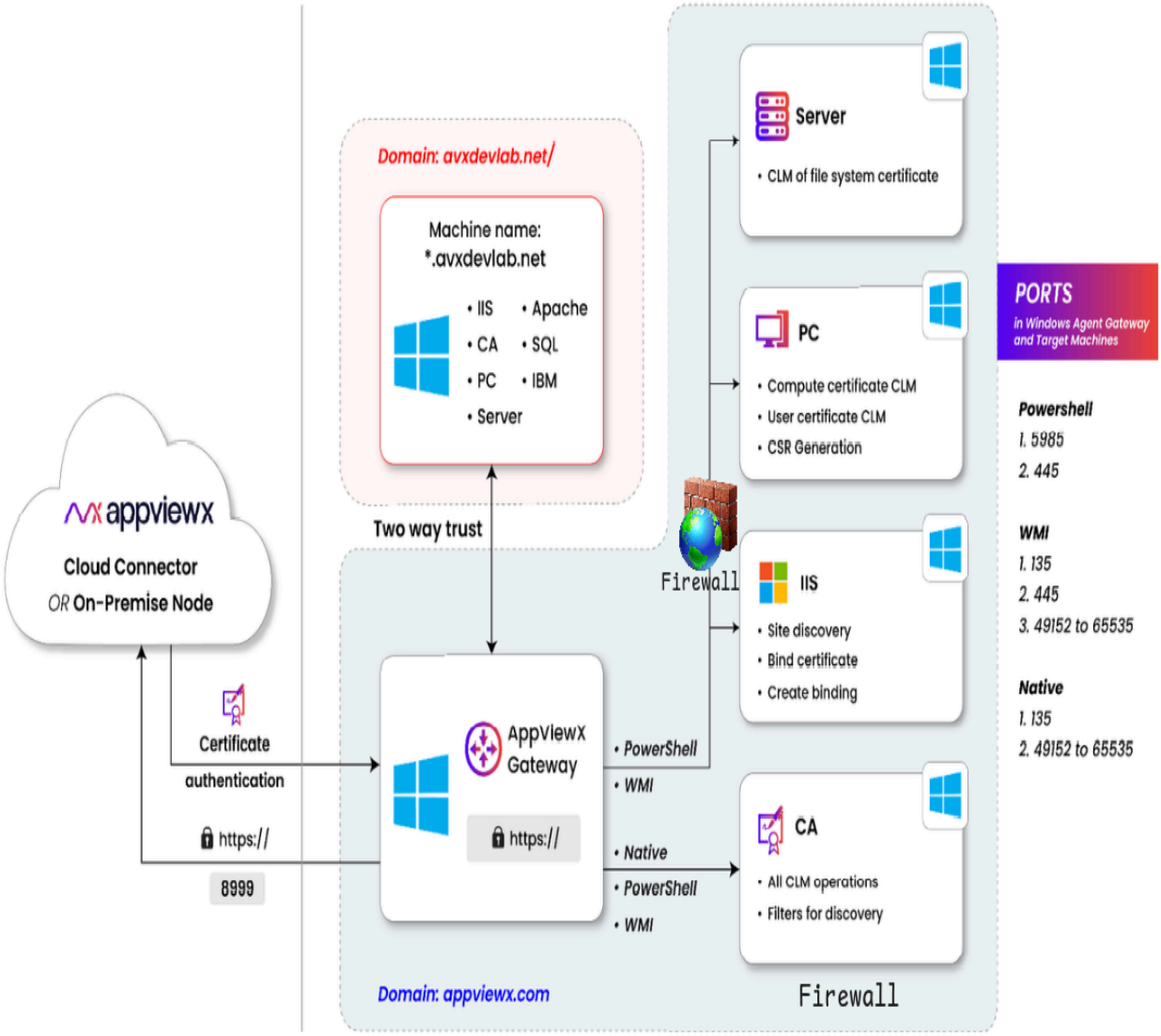
2. **Certificate Binding for IIS and SQL Server** - The service automates the binding of SSL/TLS certificates to only Internet Information Services (IIS) and SQL Server, ensuring secure communication channels for web applications and databases.
3. **Certificate Discovery** - AppViewX provides capabilities to discover certificates from multiple sources, including:
   - File System *(Only if onboarded as Microsoft Server)*
   - Windows Certificate Store

   This feature enables streamlined identification and management of certificates deployed across different locations within the server infrastructure.
4. **Custom Script Execution via PowerShell** - AppViewX allows for the execution of custom scripts, providing administrators with the flexibility to automate and customize certificate management and other related tasks. This integration with PowerShell enhances the extensibility and control of the service.

This document is designed to assist in diagnosing and resolving issues related to the communication between the **AppViewX Windows Gateway Service** and the Windows server infrastructure within the customer environment. This guide provides step-by-step troubleshooting procedures, frequently asked questions, and solutions to common problems, enabling administrators to quickly identify and address communication challenges.

The AppViewX Windows Gateway Architecture

Copyright © 2025 AppViewX Inc. All Rights Reserved.



**If firewall is configured the incoming for ports need to be open in target machine and outgoing in gateway machine.

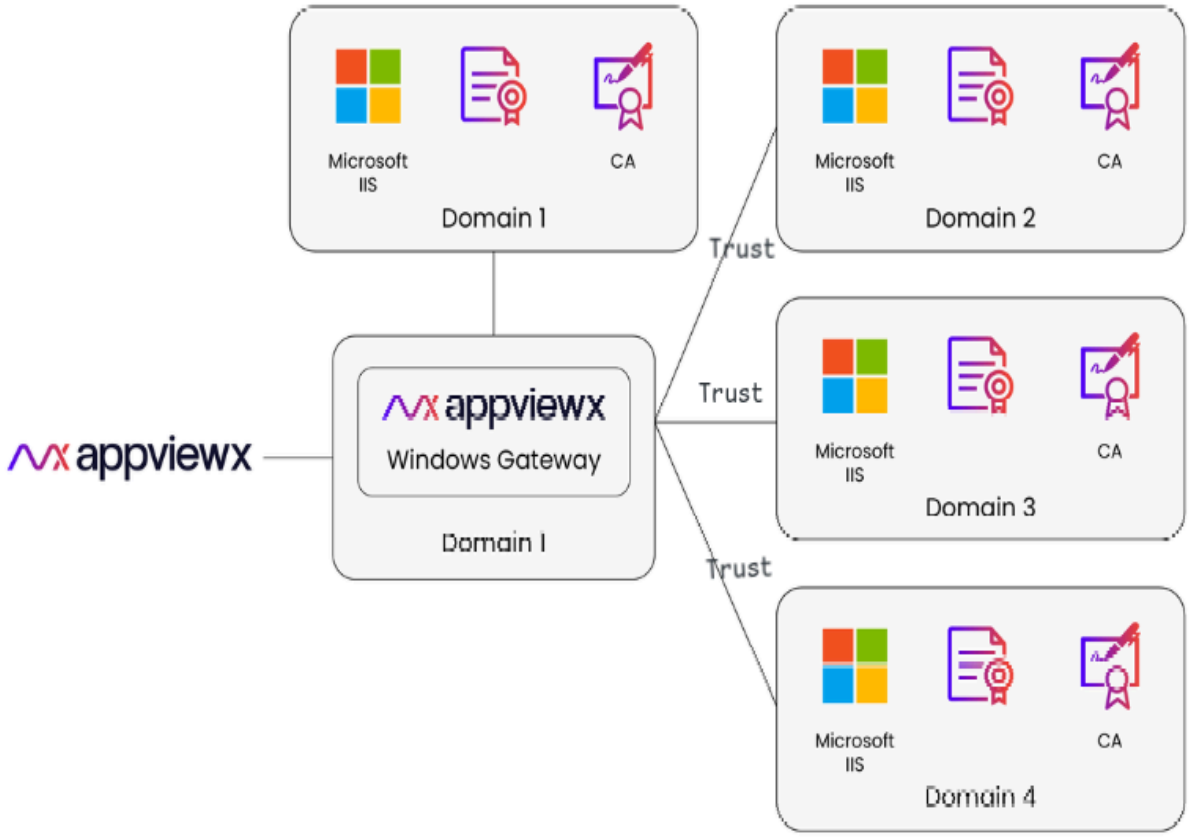
Deployment Modes

The deployment or installation of the **AppViewX Windows Gateway (AWG)** differs based on whether the domains in the environment are trusted or untrusted. The deployment scenarios for both trusted and untrusted domains are as follows:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

• Trusted Domains (2 way Trust)
• Independent Domains

Trusted Domains (2 way Trust)



- A single **AWG installation** manages infrastructure across all trusted domains.
- **AWG** securely communicates across domains, eliminating the need for separate installations.
- Centralizes control and simplifies configuration and management for all domains within the network.
- The steps to enable the two way trust between the domains are Windows Agent FAQ and Troubleshooting Guide (point 7)

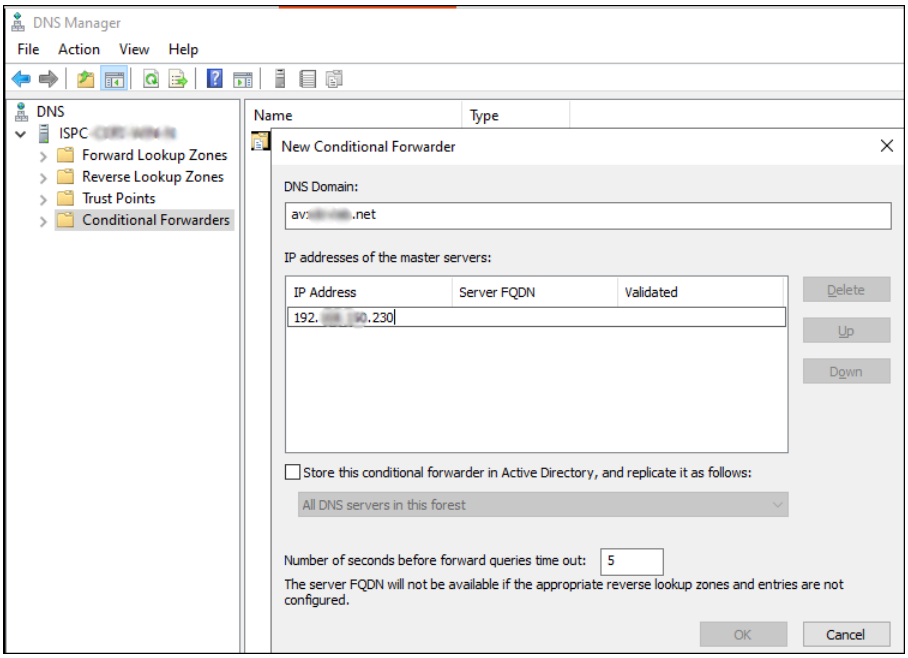
Copyright © 2025 AppViewX Inc. All Rights Reserved.

Enabling Two-way trust between Two Different Domains

Note:

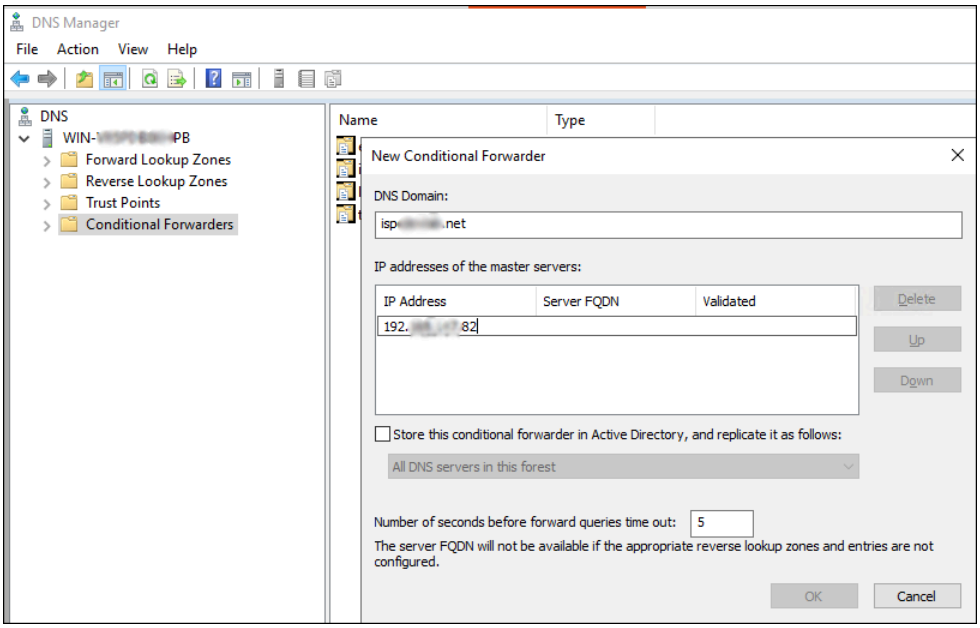
- Ensure that DNS settings on both servers are correctly configured and there is no firewall or network issue preventing communication between the two servers.
- You may need administrative privileges on both domain servers to complete the trust configuration.

1. Create a conditional forwarder on Domain Server 1
   a. Open the **DNS Manager** on the PDC of Domain Server 1
   b. Right-click on **Conditional Forwarders** and select **New Conditional Forwarder**.
   c. In the window that appears, enter the domain name of the other domain 2 and the IP address of the corresponding domain server.
   d. To save changes, click **OK**.

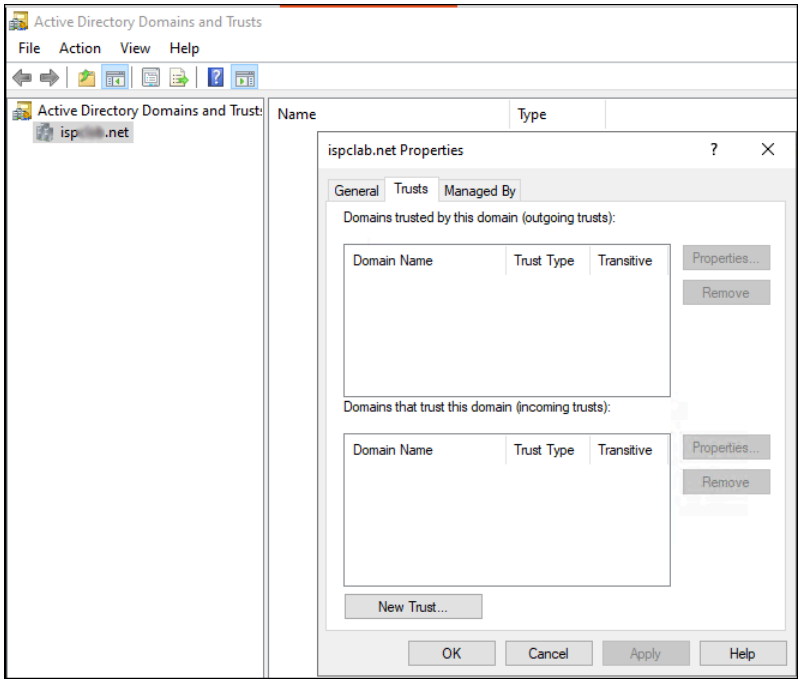


2. Create a conditional forwarder on Domain Server 2
   a. Open the **DNS Manager** on the PDC of Domain Server 2
   b. Right-click on **Conditional Forwarders** and select **New Conditional Forwarder**
   c. In the window that appears, enter the domain name of the other domain and the IP address of the corresponding domain server
   d. To save changes, click **OK**.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



3. Create a trust relationship between the two domain servers
   a. Open **Active Directory Domains and Trusts** on Domain Controller of Domain-1.
   b. Right-click the domain name and select **Properties**.



   c. Go to the **Trusts** tab and click **New Trust**.
   d. In the **New Trust Wizard**, do the following:
      i. Enter the Domain Name of Domain Server 2 and click **Next**.
      ii. Choose the **Forest trust** option and click **Next**.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

iii. Choose the **Two-way** trust option and click **Next**.

iv. Select **Both this domain and the specified domain** and click **Next**.

v. Provide the necessary credentials for the service account of Domain 2 and click **Next**.

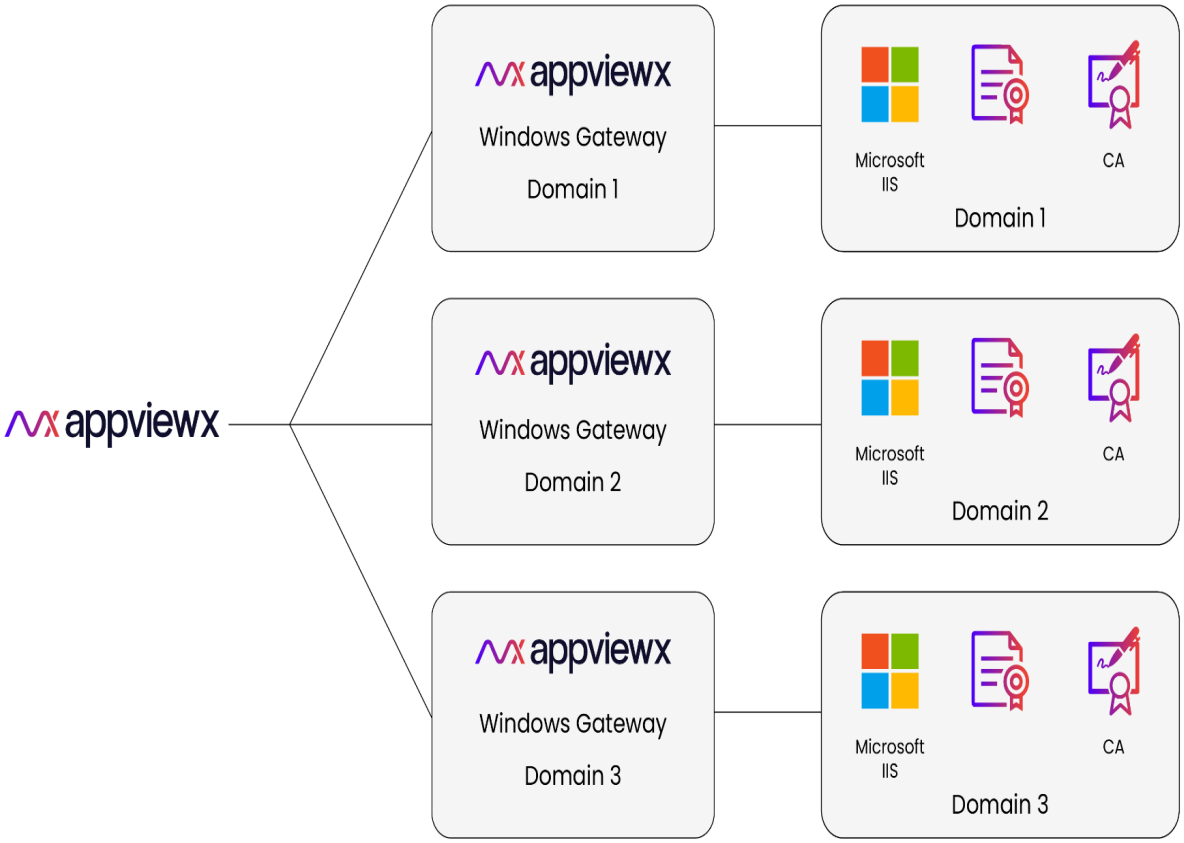
vi. Choose the **Forest-wide authentication** option and click **Next**.

vii. To complete the trust relationship setup, click **Finish**.

4. Verify the Trust Relationship.

   After setting up the trust, test the two-way communication between both domain servers to ensure proper name resolution and authentication are working correctly.

Independent Domains



- Each **untrusted domain** requires at least one separate **AWG installation**.
- **AWG** manages infrastructure independently for each domain.
- Ensures secure and isolated management per domain.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Communication Modes

The three modes of communication supported by Windows Agent to interact with any target machine are:

- Powershell
- WMI
- Native

Powershell

Overview

PowerShell Remoting is a built-in feature in Windows hosts that allows administrators to connect to remote hosts to execute scripts and PowerShell commands. Administrators need to enable PowerShell Remoting on the target machine for smooth communication. It is a powerful tool for efficiently and securely managing remote systems.

- In this mode, a powershell remoting session is created between the windows agent machine and the configured device. The powershell scripts are executed in session for different operations.
- This is the most modern and recommended mode of communication for performing different tasks.
- For more information on the pre-requisites for Powershell mode, refer

**Allow logon locally* is required if different accounts are being used in windows agent and target machine.The account used in target machine needs allow logon locally right to be assigned

Recommended usage

- This is the recommended mode as it is the most modern and supports all the use cases without any specific restrictions.
- WinRM is a more modern protocol used by many organizations.
- Ports 5985 are used in WinRM and WinRM v2.
- All WinRM data is encrypted using "Integrated Windows Authentication," preferably set to Kerberos authentication on the host machine.
- Essentially, WinRM is an HTTP-based API, and the data returned is in XML format rather than objects, as WinRM is XML-based.
- It is comparatively faster than WMI and Native Mode.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Permissions and Port Requirements

The table below summarizes the different permissions and port requirements in Powershell:

Requirement	AppViewX Windows Gateway	Target Server
User account type	Service account with local admin	Service account with local admin
Services	RPC Service WinRM Service WinRM Configuration Powershell remoting certutil.exe command availability	RPC Service WinRM Service WinRM Configuration Powershell remoting certutil.exe command availability
Ports	8999 (Customisable) 5985 (HTTP) or 5986 (HTTPS) - WSMAN 445 - SMB (needed only in case of push functionality)	5985 (HTTP) or 5986 (HTTPS) - WSMAN 445 - SMB (needed only in case of push functionality)

Prerequsisites for Powershell Mode

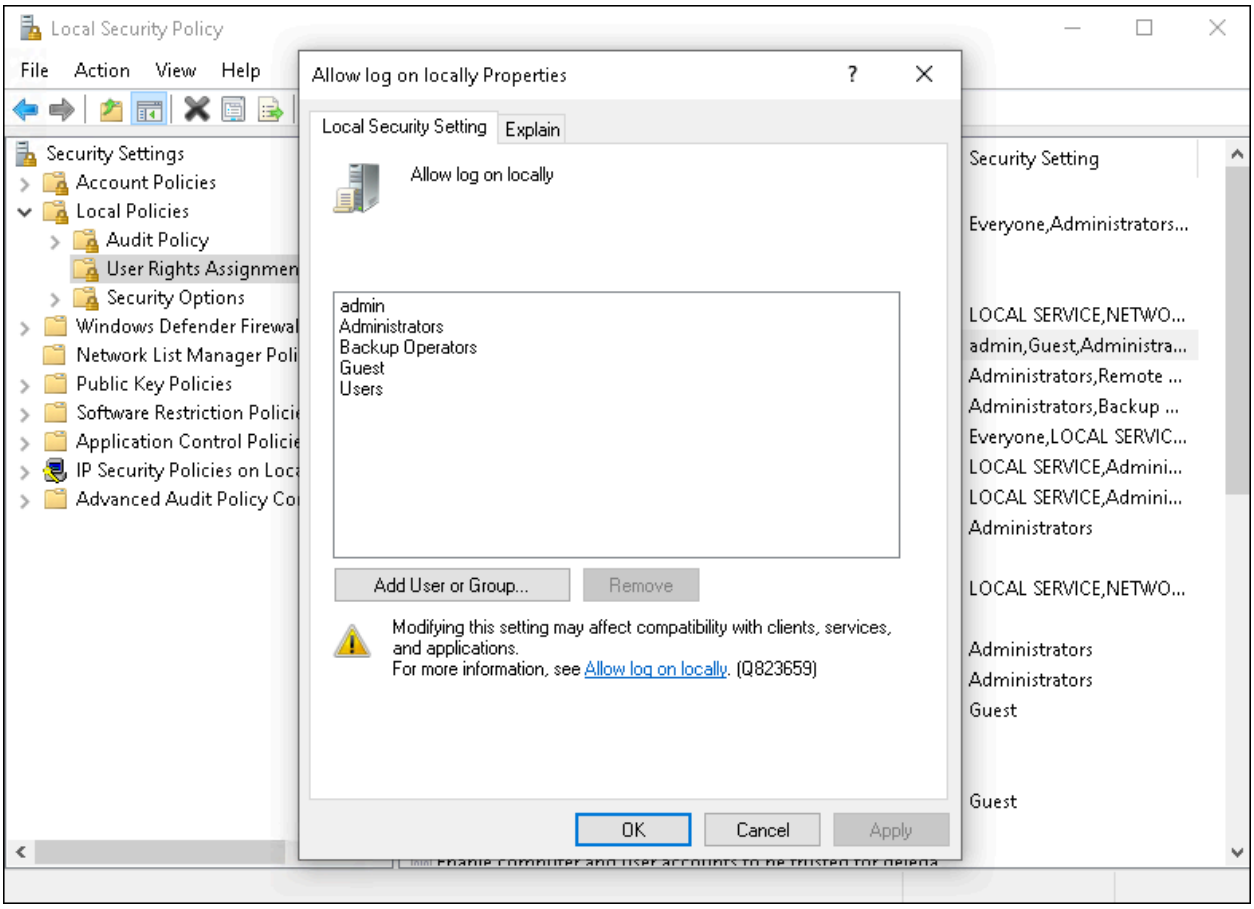
To ensure successful management of Windows servers in **PowerShell mode**, the following prerequisites must be met:

Local Administrator Account on Target Machines

The account used for onboarding the device must be a member of the **Local Administrators Group** on the target machine. Administrative privileges are required for enabling remote PowerShell sessions and managing the device remotely.

Note: If the onboarding account is not part of the local administrators group on the Windows agent, the account must be granted the **Allow Logon Locally** right via local security policies on the Windows agent machine.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

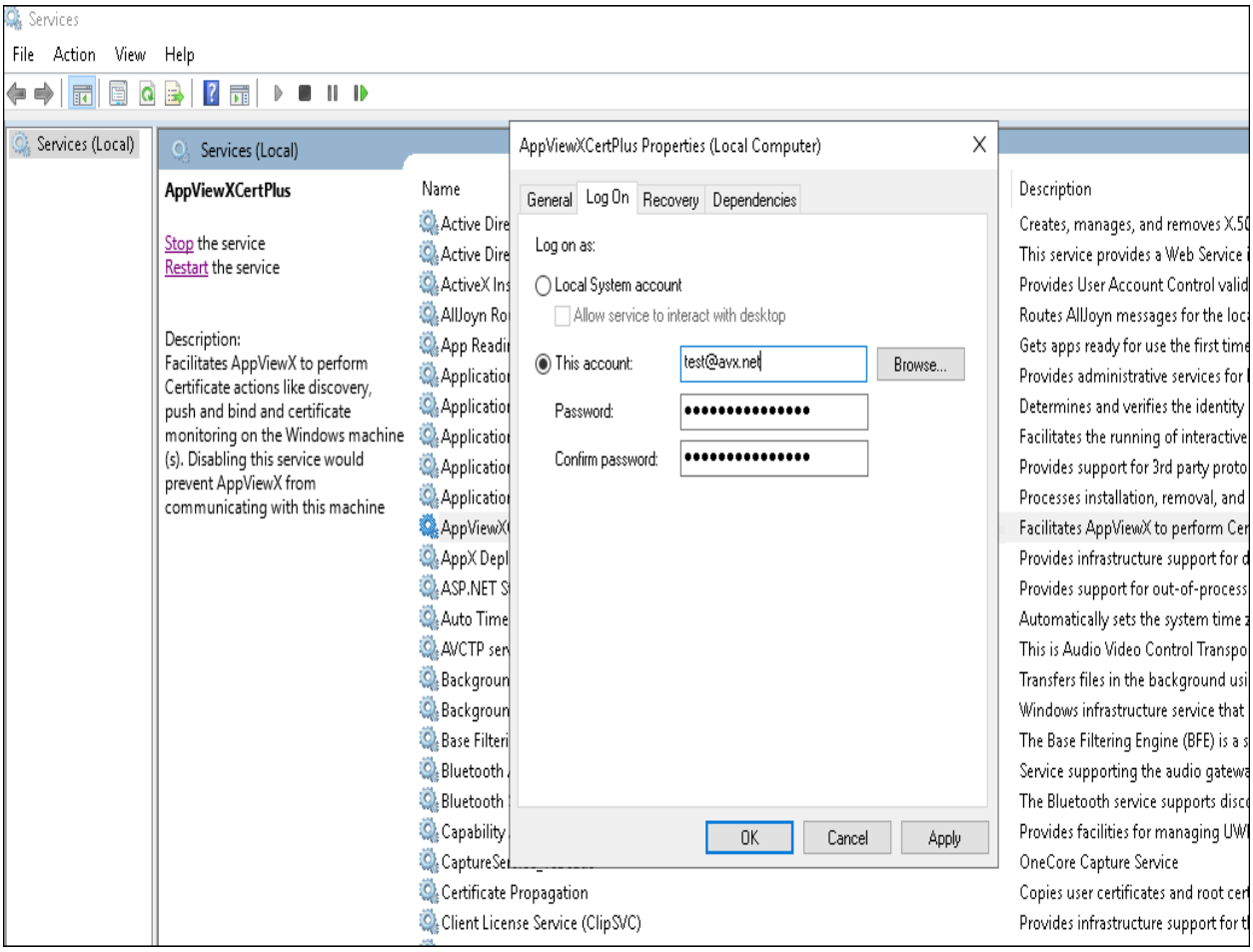


For more details, refer to the Microsoft documentation: about_Remote_Requirements - PowerShell | Microsoft Learn

Logon Service Account on the Windows Agent Machine

The Windows agent machine must have a logon service account with local administrator privileges to run the required services.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



Network Ports

The following ports must be open to facilitate communication between the target machines, the Windows agent machine, and the CC:

- **Port 8999 (default):** Open on the Windows agent machine for communication with the Cloud connector or AppViewX.
- **Port 5985 (HTTP) or 5986 (HTTPS):** Required for Windows Remote Management (WinRM) to enable remote PowerShell.
- **Port 445 (SMB):** Used for certificate file transfer during push/bind operations to a target server via SMB protocol.

Note: The port configuration is customizable, allowing for the use of alternative ports. It is recommended to select a non-reserved port and ensure that no other services are currently using the designated port to avoid conflicts.

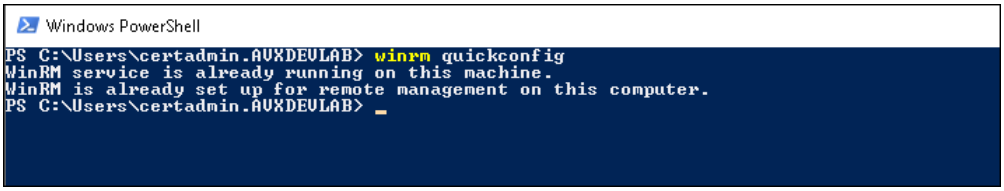
Copyright © 2025 AppViewX Inc. All Rights Reserved.

Enable Remote Powershell in the Target Machines

To ensure that remote powershell is enabled in the target machine, execute the command:

```
winrm quickconfig
```

Sample output:



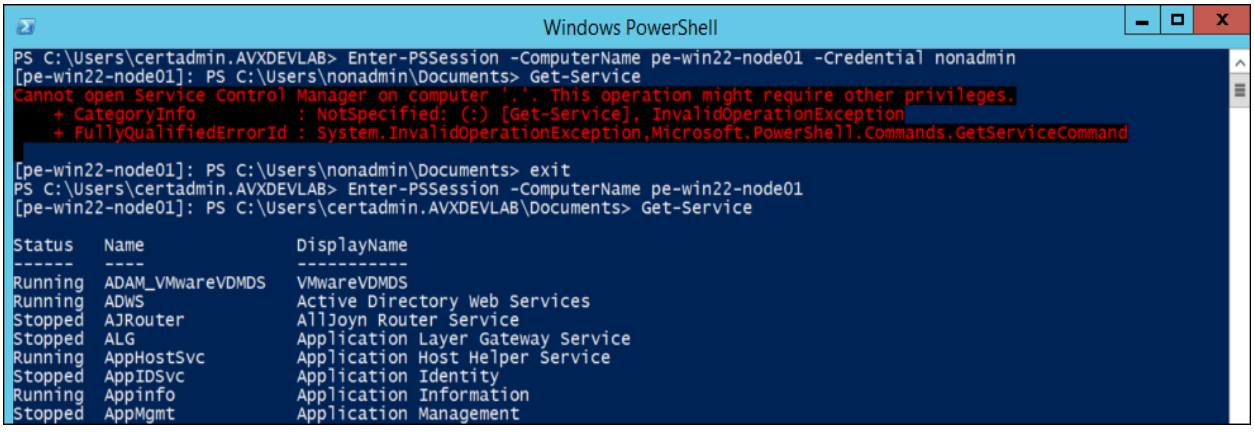
Kerberos Authentication

- Ensure all accounts involved are within the same domain to enable the use of the Kerberos authentication mechanism for remote PowerShell connections.
- Validate the Kerberos setup between the target machines and the Windows agent machine to ensure proper authentication and communication.

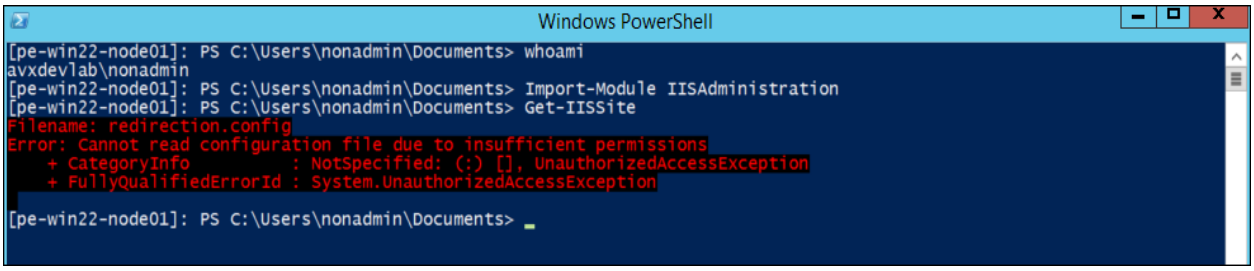
Justifying Admin Access

To create remote sessions and run remote commands, the current user must, by default, be a member of the Administrators group on the remote computer or provide administrator credentials. Otherwise, the command will fail.

PowerShell remoting can be enabled for a standard user, but they need administrative privileges to manage IIS and the Local Computer Store. In the screenshot, a non-admin user faces privilege restrictions when listing system services, whereas an admin user can view the services without issue.



Copyright © 2025 AppViewX Inc. All Rights Reserved.



Certificate Store Access

- **Admin Privileges for Certificate Management:** Managing certificates in system-wide locations, such as the Local Machine certificate store, requires admin privileges. This includes tasks like installing, renewing, revoking, and deleting certificates.
- **Service Configuration:** Administrative access is necessary for configuring services to use specific system certificates for secure communication. This includes modifying service configurations, such as those for SQL Server.
- **Private Key Management:** Admin privileges are needed to manage private keys associated with certificates. This includes importing/exporting certificates with their private keys, configuring key access permissions, and configuring key archival and recovery policies.
- **MS Certificate Authority (CA) Operations:** Tasks related to managing a Certificate Authority (CA), such as configuring CA settings, issuing and revoking certificates, and managing certificate templates, often require admin privileges.
- **System Integrity:** Certificate management is crucial for system integrity and security. Admin privileges ensure that only authorized users can manage certificates, reducing the risk of unauthorized access, tampering, or misuse.
- **Security:** PowerShell remoting allows users to execute commands and scripts on remote computers. Admin access ensures that only users with sufficient privileges can perform potentially sensitive or impactful actions on remote systems.
- **System Management:** Many administrative tasks, such as IIS administration and accessing certain registry keys, require administrative privileges. PowerShell remoting enables administrators to perform these tasks remotely, but admin access is necessary to execute the required commands successfully.
- **Resource Control:** Admin access ensures that users have the necessary permissions to access and modify system resources, such as files, directories, and registry keys, on remote machines. This level of access is often required for effective system management and troubleshooting.
- **Configuration Management:** PowerShell remoting is commonly used in configuration management and automation scenarios. Administrators need admin access to deploy configurations, install updates, and perform other management tasks remotely.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

To learn about the system and configuration requirements for running remote commands in PowerShell, refer to the "about Remote Requirements" section on the Microsoft documentation website.

WMI

Overview

The WMI infrastructure is a Microsoft Windows operating system component known as the WMI service (winmgmt). The ability to obtain management data from remote computers is what makes WMI useful. Remote WMI connections are made through DCOM.

- In this mode first the powershell script is copied from the windows agent server to the configured device machine. Powershell script is executed in a configured device with use of WMI command.
- User Account Control (UAC) affects the WMI data that is returned from a command-line tool, remote access, and how scripts must run. For more details refer to User Account Control and WMI - Win32 apps | Microsoft Learn.
- In WMI mode, AppViewX generates PowerShell scripts that are executed on the target machine. This process necessitates special access to the drive of the end machine in order to execute the scripts effectively.
- The WMI command used to execute the script itself necessitates admin privileges.

**Allow logon locally* is required if different accounts are being used in windows agent and target machine.The account used in target machine needs allow logon locally right to be assigned

Recommended Usage

- This mode is relatively older and slower when compared with Powershell mode.
- This mode is generally chosen only when remote execution privileges (Powershell remoting permissions are disabled) are not given to the user account in the target machine.
- WMI is enabled by default on many Windows servers.
- DCOM remains integrated into the Windows OS and is used by the Windows services to communicate, such as Microsoft Management Console certificate store.
- Organizations that prefer not to use PowerShell remoting or WinRM can use DCOM (WMI) as a communication method.
- It is commonly used in older Windows servers, such as Windows Server 2012 or 2008 R2.

Standard remote WMI queries use RPC to connect and RPC in turn uses a mess of ports. Initially, the Collector connects to the remote system via TCP port 135. The remote system then selects a high port

Copyright © 2025 AppViewX Inc. All Rights Reserved.

and instructs the Collector to use this new port for subsequent communications. The high port depends on the OS but the current Windows OS uses ports 49152 to 65535.

PORTS USED: 445, 135 + dynamic port: 49152-65534

To use a static DCOM port for WMI in Windows, instead of numerous high ports, please follow the instructions on the Microsoft site for allocating a static port for WMI communication. Refer to Setting Up a Fixed Port for WMI | Microsoft Learn.

Permissions and Port Requirements

The table below summarizes the different permissions and port requirements in WMI:

Requirement	AppViewX Windows Gateway	Target Server
User account type	Service account with local admin	Service account with local admin
Services	WMI Service certutil.exe command availability	WMI Service certutil.exe command availability
Ports	8999 (Customisable) 445 - SMB 135 RPC Port 49152-65535 - Dynamic Ports	445 - SMB 135 RPC Port 49152-65535 - Dynamic Ports

Native

Overview

The Native API mode is only used at Microsoft CA communication. It uses the RPC based protocol for communication and sends a DCOM message.

- In this mode, communication between the Windows Agent Machine and the target machine is through RPC and DCOM.
- This mode requires permission which is applicable for WMI, since it uses the same communication method.
- The difference between WMI and native mode is native mode uses, the native sdk provided by Microsoft for managing the CA. It doesn’t require drive access.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

PORTS USED: 135, 145

Recommended Usage

- This is one of the oldest communication protocols used in Windows.
- This mode supports only CA related operations when using AppViewX Windows Gateway.
- The customer retains control of the credentials since they enter them directly during local logon. Additionally, they have the option to use a managed service account with this approach.
- Because the libraries are built by Microsoft, communication occurs more quickly in the native mode.

Permissions and Ports

The table below summarizes the different permissions and port requirements in Native:

Requirement	AppViewX Windows Gateway	Target Server
User account type	Service account with local admin	Service account with local admin
Services	WMI Service certutil.exe command availability	WMI Service certutil.exe command availability
Ports	8999 (Customisable) 135 - RPC Port 49152-65535 - Dynamic Ports	135 - RPC Port 49152-65535 - Dynamic Ports

Setting up the AppViewX Windows Gateway

- Step 1: Prerequisites for Installing the AppViewX Windows Gateway
- Step 2: Downloading the AppViewX Windows Gateway Installer
- Step 3: Installing the AppviewX Windows Gateway
- Step 4: Verifying the AppviewX Windows Gateway Installation
- Step 5: Managing a Target Server
- Step 6: Disabling Current Operating System Information
- Installation using Non-Admin Service Account

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Step 1: Prerequisites for Installing the AppViewX Windows Gateway

The **AppViewX Windows Gateway** service is a Windows Communication Foundation (WCF) service that facilitates secure communication between AppViewX and the Windows server infrastructure.

Software Prerequisites

Name	Description
Operating System	• Windows Server 2012 • Windows Server 2012 R2 • Windows Server 2016 • Windows Server 2019 • Windows Server 2022 • Windows 10 • Windows 11 Refer to the following articles: • Windows Server 2012 and 2012 R2 reaching end of support - Microsoft Lifecycle • Windows 8.1 Support End of Life • Windows 7 - Microsoft Lifecycle
.NET framework	.NET framework version 4.5.2 is required.
Admin access	Administrator privilege is needed to install AppViewX Windows Gateway.
PowerShell	PowerShell version 4.0 is needed. (Refer to Prerequsisites for Powershell Mode for mode details.)

Hardware Prerequisites

Hardware prerequisites

Hardware	Capability
RAM	8 GB
HDD	10 GB

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Hardware prerequisites (continued)

Hardware	Capability
CPU	Intel or AMD processor with 64-bit support, 1.8 GHz or faster processor

Firewall Prerequisites

Component	Port
Default Port communication from AppViewX to a AppViewX Windows Gateway	8999

Refer to the **Permissions and Ports** (Powershell, WMI, Native) section for more details.

Note:

- The firewall must not block the following port and the respective port must open on the Agent.
- During the installation of AppViewX Windows Gateway, the default port can be reconfigured. For more details refer Step 3 of the installation process.

Additional Prerequisites

Refer to the General Prerequsistes section for additional prerequisites that can be validated manually or by the AppViewX Windows Gateway Troubleshooting tool provided with the AppViewX Windows Gateway.

Step 2: Downloading the AppViewX Windows Gateway Installer

Download and unarchive the **AppViewX.CertPlus.Installer.zip** file from the release portal, under the latest version of AppViewX, in the **Additional Resources** section. The download package consists of the following files:

File Name	Description
AppViewX.CertPlus.Installer.exe	Installer executable
ClientCertificateGateway.pfx	Default client certificate
ServerCertificateGateway.pfx	Default server certificate

Copyright © 2025 AppViewX Inc. All Rights Reserved.

File Name	Description
config.xml	Application configuration settings that will override the default settings after the AppViewX Windows Gateway is installed.
Readme.txt	Help file with details of the AppViewX Windows Gateway.
InstallationLog.txt	Logs the success and error messages from the installation process.

Step 3: Installing the AppviewX Windows Gateway

Before you begin: By default, the AppViewX Windows Gateway securely communicates with AppViewX using the server/client certificates that are shipped along with the AppViewX Windows Gateway installer. If you choose to use a different server and client certificate for authentication, then follow the steps below:

1. From Windows explorer, browse to the location where you have unarchived the AppViewX Windows Gateway installer package.
2. Rename the default server certificate **ServerCertificateGateway.pfx** to **ServerCertificateGateway-Backup.pfx** and the client certificate file **ClientCertificateGateway.pfx** to **ClientCertificateGateway-Backup.pfx**.
3. Copy the server and client certificates that you intend to use in this directory.
4. Rename the server certificate file to **ServerCertificateGateway.pfx** and the client certificate file to **ClientCertificateGateway.pfx**, and then replace the default certificates in the installation folder.

Note: While installing the AppViewX Windows Gateway, you will be prompted to provide the server and client passwords.

CAUTION: If the certificate is replaced, ensure that the respective password has been provided to add the certificate to the store. The incorrect password during the installation of AppViewX Windows Gateway will cause the Windows Agent installation to fail.

1. Execute the **AppViewX.CertPlus.Installer.exe** file.
   The welcome screen for the setup wizard is displayed.
2. Click **Next**.
   The **License Agreement** is displayed.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

3. Select **I accept the terms in the license agreement**.
4. Click **Next**.
   The **Destination Folder** screen is displayed.
5. To install the AppViewX Windows Gateway at the default location, click **Next**.
   OR

   To change the default destination folder:

   a. Click **Change**.

   b. On the **Change Current Destination Folder** screen, use the **Look in** dropdown list/  (up one level) icon/  (create new folder) icon to navigate to/create the required destination folder.

   c. On the **Change Current Destination Folder** screen, use the **Look in** dropdown list/  (up one level) icon/  (create new folder) icon to navigate to/create the required destination folder.

   d. Click **OK**.

   e. On the **Destination Folder** screen, click **Next**.
      The **Optionally you can modify the below details** screen is displayed.

   f. Enter the details as required.

   Field descriptions for the details

Field	Description
Please select default certificate store	Select the certificate store from which the certificates will be discovered and pushed to by AppViewX from the following options: • Current User Store This type of certificate store is local to a user account on a computer. It is located in the registry under the HKEY_CURRENT_USER root. • Local Machine Store (default) This type of certificate store is local to a computer and global to all the user accounts

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Field	Description
	on the computer. It is located in the registry under the HKEY_LOCAL_MACHINE root. This configures the gateway for communicating with the appropriate certificate store.
Port	Port for accessing the service. Default value: 8999 (can be modified if required)
Server certificate thumbprint	If you are using a custom certificate, enter the corresponding server certificate thumbprint value.
Client certificate password	Password for accessing the client certificate For custom client certificates, enter the certificate password.
Server certificate password	Password for accessing the server certificate For custom server certificates, enter the certificate password.



Note: Refer to the **Before you Begin** section to use custom server and client certificates.

6. Click **Next**.
   The **Ready to Install the Program** screen is displayed.
7. Click **Install**.
   This will:
   - Install the AppViewX Windows Gateway Troubleshooter tool
   - AppViewX Windows Gateway service

• Navigating through the Installation

Navigating through the Installation

Copyright © 2025 AppViewX Inc. All Rights Reserved.

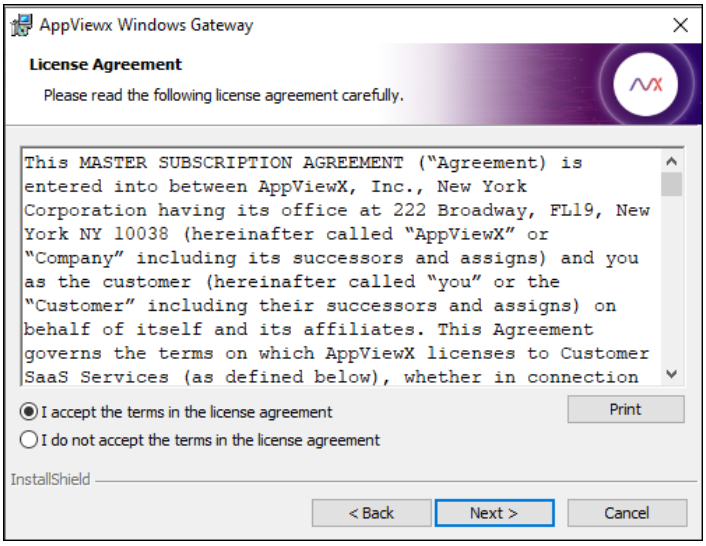
1. Execute the **AppViewX.CertPlus.Installer.exe** file.

   The following welcome screen for the setup wizard is displayed.



2. Click **Next**.

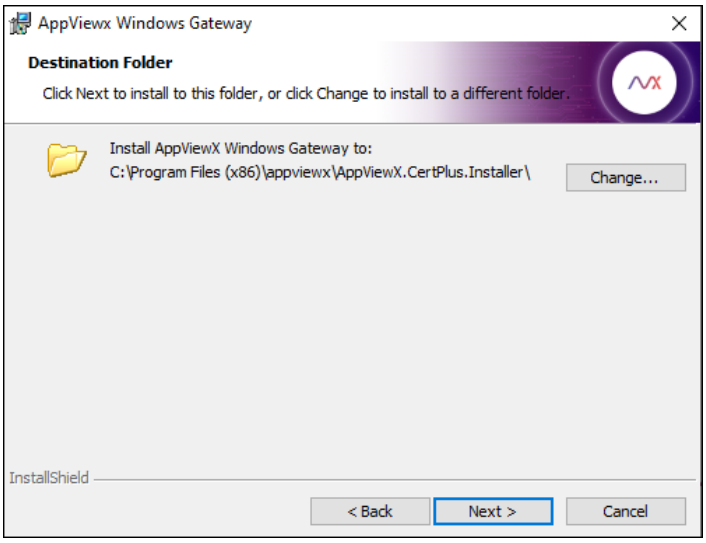
   The **License Agreement** is displayed.



3. Select **I accept the terms in the license agreement**.
4. Click **Next**.

   The **Destination Folder** screen is displayed.

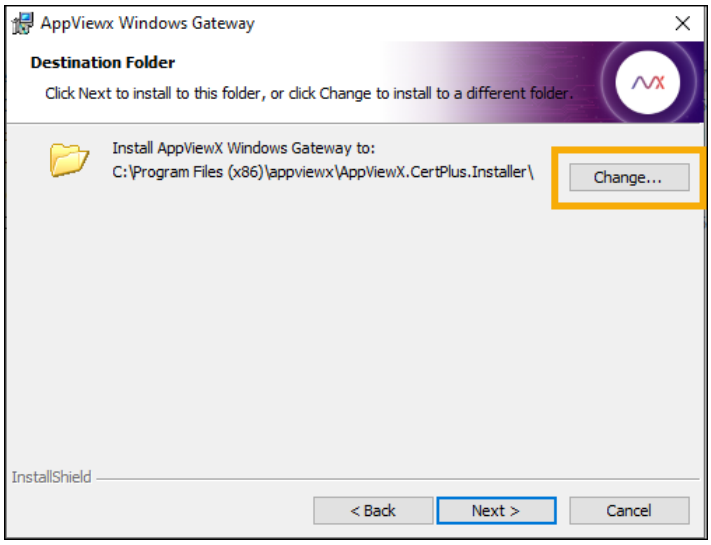
Copyright © 2025 AppViewX Inc. All Rights Reserved.



5. To install the AppViewX Windows Gateway at the default location, click **Next**.

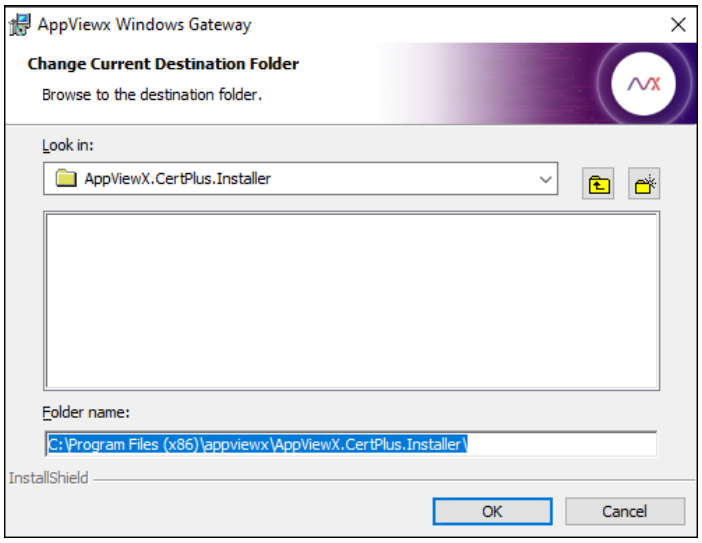
   To change the default destination folder:

   a. Click **Change**.



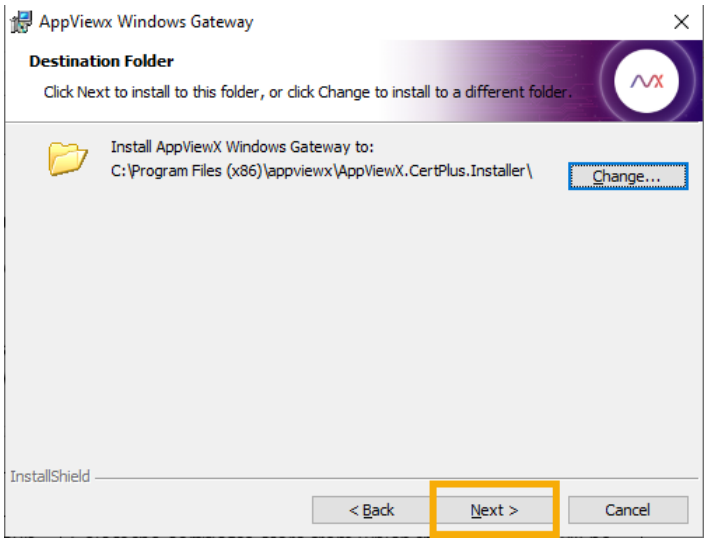
   b. On the **Change Current Destination Folder** screen, use the **Look in** dropdown list/  (up one level) icon/  (create new folder) icon to navigate to/create the required destination folder.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



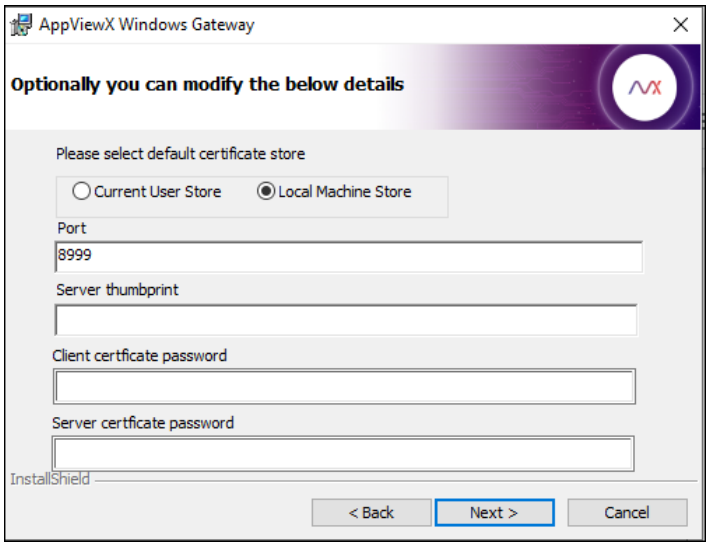
c. Click **OK**.

d. On the **Destination Folder** screen, click **Next**.



The **Optionally you can modify the below details** screen is displayed.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



Enter the following details (optional):

Field	Description
Please select default certificate store	Select the certificate store from which the certificates will be discovered and pushed to by AppViewX from the following options: • Current User Store This type of certificate store is local to a user account on a computer. It is located in the registry under the HKEY_CURRENT_USER root. • Local Machine Store (default) This type of certificate store is local to a computer and global to all the user accounts on the computer. It is located in the registry under the HKEY_LOCAL_MACHINE root. This configures the gateway for communicating with the appropriate certificate store.
Port	Port for accessing the service.

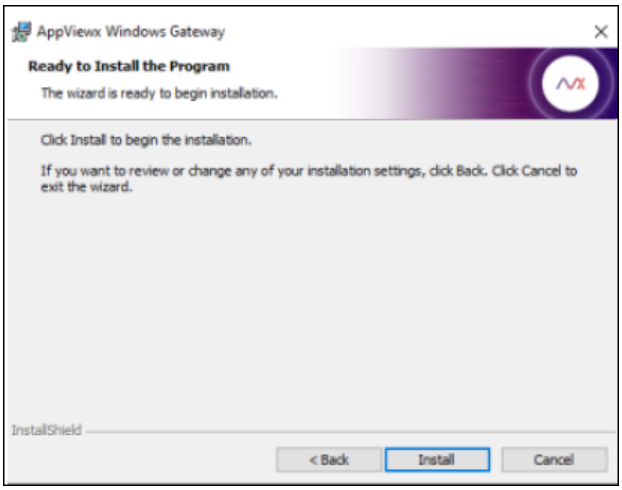
Copyright © 2025 AppViewX Inc. All Rights Reserved.

Field	Description
	Default value: 8999 (can be modified if required)
Server certificate thumbprint	If you are using a custom certificate, enter the corresponding server certificate thumbprint value.
Client certificate password	Password for accessing the client certificate For custom client certificates, enter the certificate password.
Server certificate password	Password for accessing the server certificate For custom server certificates, enter the certificate password.

Note: Read the **Before you Begin** section to use custom server and client certificates.

6. Click **Next**.

   The Ready to Install the Program screen is displayed.



7. Click **Install**.

   This will:

   • Install the AppViewX Windows Gateway Troubleshooter tool

   • AppViewX Windows Gateway service.

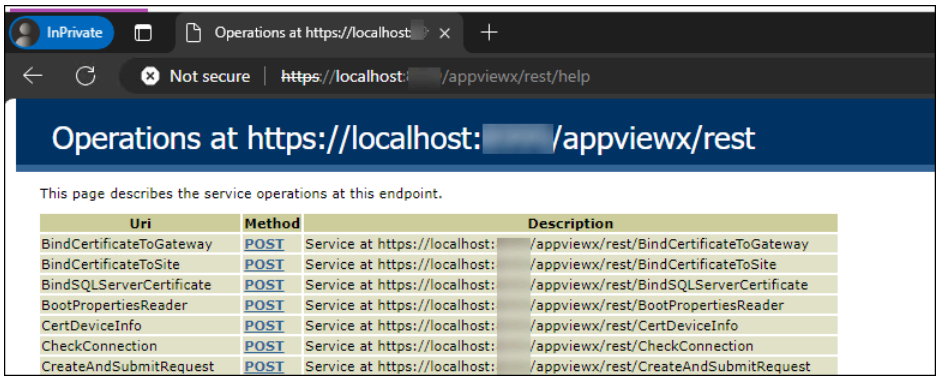
Copyright © 2025 AppViewX Inc. All Rights Reserved.

Step 4: Verifying the AppviewX Windows Gateway Installation

1. Open **Services.msc** and ensure *AppViewXCertPlus* service is in running state.
2. Perform a Verification Test as follows:
   a. Access the following URL in an **Incognito** or **Private Browsing** window: *https://localhost:8999/appviewx/rest/help*.
   b. Upon accessing the URL, a dialog box will appear requesting a certificate to be chosen.
   c. Please select the AppViewX issued client certificate from the Installer package or choose the required certificate if any custom certificate is used for authentication.

> **Note:** The same custom certificate has to be uploaded in the AppViewX CERT+ Application when configuring the Windows Gateway

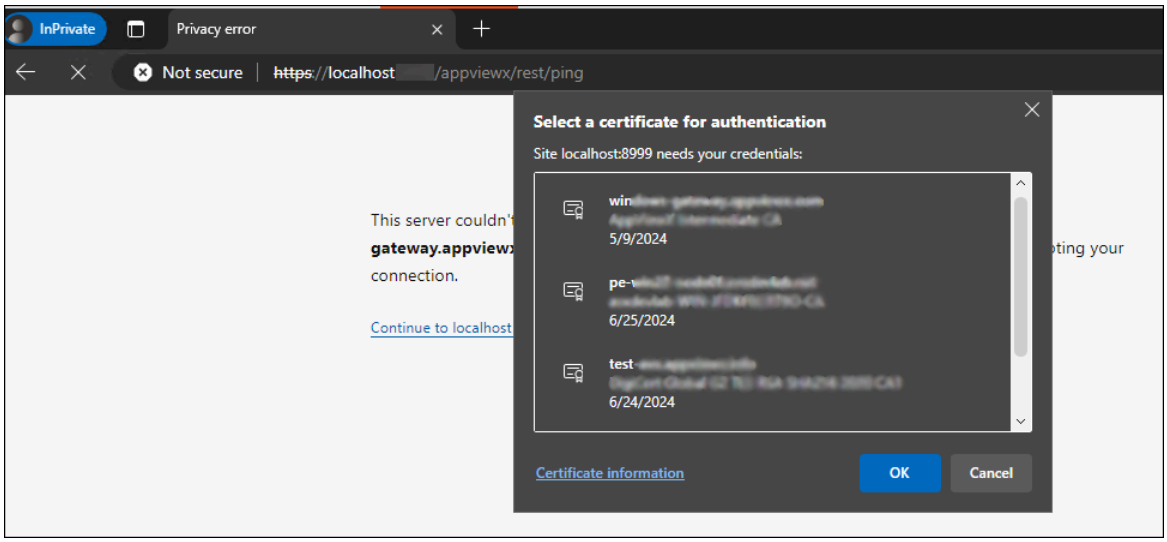
Sample output:



   d. The browser should then display the available services as a response.

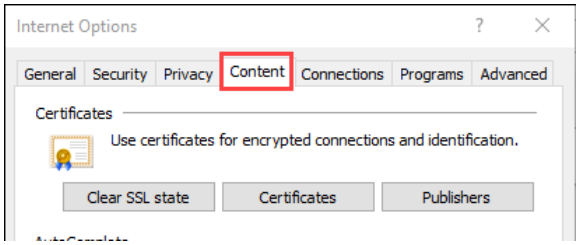
Copyright © 2025 AppViewX Inc. All Rights Reserved.

Sample output



3. To verify the Windows AppViewX Gateway installation on Internet Explorer, import the client authentication certificate **ClientCertificateGateway.pfx**, from the download package (password: **appviewx**).

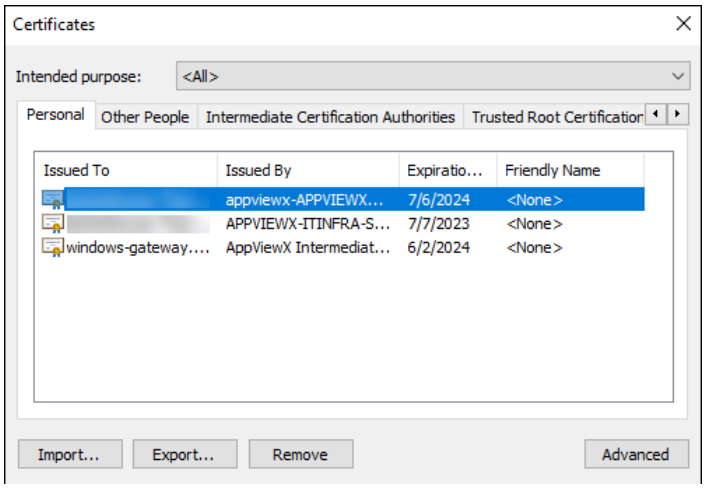
   a. Navigate to Internet Explorer's **Settings** > **Internet Options**, and then click the **Content** tab.



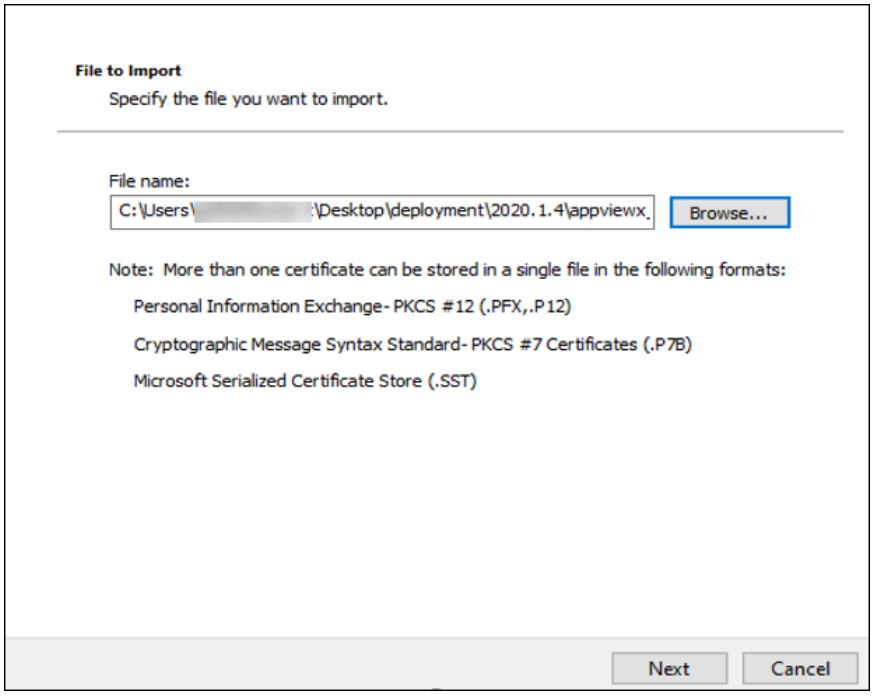
   b. Click the **Certificates** button.

   The **Certificates** popup window opens.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



c. Click the **Import** button on the **Certificates** page.



d. Go to the URL in the format: **https://hostname:portnumber/appviewx/rest/help**. For example: https://10.10.10.10:8999/appviewx/rest/help

The page displayed confirms the accessibility and installation of the service.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



Operations at https://localhost:8999/appviewx/rest

This page describes the service operations at this endpoint.

Uri	Method	Description
BindCertificateToGateway	POST	Service at https://localhost:8999/appviewx/rest/BindCertificateToGateway
BindCertificateToSite	POST	Service at https://localhost:8999/appviewx/rest/BindCertificateToSite
BindSQLServerCertificate	POST	Service at https://localhost:8999/appviewx/rest/BindSQLServerCertificate
BootPropertiesReader	POST	Service at https://localhost:8999/appviewx/rest/BootPropertiesReader
CertDeviceInfo	POST	Service at https://localhost:8999/appviewx/rest/CertDeviceInfo
CheckConnection	POST	Service at https://localhost:8999/appviewx/rest/CheckConnection
CreateAndSubmitRequest	POST	Service at https://localhost:8999/appviewx/rest/CreateAndSubmitRequest
CreateCSR	POST	Service at https://localhost:8999/appviewx/rest/CreateCSR
CreateCSRKey	POST	Service at https://localhost:8999/appviewx/rest/CreateCSRKey
DeleteFile	POST	Service at https://localhost:8999/appviewx/rest/DeleteFile
DeleteKeys	POST	Service at https://localhost:8999/appviewx/rest/DeleteKeys
DeviceInfo	POST	Service at https://localhost:8999/appviewx/rest/DeviceInfo
DiscoverCertificates	POST	Service at https://localhost:8999/appviewx/rest/DiscoverCertificates
DiscoverCertStoreCertificates	POST	Service at https://localhost:8999/appviewx/rest/DiscoverCertStoreCertificates
DiscoverFileCertificates	POST	Service at https://localhost:8999/appviewx/rest/DiscoverFileCertificates
DiscoverIBM	POST	Service at https://localhost:8999/appviewx/rest/DiscoverIBM
DiscoverKeys	POST	Service at https://localhost:8999/appviewx/rest/DiscoverKeys
ExecuteScriptInPowershell	POST	Service at https://localhost:8999/appviewx/rest/ExecuteScriptInPowershell
ExecuteWLSTScript	POST	Service at https://localhost:8999/appviewx/rest/ExecuteWLSTScript
ExtractCertificate	POST	Service at https://localhost:8999/appviewx/rest/ExtractCertificate
GetCertStores	POST	Service at https://localhost:8999/appviewx/rest/GetCertStores
LatestLog	POST	Service at https://localhost:8999/appviewx/rest/LatestLog
MicrosoftCAs	POST	Service at https://localhost:8999/appviewx/rest/MicrosoftCAs
MqConnector	POST	Service at https://localhost:8999/appviewx/rest/MqConnector
Ping	GET	Service at https://localhost:8999/appviewx/rest/Ping
PushAndBindCertificate	POST	Service at https://localhost:8999/appviewx/rest/PushAndBindCertificate
PushCertificate	POST	Service at https://localhost:8999/appviewx/rest/PushCertificate
PushDiscoveredCertificates	POST	Service at https://localhost:8999/appviewx/rest/PushDiscoveredCertificates
ReadFile	POST	Service at https://localhost:8999/appviewx/rest/ReadFile
ReadMultipleFiles	POST	Service at https://localhost:8999/appviewx/rest/ReadMultipleFiles
RemoveCertificateFromStore	POST	Service at https://localhost:8999/appviewx/rest/RemoveCertificateFromStore
RemoveSiteBinding	POST	Service at https://localhost:8999/appviewx/rest/RemoveSiteBinding
RevokeCertificate	POST	Service at https://localhost:8999/appviewx/rest/RevokeCertificate
SaveKeys	POST	Service at https://localhost:8999/appviewx/rest/SaveKeys

Note: In the event that a custom client authentication certificate is used, ensure that the CRL mentioned in the certificate is reachable from the AppViewX Windows Gateway hosting server.

Note: The steps to import the client certificate will differ depending on the web browser.

e. To register the AppViewX Windows Gateway with AppViewX, navigate to the AppViewX Cert+ (on the SaaS deployment) admin UI/UX, and then **Settings** > **Certificate**.

Note: To add the AppViewX Windows Gateway for

- Microsoft Enterprise CA integration, see Microsoft Enterprise CA.
- Microsoft Standalone CA integration, see Microsoft Standalone CA
- Microsoft Device integrations, refer to the following sections:
  - Microsoft IIS
  - Microsoft PC
  - Microsoft Server
  - Microsoft SQL

Copyright © 2025 AppViewX Inc. All Rights Reserved.

f. Register the gateway using the following URL format: **https://hostname:portnumber/appviewx**. For example: https://10.10.10.10:8999/appviewx

Note:

- The AppViewX's custom client authentication uses CRL and OCSP as proposed by Microsoft. If you choose to use Microsoft's client authentication then comment the config file as below:

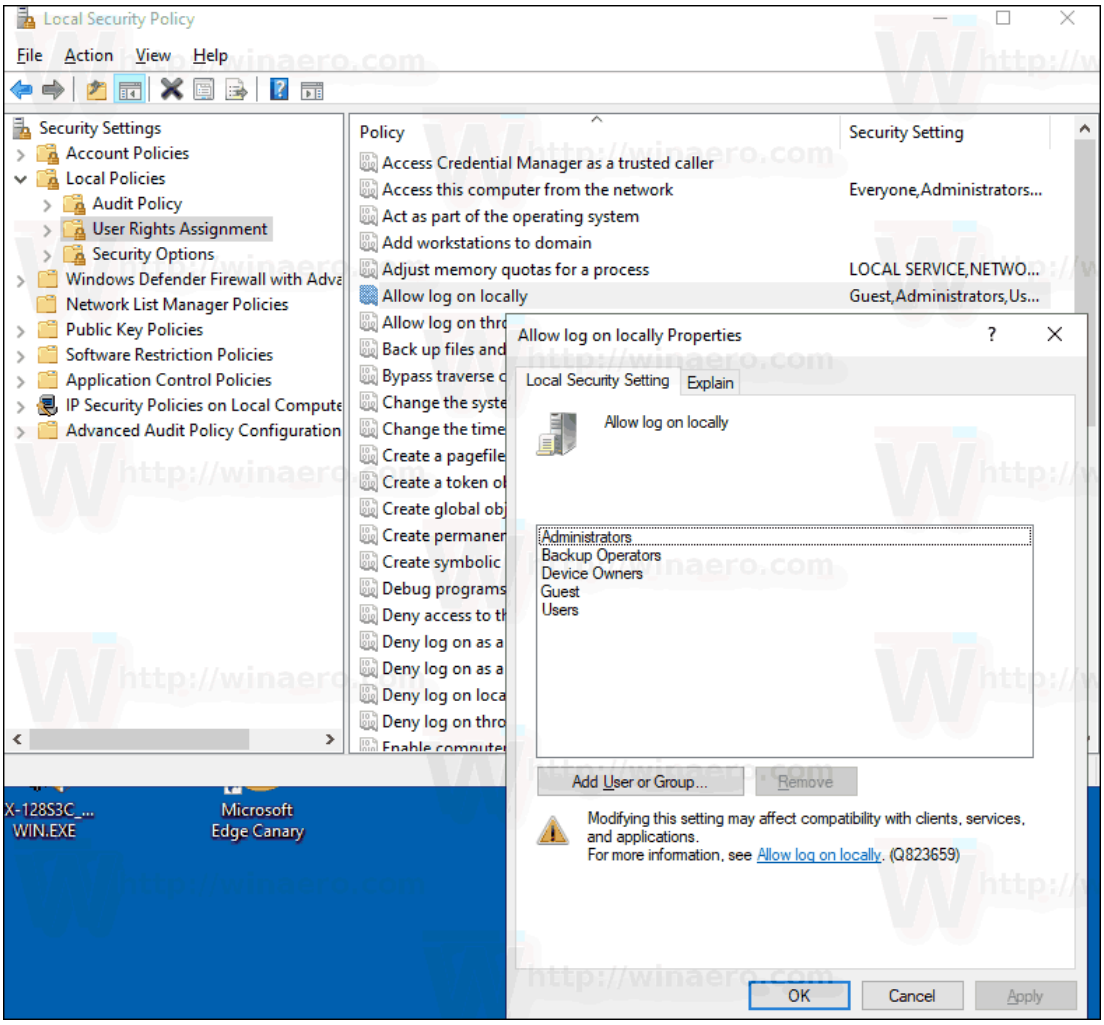
```
<!--<serviceCredentials>
<clientCertificate>
<authentication certificateValidationMode="Custom"
customCertificateValidatorType="AppViewX.CertPlus.Service.CustomValidator, AppViewX.CertPlus.Service" />
</clientCertificate>
</serviceCredentials>-->
```

- AppViewX recommends customers not to change this default authentication configuration provided by AppViewX.
- Refer Appendix A for the prerequisites for managing the Windows Server infrastructure and Troubleshooting for troubleshooting the target machine.

Step 5: Managing a Target Server

To manage a target server with different credentials, the user account can be configured using the AppViewX user interface. Enable the **Allow log on locally** user rights assignment security policy for the account.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



Step 6: Disabling Current Operating System Information

On the https header, modify the registry.

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\HTTP\Parameters] "DisableServerHeader"=dword:00000002
```

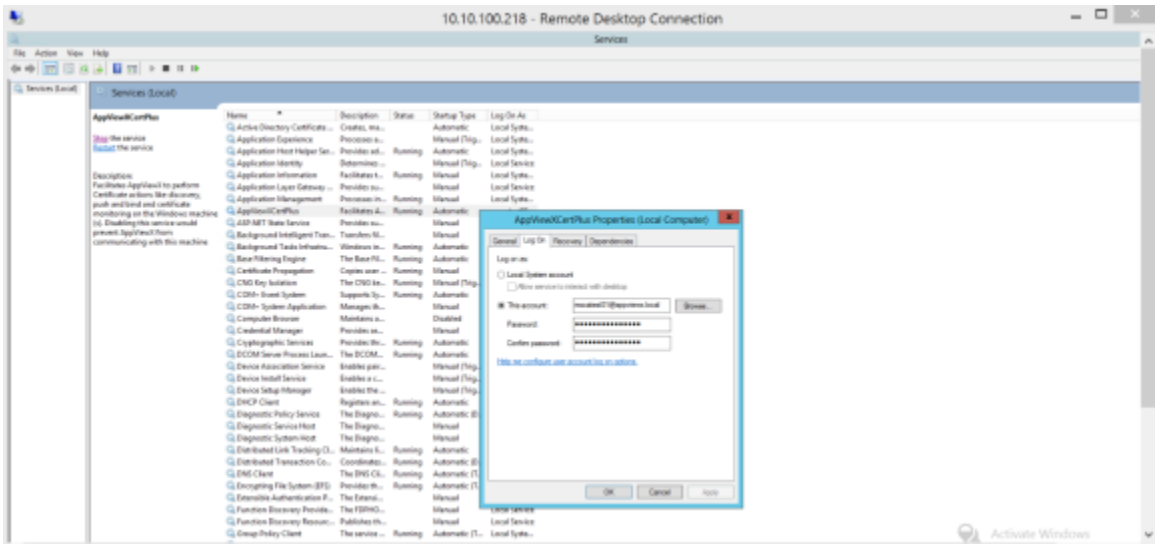
Installation using Non-Admin Service Account

- The AppViewX Windows Gateway can be installed using a service account that is part of the local administrator group account.
- In this case, the following command has to be executed from the PowerShell:

```
netsh http add urlacl url=https://+:8999/appviewx/user=Username@domainname
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- In the above command, the value for `user = <domainserviceaccount>` and the `URL` must be changed respectively.
- On the Regedit path, "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog\Application\AppViewXCertPlus" add the service account and give Full Control permission.
- On the Installation path of the application, the user needs permission to read and write.
- If the network has a policy that the service account cannot be part of the administrator group or that the service account is only a part of the user group, then:
  - The AppViewX Windows Gateway is installed using an admin account.
  - It is then associated with the service account in **services.msc**, by adding the account in the properties of the AppViewXCertPlus service. Refer to the following image.



- Once this is done, stop and start the `AppViewXCertPlus Service` in `services.msc`.

Configuration Parameters

The configuration settings can be found in the *AppViewX.CertPlus.WindowsService.exe.config*.

AppViewX.CertPlus.WindowsService.exe.config file - Fields and Description

Parameter	Description	Value
WMIrequestFolderPath	Specifies the destination folder path for WMI-based use cases.	Folder path
ExportPrivateKey	Configures the option to export the private key during	Yes or No

Copyright © 2025 AppViewX Inc. All Rights Reserved.

AppViewX.CertPlus.WindowsService.exe.config file - Fields and Description (continued)

Parameter	Description	Value
	Certificate Signing Request (CSR) generation.	
CertExportable	Indicates whether the certificate should be marked as exportable when installing in the trust store (Local Machine or Current User).	Yes or No
DefaultStore	Specifies the Default Certificate trust store to be used.	LocalMachine or CurrentUser
DiscoverCurrentStoreForIIS	A flag that triggers discovery from the current user store if set to "YES", specifically for IIS certificate discovery.	Yes or No
WMI	A list of connectivity checks performed during configuration fetch for WMI mode.	Windows Gateway Reachability,User Validation,Windows Temp Folder,WMI Service,CA Discovery Test
NATIVE	A list of connectivity checks performed during configuration fetch for Native mode.	Windows Gateway Reachability,CA Discovery Test
POWERSHELL	A list of connectivity checks performed during configuration fetch for PowerShell mode.	Windows Gateway Reachability,User Validation,RPC Service,WinRM Service,WinRM Configuration,CA Discovery Test
SkipCRLURL	If set to "YES", skip the Certificate Revocation List (CRL) during Download.	Yes or No
DeltaCRL	If set to "YES", enables the download of Delta CRLs and	Yes or No

Copyright © 2025 AppViewX Inc. All Rights Reserved.

AppViewX.CertPlus.WindowsService.exe.config file - Fields and Description (continued)

Parameter	Description	Value
	checks if the certificate has been revoked.	
ValidateRevocation	If set to "YES", checks certificate revocation using both OCSP and CRL.	Yes or No
ValidateThumbprintOnly	If set to "YES", only validates the end entity certificate using its thumbprint and verifies the certificate chain.	Yes or No
HttpsMode	If set to "Yes", remote PowerShell operates in HTTPS mode (port 5986). If set to "No" it operates in HTTP mode (port 5985). If set to "Both", it checks if HTTPS mode is available , if not , falls back to HTTP.	Yes or No or Both
LogonMode	Defines the logon mode	Interactive or Network

• Allow Logon Locally Permission
• WinRM HTTPS Listeners

Allow Logon Locally Permission

The **Allow Logon Locally** policy setting determines which users can start an interactive session on the computer. In the case of AppViewX, the windows agent does an interactive session to the target machine to perform the desired operations. Hence this permission is required if the account used is not part of the local administrator group in the Windows Gateway Machine.

For more information on this setting please refer Allow log on locally - security policy setting - Windows 10 | Microsoft Learn

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Enabling and Configuring the LogonMode Parameter

From AppViewX version 2024.3.0.0, an additional configuration parameter has been introduced to avoid explicitly providing "Allow Logon Locally" permission. It is enabled through a configuration parameter called **LogonMode** which can be configured in the *AppViewX.CertPlus.WindowsService.exe.config* file.

Follow the steps below if the “Allow Logon Locally” permission cannot be provided or the Service Account is part of “Deny Logon Locally”.

1. Go to the AppViewX Windows Gateway Installation Path. The default path is C:\Program Files (x86)\appviewx\AppViewX.CertPlus.Installer
2. Open the **AppViewX.CertPlus.WindowsService.exe.config** file using a text editor.
3. Locate the <appSettings> section and set the following entry: <add key="LogonMode" value="Network" />

   Supported values for the "LogonMode" configuration:
   - **Interactive:** Uses the Interactive logon mode (Requires “Allow Logon Locally” Permission)
   - **Network:** Uses the Network logon method, which does not require explicit "Allow Logon Locally" permission.

   Note: If the "LogonMode" parameter is not specified, the system defaults to **Interactive** logon mode.

WinRM HTTPS Listeners

This section describes the use of WinRM HTTPS Listeners for Powershell Communication mode with AppViewX Windows Gateway.

From AppViewX version 2024.3.0.0, an additional configuration parameter has been introduced to enable support for WinRM HTTPS Listener. A new configuration parameter, **HttpsMode** has been introduced in the *AppViewX.CertPlus.WindowsService.exe.config* file. To configure the **HttpsMode**,

1. Go to the AppViewX Windows Gateway Installation Path. The default path is C:\Program Files (x86)\appviewx\AppViewX.CertPlus.Installer
2. Open the **AppViewX.CertPlus.WindowsService.exe.config** file using a text editor.
3. Locate the <appSettings> section and set the following entry: <add key="HttpsMode" value="Network" />
   Supported values for the "HttpsMode" configuration:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- **Yes**: Enforces the use of only HTTPS for all communication.
- **No**: Explicitly sets HTTP as the communication protocol.
- **Both**: Prioritizes HTTPS; if HTTPS is unavailable, the system will fall back to HTTP.

> **Note:** If the **HttpsMode** configuration is not explicitly set, the system will default to the use **HTTP** for communication.

Prerequisites for setting up and using HTTPS mode

1. **WinRM HTTPS Service Setup**: The WinRM service must be configured and bound to a valid **SSL certificate**.
2. **Certificate Requirements**:
   a. The certificate must be issued using the **Web Server template**.
   b. The **Common Name (CN)** in the certificate must match the **Fully Qualified Domain Name (FQDN)** of the respective machine.
   c. This is a mandatory requirement by Microsoft for **WSMAN over HTTPS** when using **PowerShell Remoting**.

Procedure for Unconfigured HTTPS Listeners on Target Devices

1. Set the **HttpsMode** flag to **Both** in the configuration file. (By default it will be set to “No”).
2. If the required target devices are not configured with HTTPS Listeners, it will be on-boarded using HTTP.
3. Post that, the required target devices can be configured with HTTPS listeners bound with the required certificates (requirements of the certificates mentioned above).
4. Once all the required certificates are bound and the HTTPS listeners are available in all the required target devices, set the HttpsMode flag to "Yes" to use only HTTPS Listeners for any further communications.

Uninstalling the AppViewX Windows Gateway

Uninstalling AppViewX Windows Gateway involves the following steps:

1. Go to Windows control panel, select **Add or Remove program**.
2. Select **AppViewX Windows Gateway**, and then click on **Uninstall** button.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Updating AppViewX Windows Gateway

Before updating the AppViewX Windows Gateway to a newer version, the old version of the AppViewX Windows Gateway should be uninstalled. Follow the instructions in Chapter 5 to uninstall AppViewX Windows Gateway.

After uninstalling the older version of AppViewX Windows Gateway, proceed with the installation of the new AppViewX Windows Gateway. Refer Chapter 2 for instructions on Installing the AppViewX Windows Gateway.

Appendix A

- General Prerequisites
- Firewall Requirements
- Minimum Permissions Required for Communication

General Prerequisites

If a device that has the AppViewX Microsoft Gateway installed on it has to be managed in AppViewX, communication mode reset to WMI always.

Additional prerequisites that can be validated manually or by the AppViewX Windows Gateway Troubleshooting tool provided with the AppViewX Windows Gateway

Component	Description	Scripts
.Net Framework 4.5 and above	Download dotnet-framework-runtime from Microsoft software download center.	
POWERSHELL 4+	Download PowerShell from Microsoft software download center.	Powershell $PSVersionTable.PSVersion
Certadm.dll (Applicable ONLY if CA servers to be managed)	Check if dll is available in the C: WindowsSystem32 folder or install the Microsoft Remote Server Administration Tools (RSAT) for the respective	cd C:WindowsSystem32 and then dir certadm.dll

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Additional prerequisites that can be validated manually or by the AppViewX Windows Gateway Troubleshooting tool provided with the AppViewX Windows Gateway (continued)

Component	Description	Scripts
	OS from Microsoft software download center.	
CertUtil	File will be available at the System32 folder.	Run certutil in the command prompt.
NetSH	Copy to the System32 folder if it is not available.	Run netsh in the command prompt.
RPC	Start the Remote procedure call in the services	net start RpcSs
WMI	Start the Windows Management Instrumentation in the services.	net start Winmgmt
WinRM	Start the Windows Remote Management.	net start WinRM
User Permission	When the users are added in the Group and the machine is not restarted a permission error will occur. Ensure that the machine is restarted when the user is added to a group.	Gwmi win32_groupuser -computer ptpll594|? {$_.groupcomponent -like '*""Administrators""'} \|select PartComponentnet localgroup administratorsCheck if user can access C$/windows/temp or admin$/Temp Local admin addition needs to restart.
File Operations	-	Check if the user can access C $/windows/temp or admin$/temp. If you do not have c-drive then change the configuration to the available drive.
Port	Check if the port is already in use.	netstat -an |find ""8999" Check the Firewall outbound rules for the port

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Additional prerequisites that can be validated manually or by the AppViewX Windows Gateway Troubleshooting tool provided with the AppViewX Windows Gateway (continued)

Component	Description	Scripts
		Ping test from AppViewX Antivirus block for the port Turn off the local firewall Check the server, client, root, and intermediate certificates Check if the C: Logs folder exists and the permissions If you check in the Internet Explorer then the enhanced security must be disabled in the server role local server.
Powershell Remoting	-	Enter-PSSession -ComputerName <computername> -Credential <username>

Firewall Requirements

The firewall must not block the following ports:

Component	Port
Powershell	5985 (HTTP), 5986 (HTTPS)
WMI	135 + Dynamic port: 49152-65534
SMB	445
Native	135

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Minimum Permissions Required for Communication

The AppViewX Windows Gateway agent communicates with the CAs via the following three communication modes:

- PowerShell
- WMI
- Native API

PowerShell

For the following use cases, this section lists the minimum permissions required for the AppViewX Windows Gateway to communicate with the CAs via PowerShell:

> **Important:** Log on locally is needed to impersonate the user, to prevent execution of arbitrary scripts or commands on a remote Windows machine without proper authentication or authorization.

- Discovery
- Create CSR
- Create Certificate
- Create Certificate-Upload CSR
- Renew Certificate
- Revoke Certificate
- Certificate Push
- Certificate Bind

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Discovery

Microsoft CA

Requirement	AppViewX Windows Gateway	Microsoft CA
User account type	Service account	Service account
User permission	NA	• Full control permission to **C:\Windows\Temp** • Read permission at CA level for the service account or the service account group
Services	RPC Service,WinRM Service,WinRM Configuration, Powershell remoting,certutil.exe command availability	RPC Service,WinRM Service,WinRM Configuration, Powershell remoting,certutil.exe command availability

IIS

Requirement	AppViewX Windows Gateway	IIS
User account type	Admin account	Admin account
Services	RPC Service,WinRM Service,WinRM Configuration, Powershell remoting	RPC Service,WinRM Service,WinRM Configuration, Powershell remoting

Microsoft PC

Requirement	AppViewX Windows Gateway	Microsoft PC
User account type	Admin account	Admin account
Services	RPC Service,WinRM Service,WinRM Configuration, Powershell remoting`	RPC Service,WinRM Service,WinRM Configuration, Powershell remoting

Microsoft Server

Requirement	AppViewX Windows Gateway	Microsoft Server
User account type	Admin account	Admin account

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Microsoft Server (continued)

Requirement	AppViewX Windows Gateway	Microsoft Server
		Note: The Microsoft Server administration tool can be handled only by local administrators, as local admin rights are typically required to register and run a Windows service that listens on HTTPS-based APIs for Certificate Management in the Windows Certificate Store or filesystem.
User permission	NA	Read permission for the folder to be discovered
Services	RPC Service,WinRM Service,WinRM Configuration, Powershell remoting	RPC Service,WinRM Service,WinRM Configuration, Powershell remoting

Create CSR

IIS

Requirement	AppViewX Windows Gateway	IIS
User account type	Admin account	Admin account

Copyright © 2025 AppViewX Inc. All Rights Reserved.

IIS (continued)

Requirement	AppViewX Windows Gateway	IIS
		Note: The IIS administration tool can be handled only by local administrators, as local admin rights are typically required to register and run a Windows service that listens on HTTPS-based APIs for Certificate Management in the Windows Certificate Store or filesystem.
Services	RPC Service,WinRM Service,WinRM Configuration, Powershell remoting	RPC Service,WinRM Service,WinRM Configuration, Powershell remoting

Microsoft PC

Requirement	AppViewX Windows Gateway	Microsoft PC
User account type	Admin account	Admin account

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Microsoft PC (continued)

Requirement	AppViewX Windows Gateway	Microsoft PC
		Note: The Microsoft PC administration tool can be handled only by local administrators, as local admin rights are typically required to register and run a Windows service that listens on HTTPS-based APIs for Certificate Management in the Windows Certificate Store or filesystem.
Services	RPC Service,WinRM Service,WinRM Configuration, Powershell remoting`	RPC Service,WinRM Service,WinRM Configuration, Powershell remoting

Create Certificate

Microsoft CA

Requirement	AppViewX Windows Gateway	Microsoft CA
User account type	Service account	Service account
User permission	NA	• Request certificates permission at CA level for the service account/service account group/ authenticated users • Enroll permission at the certificate template level for

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Microsoft CA (continued)

Requirement	AppViewX Windows Gateway	Microsoft CA
		the service account/service account group/ authenticated users
Services	RPC Service,WinRM Service,WinRM Configuration, Powershell remoting,certutil.exe command availability	RPC Service,WinRM Service,WinRM Configuration, Powershell remoting,certutil.exe command availability

Create Certificate-Upload CSR

Microsoft CA

Requirement	AppViewX Windows Gateway	Microsoft CA
User account type	Service account	Service account
User permission	NA	• Request certificates permission at CA level for the service account/service account group/ authenticated users • Enroll permission at the certificate template level for the service account/service account group/ authenticated users
Services	RPC Service,WinRM Service,WinRM Configuration, Powershell remoting,certutil.exe command availability	RPC Service,WinRM Service,WinRM Configuration, Powershell remoting,certutil.exe command availability

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Renew Certificate

Microsoft CA

Requirement	AppViewX Windows Gateway	Microsoft CA
User account type	Service account	Service account
User permission	NA	• Request certificates permission at CA level for the service account/service account group/ authenticated users • Enroll permission at the certificate template level for the service account/service account group/ authenticated users
Services	RPC Service,WinRM Service,WinRM Configuration, Powershell remoting,certutil.exe command availability	RPC Service,WinRM Service,WinRM Configuration, Powershell remoting,certutil.exe command availability

Revoke Certificate

Microsoft CA

Requirement	AppViewX Windows Gateway	Microsoft CA
User account type	Service account	Service account
User permission	NA	Issue and manage certificates permission at CA level for the service account or the service account group or the authenticated users
Services	RPC Service,WinRM Service,WinRM Configuration,	RPC Service,WinRM Service,WinRM Configuration,

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Microsoft CA (continued)

Requirement	AppViewX Windows Gateway	Microsoft CA
	PowerShell remoting,certutil.exe command availability	PowerShell remoting,certutil.exe command availability

Certificate Push

IIS

Requirement	AppViewX Windows Gateway	IIS
User account type	Admin account	Admin account **Note:** The IIS administration tool can be handled only by local administrators, as local admin rights are typically required to register and run a Windows service that listens on HTTPS-based APIs for Certificate Management in the Windows Certificate Store or filesystem.
User permission	NA	Full control permission to **C:\Windows\Temp**
Services	RPC Service,WinRM Service,WinRM Configuration, Powershell remoting	RPC Service,WinRM Service,WinRM Configuration, Powershell remoting

Microsoft PC

Requirement	AppViewX Windows Gateway	Microsoft PC
User account type	Admin account	Admin account

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Microsoft PC (continued)

Requirement	AppViewX Windows Gateway	Microsoft PC
		Note: The Microsoft PC administration tool can be handled only by local administrators, as local admin rights are typically required to register and run a Windows service that listens on HTTPS-based APIs for Certificate Management in the Windows Certificate Store or filesystem.
Services	RPC Service,WinRM Service,WinRM Configuration, Powershell remoting`	RPC Service,WinRM Service,WinRM Configuration, Powershell remoting

Microsoft Server

Requirement	AppViewX Windows Gateway	Microsoft Server
User account type	Admin account	Admin account

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Microsoft Server (continued)

Requirement	AppViewX Windows Gateway	Microsoft Server
		Note: The Microsoft Server administration tool can be handled only by local administrators, as local admin rights are typically required to register and run a Windows service that listens on HTTPS-based APIs for Certificate Management in the Windows Certificate Store or filesystem.
User permission	NA	Write permission for the folder to be discovered
Services	RPC Service,WinRM Service,WinRM Configuration, Powershell remoting`	RPC Service,WinRM Service,WinRM Configuration, Powershell remoting

Certificate Bind

IIS

Requirement	AppViewX Windows Gateway	IIS
User account type	Admin account	Admin account

Copyright © 2025 AppViewX Inc. All Rights Reserved.

IIS (continued)

Requirement	AppViewX Windows Gateway	IIS
		Note: The IIS administration tool can be handled only by local administrators, as local admin rights are typically required to register and run a Windows service that listens on HTTPS-based APIs for Certificate Management in the Windows Certificate Store or filesystem.
Services	RPC Service,WinRM Service,WinRM Configuration, Powershell remoting	RPC Service,WinRM Service,WinRM Configuration, Powershell remoting

WMI

For the following use cases, this section lists the minimum permissions required for the AppViewX Windows Gateway to communicate with the CAs via WMI:

Important: Log on locally is needed to impersonate the user, to prevent execution of arbitrary scripts or commands on a remote Windows machine without proper authentication or authorization.

- Discovery
- Create CSR
- Create Certificate
- Create Certificate-Upload CSR
- Renew Certificate

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- Revoke Certificate
- Certificate Push
- Certificate Bind

Discovery

Microsoft CA

Requirement	AppViewX Windows Gateway	Microsoft CA
User account type	Service account	Service account
User permission	NA	• Full control permission to **C:\Windows\Temp** • Read permission at CA level for the service account or the service account group
Services	WMI Service, certutil.exe command availability	WMI Service, certutil.exe command availability

IIS

Requirement	AppViewX Windows Gateway	IIS
User account type	Admin account	Admin account

Copyright © 2025 AppViewX Inc. All Rights Reserved.

IIS (continued)

Requirement	AppViewX Windows Gateway	IIS
		Note: The IIS administration tool can be handled only by local administrators, as local admin rights are typically required to register and run a Windows service that listens on HTTPS-based APIs for Certificate Management in the Windows Certificate Store or filesystem.
User permission		Full control permission to **C:\Windows\Temp**
Services	WMI Service	WMI Service

Microsoft PC

Requirement	AppViewX Windows Gateway	Microsoft PC
User account type	Admin account	Admin account

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Microsoft PC (continued)

Requirement	AppViewX Windows Gateway	Microsoft PC
		Note: The Microsoft PC administration tool can be handled only by local administrators, as local admin rights are typically required to register and run a Windows service that listens on HTTPS-based APIs for Certificate Management in the Windows Certificate Store or filesystem.
User permission		Full control permission to **C:\Windows\Temp**
Services	WMI Service	WMI Service

Microsoft Server

Requirement	AppViewX Windows Gateway	Microsoft Server
User account type	Service account	Service account
User permission	NA	• Full control permission to **C:\Windows\Temp** • Read permission at CA level for the service account or the service account group
Services	WMI Service, certutil.exe command availability	WMI Service, certutil.exe command availability

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Create CSR

IIS

Requirement	AppViewX Windows Gateway	IIS
User account type	Admin account	Admin account **Note:** The IIS administration tool can be handled only by local administrators, as local admin rights are typically required to register and run a Windows service that listens on HTTPS-based APIs for Certificate Management in the Windows Certificate Store or filesystem.
Services	WMI Service	WMI Service

Microsoft PC

Requirement	AppViewX Windows Gateway	Microsoft PC
User account type	Admin account	Admin account

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Microsoft PC (continued)

Requirement	AppViewX Windows Gateway	Microsoft PC
		Note: The Microsoft PC administration tool can be handled only by local administrators, as local admin rights are typically required to register and run a Windows service that listens on HTTPS-based APIs for Certificate Management in the Windows Certificate Store or filesystem.
Services	WMI Service	WMI Service

Create Certificate

Microsoft CA

Requirement	AppViewX Windows Gateway	Microsoft CA
User account type	Service account	Service account
User permission	NA	• Request certificates permission at the CA level for the service account or the service account group or the authenticated users • Enroll permission at the certificate template level for

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Microsoft CA (continued)

Requirement	AppViewX Windows Gateway	Microsoft CA
		the service account or the service account group or the authenticated users
Services	WMI Service, certutil.exe command availability	WMI Service, certutil.exe command availability

Create Certificate-Upload CSR

Microsoft CA

Requirement	AppViewX Windows Gateway	Microsoft CA
User account type	Service account	Service account
User permission	NA	• Request certificates permission at the CA level for the service account or the service account group or the authenticated users • Enroll permission at the certificate template level for the service account or the service account group or the authenticated users
Services	WMI Service, certutil.exe command availability	WMI Service, certutil.exe command availability

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Renew Certificate

Microsoft CA

Requirement	AppViewX Windows Gateway	Microsoft CA
User account type	Service account	Service account
User permission	NA	• Request certificates permission at the CA level for the service account or the service account group or the authenticated users • Enroll permission at the certificate template level for the service account or the service account group or the authenticated users
Services	WMI Service, certutil.exe command availability	WMI Service, certutil.exe command availability

Revoke Certificate

Microsoft CA

Requirement	AppViewX Windows Gateway	Microsoft CA
User account type	Service account	Service account
User permission	NA	• Request certificates permission at the CA level for the service account or the service account group or the authenticated users • Enroll permission at the certificate template level for

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Microsoft CA (continued)

Requirement	AppViewX Windows Gateway	Microsoft CA
		the service account or the service account group or the authenticated users
Services	WMI Service, certutil.exe command availability	WMI Service, certutil.exe command availability

Certificate Push

IIS

Requirement	AppViewX Windows Gateway	IIS
User account type	Admin account	Admin account  **Note:** The IIS administration tool can be handled only by local administrators, as local admin rights are typically required to register and run a Windows service that listens on HTTPS-based APIs for Certificate Management in the Windows Certificate Store or filesystem.
Services	WMI Service	WMI Service

Microsoft PC

Requirement	AppViewX Windows Gateway	Microsoft PC
User account type	Admin account	Admin account

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Microsoft PC (continued)

Requirement	AppViewX Windows Gateway	Microsoft PC
		Note: The Microsoft PC administration tool can be handled only by local administrators, as local admin rights are typically required to register and run a Windows service that listens on HTTPS-based APIs for Certificate Management in the Windows Certificate Store or filesystem.
Services	WMI Service	WMI Service

Microsoft Server

Requirement	AppViewX Windows Gateway	Microsoft Server
User account type	Service account	Service account
User permission	NA	• Full control permission to **C:\Windows\Temp** • Read permission at CA level for the service account or the service account group
Services	WMI Service	WMI Service

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Certificate Bind

IIS

Requirement	AppViewX Windows Gateway	IIS
User account type	Admin account	Admin account **Note:** The IIS administration tool can be handled only by local administrators, as local admin rights are typically required to register and run a Windows service that listens on HTTPS-based APIs for Certificate Management in the Windows Certificate Store or filesystem.
Services	WMI Service	WMI Service

Native API

For the following use cases, this section lists the minimum permissions required for the AppViewX Windows Gateway to communicate with the CAs via Native API:

- All Operations
- Discovery
- Create Certificate
- Create Certificate-Upload CSR
- Renew Certificate
- Revoke Certificate

Copyright © 2025 AppViewX Inc. All Rights Reserved.

All Operations

Microsoft CA

Requirement	AppViewX Windows Gateway	Microsoft CA
User account type	Service account	Service account
User permission	NA	• Read, request certificates, and issue and manage certificates permission at the CA level for the service account/service account group /authenticated users • Enroll permission at the certificate template level for the service account/service account group/ authenticated users
Services	RPC Service	RPC Service, certutil.exe command availability

Discovery

Microsoft CA

Requirement	AppViewX Windows Gateway	Microsoft CA
User account type	Service account	Service account
User permission	NA	• Read permission at the CA level for the service account or the service account group
Services	RPC Service	RPC Service, certutil.exe command availability

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Create Certificate

Microsoft CA

Requirement	AppViewX Windows Gateway	Microsoft CA
User account type	Service account	Service account
User permission	NA	• Request certificates permission at the CA level for the service account or the service account group or the authenticated users • Enroll permission at the certificate template level for the service account or the service account group or the authenticated users
Services	RPC Service	RPC Service, certutil.exe command availability

Create Certificate-Upload CSR

Microsoft CA

Requirement	AppViewX Windows Gateway	Microsoft CA
User account type	Service account	Service account
User permission	NA	• Request certificates permission at the CA level for the service account or the service account group or the authenticated users • Enroll permission at the certificate template level for

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Microsoft CA (continued)

Requirement	AppViewX Windows Gateway	Microsoft CA
		the service account or the service account group or the authenticated users
Services	RPC Service	RPC Service

Renew Certificate

Microsoft CA

Requirement	AppViewX Windows Gateway	Microsoft CA
User account type	Service account	Service account
User permission	NA	• Request certificates permission at the CA level for the service account or the service account group or the authenticated users • Enroll permission at the certificate template level for the service account or the service account group or the authenticated users
Services	RPC Service	RPC Service, certutil.exe command availability

Revoke Certificate

Microsoft CA

Requirement	AppViewX Windows Gateway	Microsoft CA
User account type	Service account	Service account

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Microsoft CA (continued)

Requirement	AppViewX Windows Gateway	Microsoft CA
User permission	NA	Issue and manage certificates permission at CA level for the service account or the service account group or the authenticated users
Services	RPC Service	RPC Service,certutil.exe command availability

Troubleshooting

Connection timeout, Error 404 Gateway Unreachable

Troubleshooting using "nc" command	
Troubleshooting Steps	1. Login to the ***appviewx*** pods CLI to check communication a. **On Prem :** Login in to the *appviewx* CLI, go to the **vendor pods**. b. **Saas :** Login to the Cloud Connector, go to the **platform pod**. 2. After logging in, run the commands below in the terminal. `nc -zv <Ip or hostname> <8999 or configured port>` where, • <Ip or hostname> is the hostname or IP of the windows agent machine. • 8999 is the default port of Windows Agent (provide the configured port in case of any custom port).
Expected Output:	`Connection to 192.168.142.124 8999 port [tcp/*] succeeded!`

Troubleshooting using "curl" command	
Troubleshooting Steps	1. Login to the ***appviewx*** pods CLI to check communication a. **On Prem :** Login in to the *appviewx* CLI, go to the **vendor pods**. b. **Saas :** Login to the Cloud Connector, go to the **platform pod**.

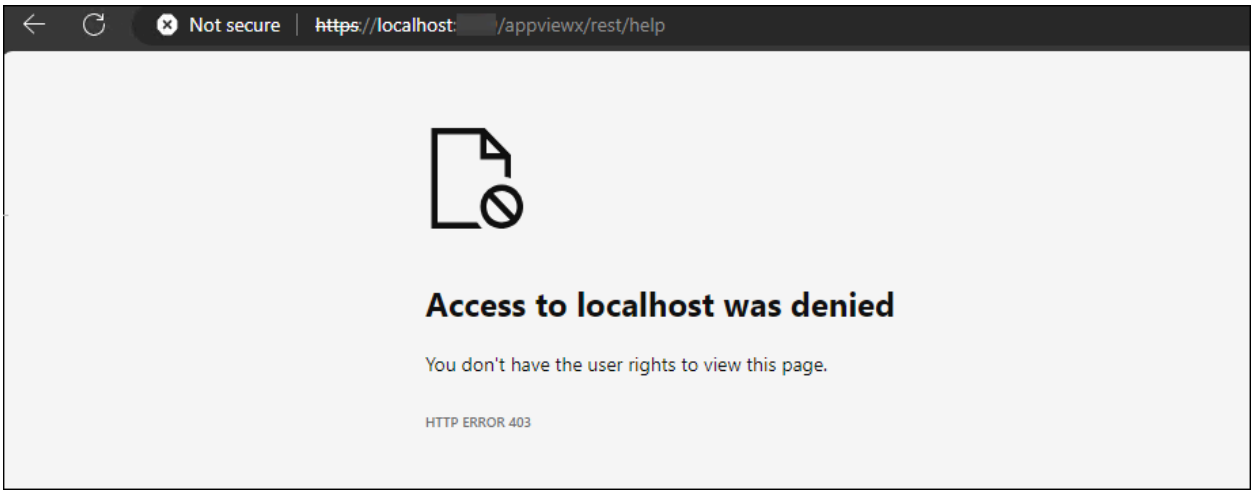
Copyright © 2025 AppViewX Inc. All Rights Reserved.

Troubleshooting using "curl" command	
	2. After logging in, run the commands below in the terminal. `curl -kv https://<hostname or ip>:8999/appviewx/rest/Ping` where, • <Ip or hostname> is the hostname or IP of the windows agent machine. • 8999 is the default port of Windows Agent (provide the configured port in case of any custom port).
Expected Output:	Trying 192.168.142.124:8999... * TCP_NODELAY set * Connected to 192.168.142.124 (192.168.142.124) port 8999 (#0) * ALPN, offering h2 * ALPN, offering http/1.1 * successfully set certificate verify locations: * CAfile: /etc/ssl/certs/ca-certificates.crt CApath: /etc/ssl/certs * TLSv1.3 (OUT), TLS handshake, Client hello (1): * TLSv1.3 (IN), TLS handshake, Server hello (2): * TLSv1.2 (IN), TLS handshake, Certificate (11): * TLSv1.2 (IN), TLS handshake, Server key exchange (12): * TLSv1.2 (IN), TLS handshake, Request CERT (13): * TLSv1.2 (IN), TLS handshake, Server finished (14): * TLSv1.2 (OUT), TLS handshake, Certificate (11): * TLSv1.2 (OUT), TLS handshake, Client key exchange (16): * TLSv1.2 (OUT), TLS change cipher, Change cipher spec (1): * TLSv1.2 (OUT), TLS handshake, Finished (20): * TLSv1.2 (IN), TLS handshake, Finished (20): * SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384 * ALPN, server accepted to use h2 * Server certificate: * subject: CN=windows-gateway.appviewx.com * start date: May 10 00:00:00 2024 GMT * expire date: May 9 23:59:59 2029 GMT * issuer: CN=AppViewX Intermediate CA; O=AppViewX Inc; L=Seattle; ST=Washington; C=US * SSL certificate verify result: unable to get local issuer certificate (20), continuing anyway. * Using HTTP2, server supports multi-use

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Troubleshooting using "curl" command	
	* Connection state changed (HTTP/2 confirmed) * Copying HTTP/2 data in stream buffer to connection buffer after upgrade: len=0 * Using Stream ID: 1 (easy handle 0x563e972e1650) > GET /appviewx/rest/Ping HTTP/2 > Host: 192.168.142.124:8999 > user-agent: curl/7.68.0 > accept: */* > * Connection state changed (MAX_CONCURRENT_STREAMS == 100)! < HTTP/2 403 < content-length: 0 < date: Thu, 15 Aug 2024 07:13:44 GMT * Connection #0 to host 192.168.142.124 left intact

Access to local host was denied

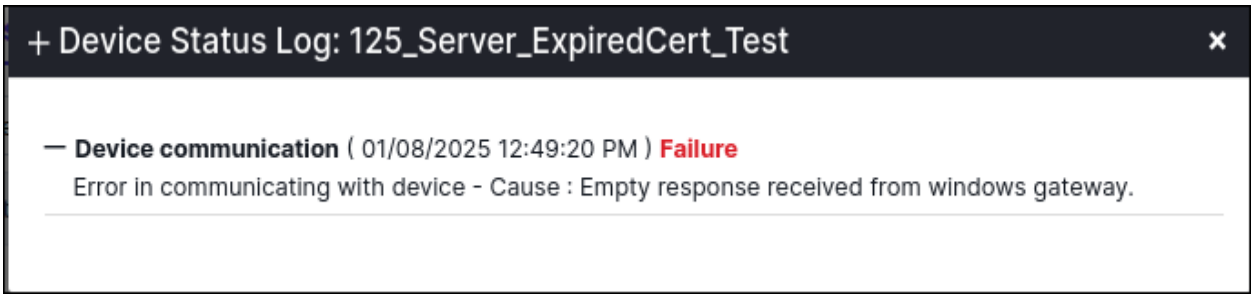


To resolve the Access Denied error during installation verification,

- When choosing the client certificate for authentication, and attempting to reach the *https://localhost:8999/appviewx/rest/help* URL in the browser, import the client certificate in the browser's personal store and then try again.

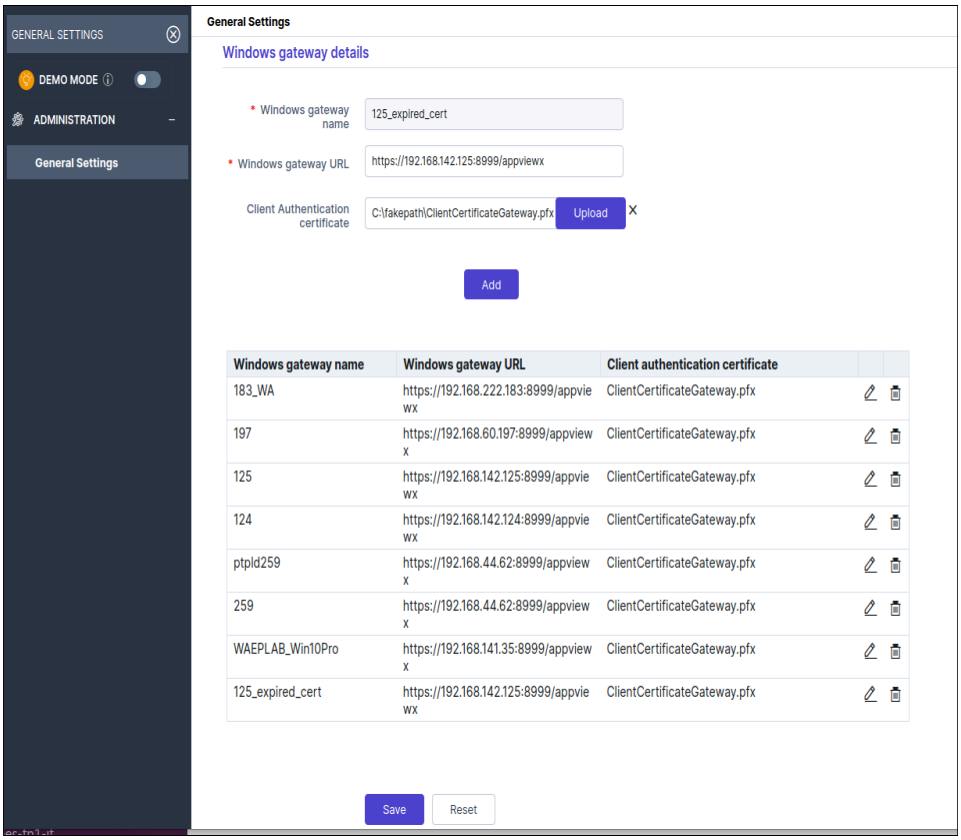
Copyright © 2025 AppViewX Inc. All Rights Reserved.

Empty response received from windows gateway

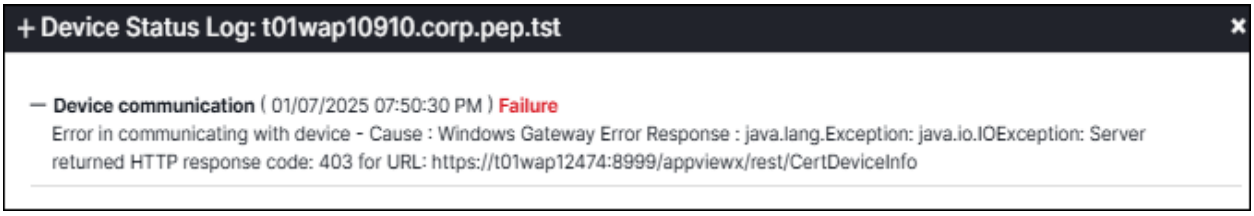


Potential Causes	The above error messages can occur when the certificate uploaded while adding the Windows Gateway is different from the Client Certificate in the Windows Gateway Installation Package or if the certificate has expired.
Fix	Ensure the following: • The same certificate used during installation has been added while configuring the Windows Gateway. • The latest version of the Client Certificate issued from AppViewX is used with AppViewX Windows Gateway.
Steps	**To update certificate in AVX UI:** 1. Go to general settings under **CERT+**. 2. Select edit option for the corresponding gateway agent and update the latest certificate under **windows gateway details**. 3. Click **Add** and then **Save**.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

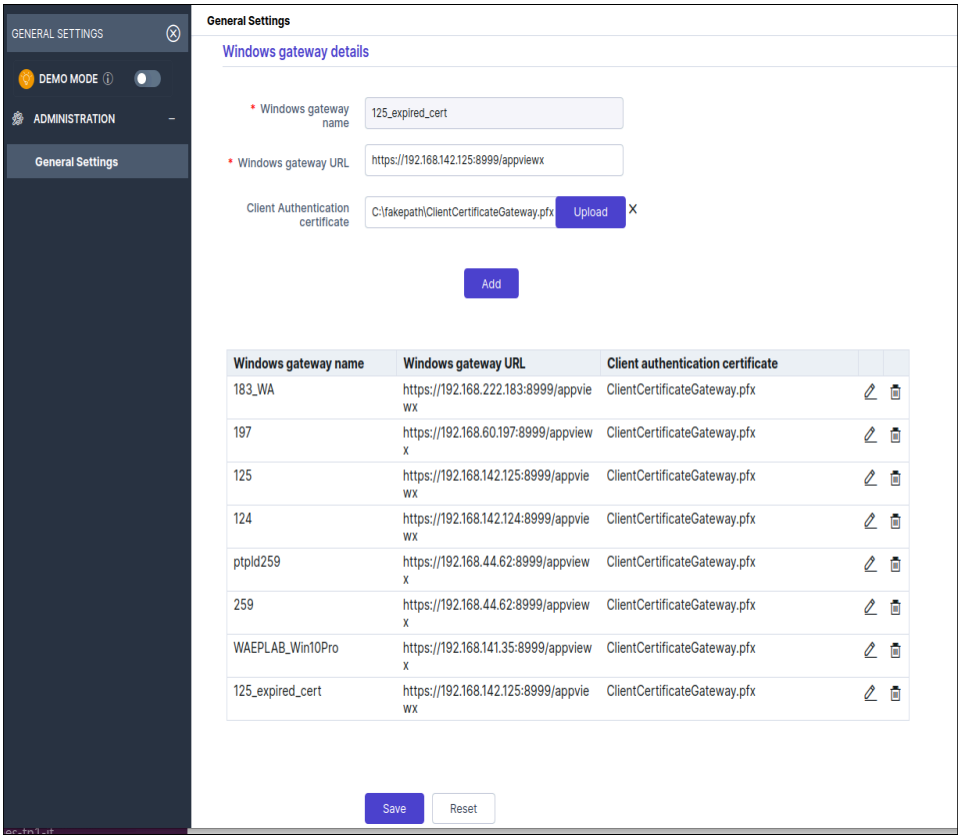


Server returned HTTP response code:403



Potential Causes	The above error messages can occur when the certificate uploaded while adding the Windows Gateway is different from the Client Certificate in the Windows Gateway Installation Package or if the certificate has expired.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Fix	Ensure the following: • The same certificate used during installation has been added while configuring the Windows Gateway. • The latest version of the Client Certificate issued from AppViewX is used with AppViewX Windows Gateway.
Steps	**To update certificate in AVX UI:** 1. Go to general settings under **CERT+**. 2. Select edit option for the corresponding gateway agent and update the latest certificate under **windows gateway details**. 3. Click **Add** and then **Save**. 

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Network Path not Found” during Certificate Push/Bind

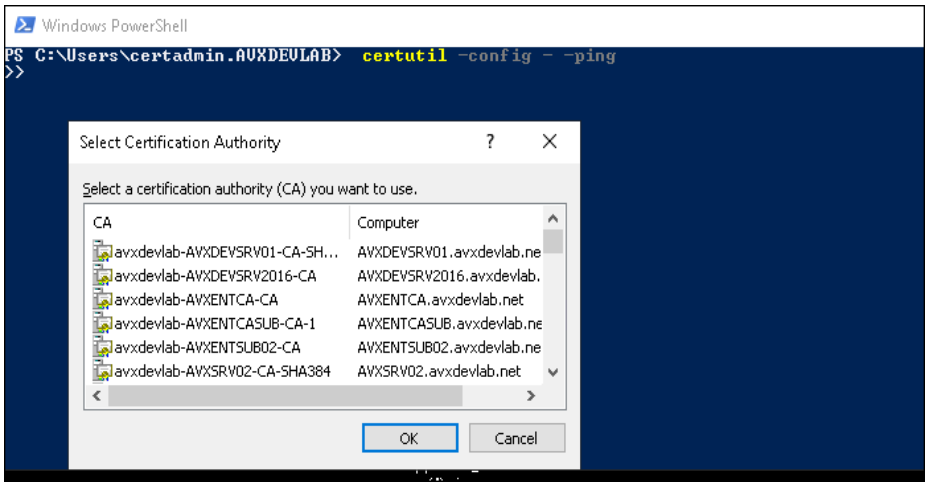
Potential Causes	This error can occur while attempting to perform a Push or Bind operation to a target server. **Port 445 Blocked:** One of the most common scenarios under which this scenario occurs is when the port 445 is blocked in the target machine. This port is used for transferring any file from the Windows Gateway machine to the target server using SMB Protocol.

Troubleshooting Steps	**To update certificate in AVX UI:**
	
	1. Check the Firewall Settings in the Windows Gateway machine and the target machine and check if any firewall rule is blocking the port 445.
	2. Allow or open the TCP port 445.
	3. After allowing/opening the port, check the communication from the windows gateway machine to the target machine using **Test-NetConnection** command.
	`Test-NetConnection -ComputerName <hostname> -Port <port>`
	Sample expected result:
	`ComputerName : computername`
	`RemoteAddress : 93.184.216.34`
	`RemotePort : 80`
	`InterfaceAlias : Ethernet`
	`SourceAddress : 192.168.1.10`
	`PingSucceeded : True`
	`TcpTestSucceeded : True`
	4. Retry the push or bind operation from the AppViewX GUI or Gateway.

RPC Server is unavailable

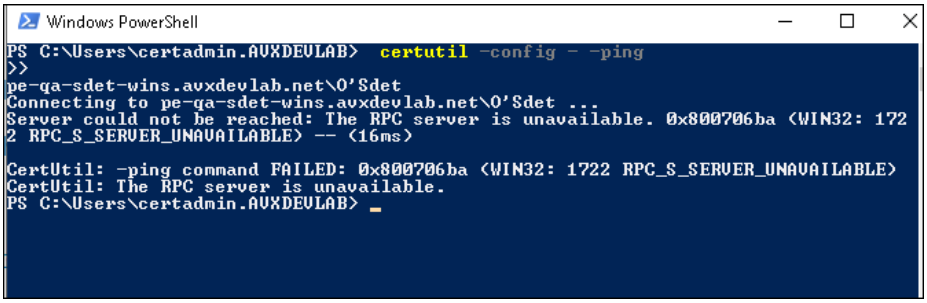
Potential Causes	This error can occur while attempting to onboard a CA or initiate a CA Discovery. Some of the most common cases under which this error can occur are as follows:

Copyright © 2025 AppViewX Inc. All Rights Reserved.

	CA Unreachable from Windows Gateway Machine: If Certificate Authority (CA) discovery fails, it typically indicates that the agent or system cannot locate or communicate with the CA.
Verification	**To verify if the CA is reachable:** 1. Execute the below command on the windows agent machine. It lists all CAs in the domain. `certutil -config - -ping` *Expected Output:*  2. Select the required CA to check the connectivity. *Expected Output if CA is reachable from the Gateway Machine* 

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Expected Output if CA is not reachable from the Gateway Machine

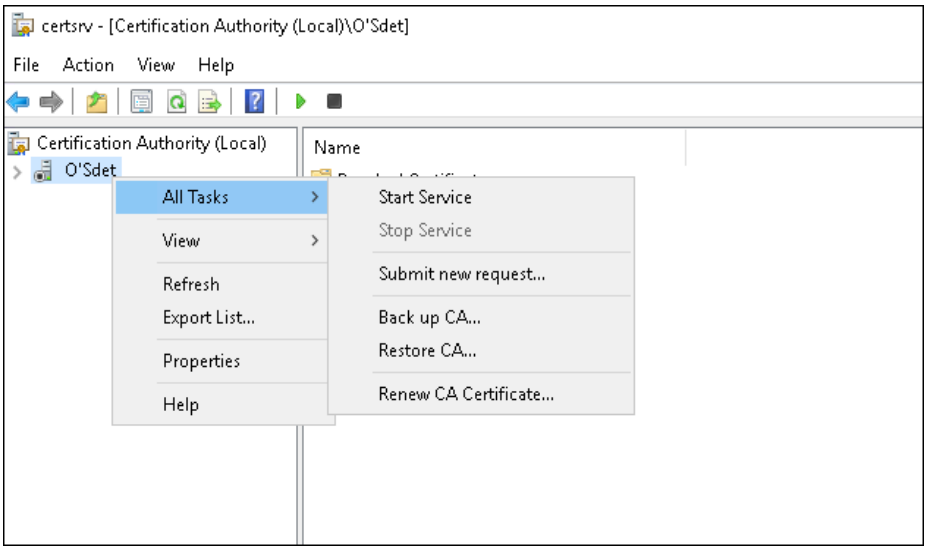


Troubleshooting Steps

Checking the Certificate Authority Service: If the RPC Server Unavailable error persists, try the steps below.

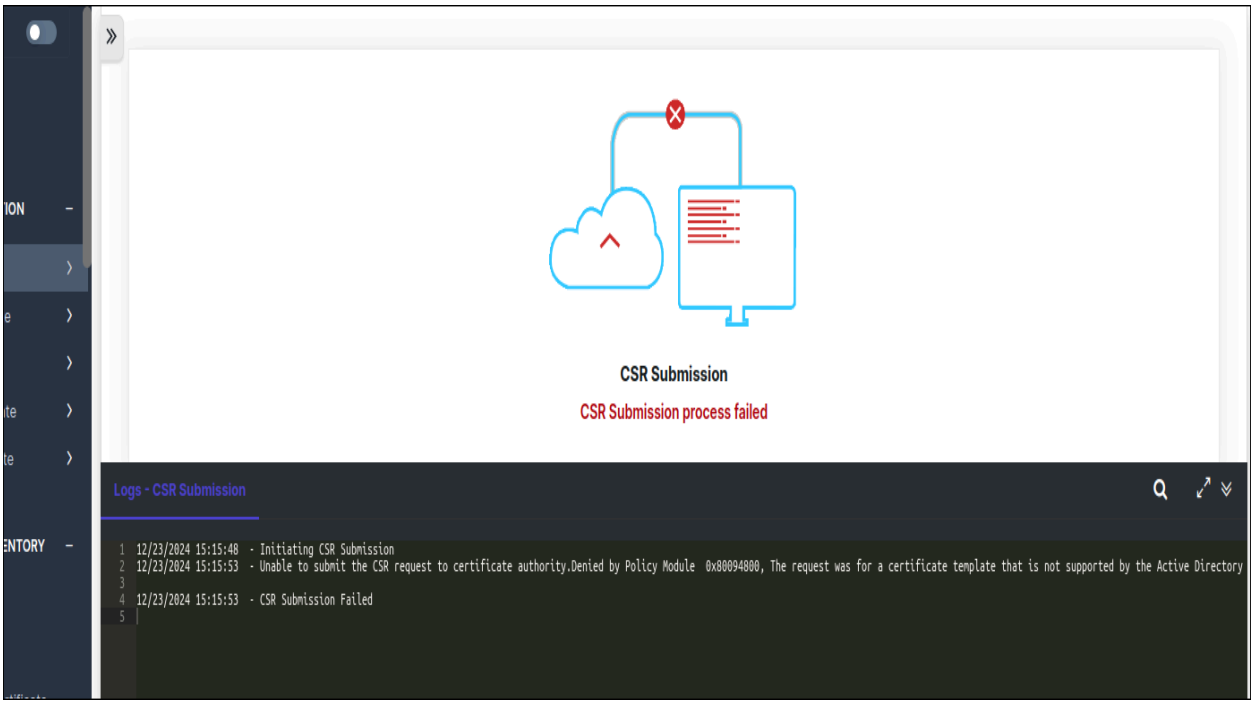
1. Navigate to the machine where the CA is configured.
2. Open **Run** (*Win* + *R*) and type **certsrv.msc**, then hit the Enter key.
3. In the **Certificate Authority** window, right-click on the CA name.
4. Select **All Tasks** and then click **Start Service**.
5. Return to the agent machine and rerun the `certutil` command to ensure the CA is now reachable.

```
certutil -config - -ping
```

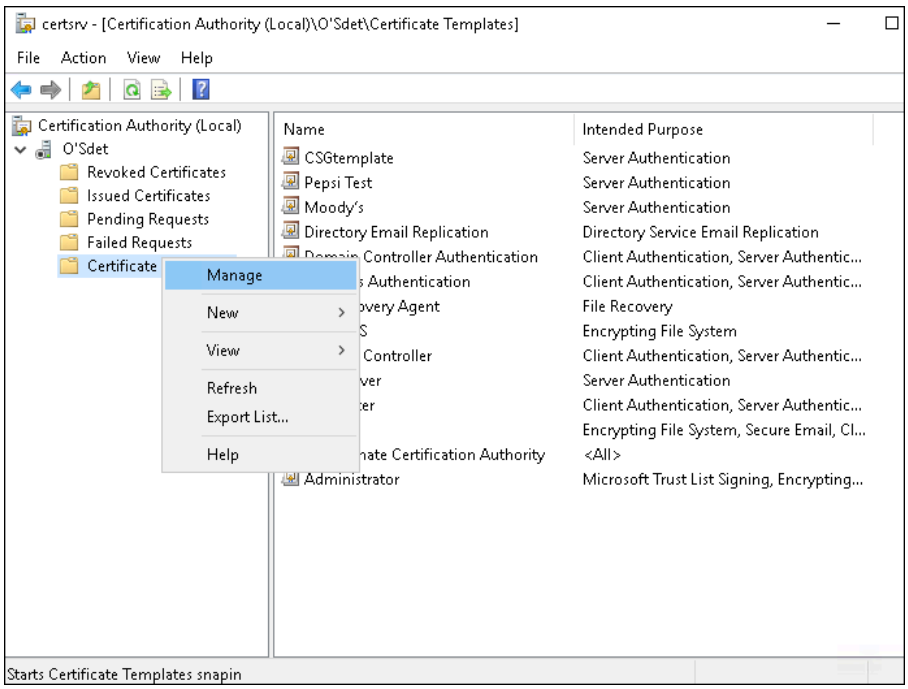
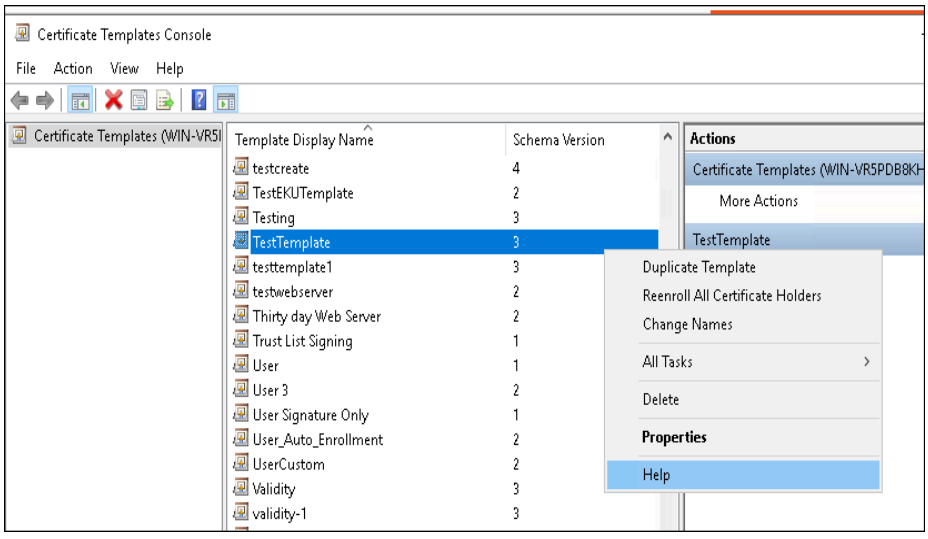


Copyright © 2025 AppViewX Inc. All Rights Reserved.

Denied by policy module

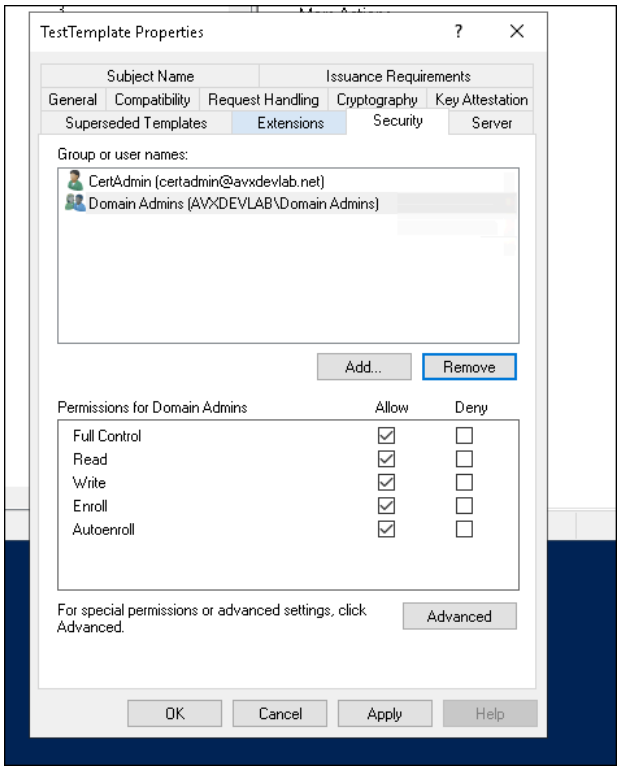


Copyright © 2025 AppViewX Inc. All Rights Reserved.

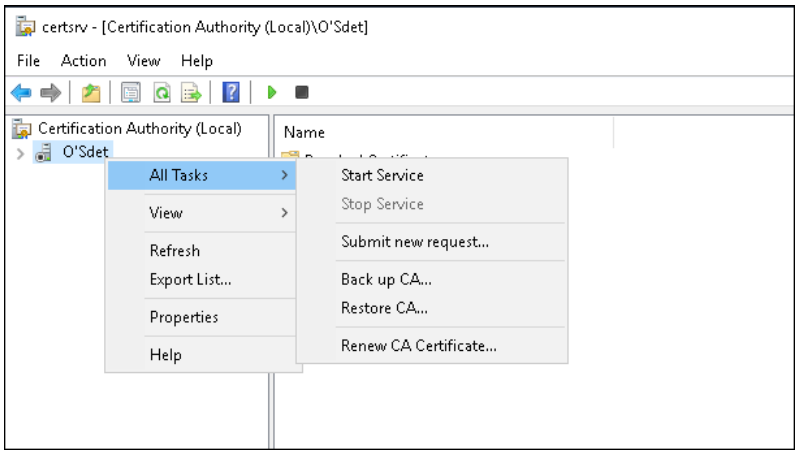
Potential Causes	This error can occur during Certificate Enrollment, if the template used during enrolment is not provided the required permissions for the requesting user or service account.
Troubleshooting Steps	1. **Access Certificate Templates:** Go to the Certificate Authority (CA), right-click on the **Certificate Templates** folder and select **Manage**.  2. **Locate the Template:** From the list of available templates, identify the specific template associated with the failed enrollment. Now, right-click on the template and select **Properties**. 

Copyright © 2025 AppViewX Inc. All Rights Reserved.

3. **Verify Security Settings:** Go to the **Security** tab, ensure that the user or group has the appropriate privileges for enrollment.



4. **Restart the CA Service**: First stop the service and then restart the CA service to apply the changes.



5. **Retry Enrollment:** After permissions have been updated, retry the certificate enrollment process.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

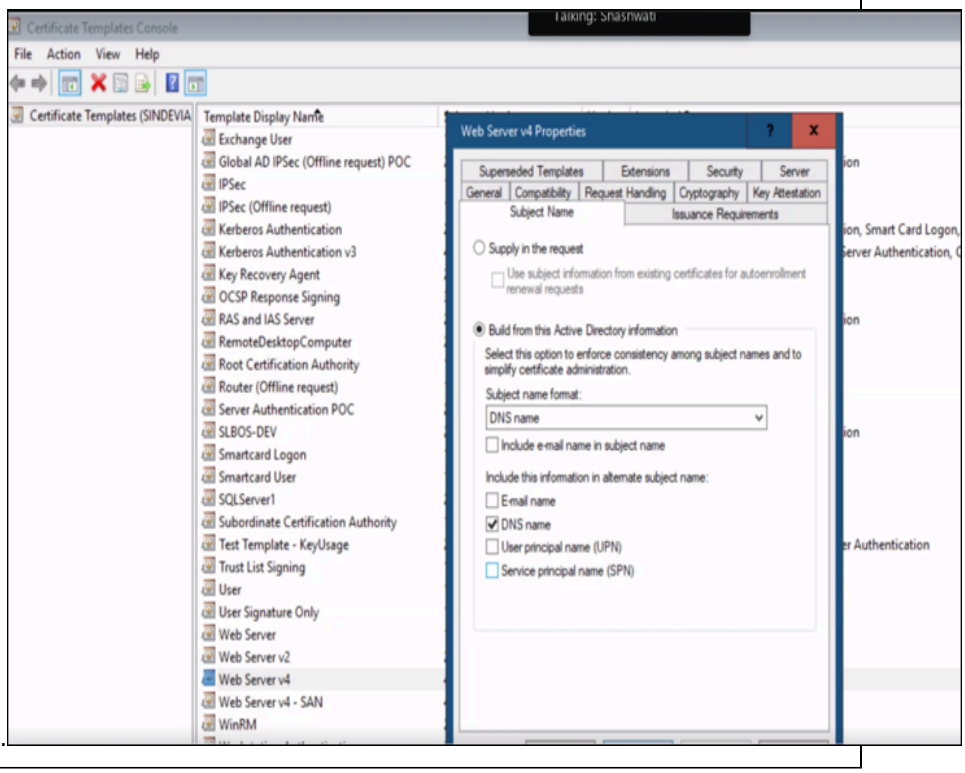
Curl test from port 8999 Fails

Description	The installation completed successfully, and the service is running; however, the cURL test on port 8999 is still failing, and the AppViewX Help site remains unreachable.
Potential Causes	This problem occurs when certificate binding fails during installation.
Troubleshooting Steps	To verify if the certificate installation was successful, please review the installation log file located in the directory where the installer executable file was extracted.

Denied by Policy Module error during Certificate Enroll

Description	When enrolling a certificate, the **Denied By Policy Module** error occurs if the Certificate Authority (CA) rejects the request due to policy validation failures.
Potential Causes	If the **Build from Active Directory** option is selected in the template, the CA attempts to verify the request's source. When using a **service account**, Active Directory (AD) tries to validate the associated properties, but certain required attributes may be missing. As a result, MSCA denies the request.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Troubleshooting Steps	To resolve this issue, it is recommended to use the **Supply in CSR** option instead of **Build from Active Directory** when configuring the certificate template. 

FAQ: Installation and Configurations

1. **Where to download the Windows Agent Installer Setup?**
   The most recent Windows agent installer is available at https://release.appviewx.com under the latest version of AppViewX, in the **Additional Resources** section.
2. **How to install AppViewX Windows Agent?**
   Extract the downloaded package from the portal and execute the **AppViewX.CertPlusInstaller.exe** and follow the onscreen instructions to install the agent.

   For a detailed step by step installation guide refer to the section Installing the AppViewX Windows Gateway.
3. **How to verify the installation of the Windows Agent?**
   Post installing the Windows Agent, to verify the installation status of the Windows Gateway Service refer to the section Step 4: Verifying the AppviewX Windows Gateway Installation.
4. **How to Uninstall Windows Agent?**
   Refer to section Uninstalling the AppViewX Windows Gateway
5. **How to enable PowerShell remoting in a target machine?**

Copyright © 2025 AppViewX Inc. All Rights Reserved.

a. Open PowerShell as **Administrator** and run the command below in the required remote or target machine.

```
Enable-PSRemoting
```

b. As a pre-requisites to establish a remote PowerShell session, allow **Port 5985** in all firewalls in both the windows agent machine and the remote machine.

c. Open an administrator PowerShell and run the command below from the Windows Gateway Machine to the target machine

```
Enter-PSSession -computerName <hostname> -credential <username>
```

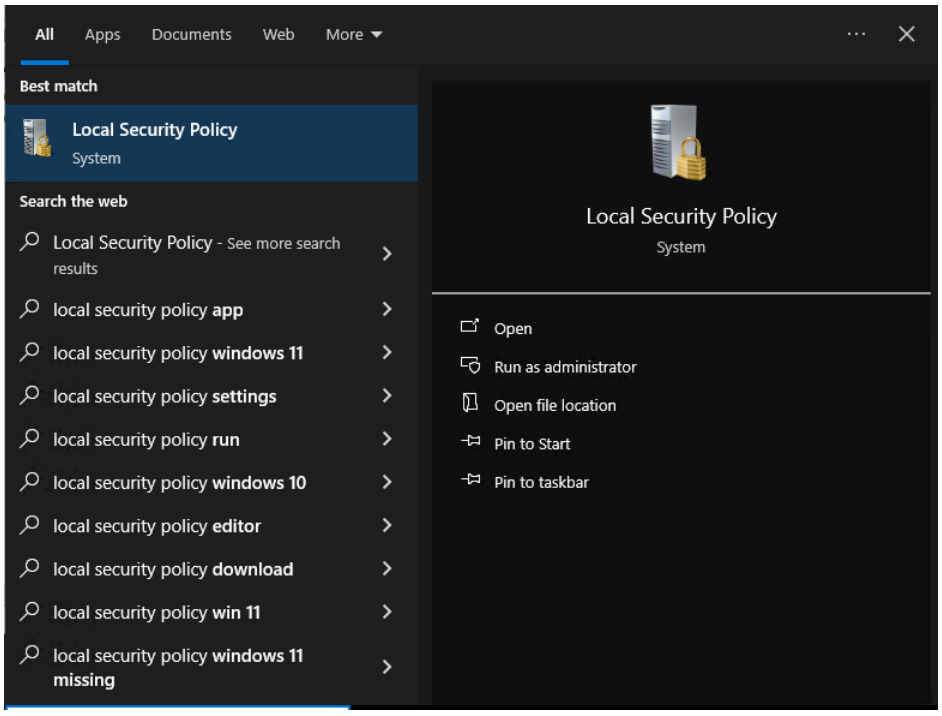
In the *<hostname>* add the hostname of the remote machine itself, and in the *<username>* enter the username of the service account.

Note: If the first step doesn't work, please try command below to enable PowerShell remoting.

```
winrm quickconfig
```

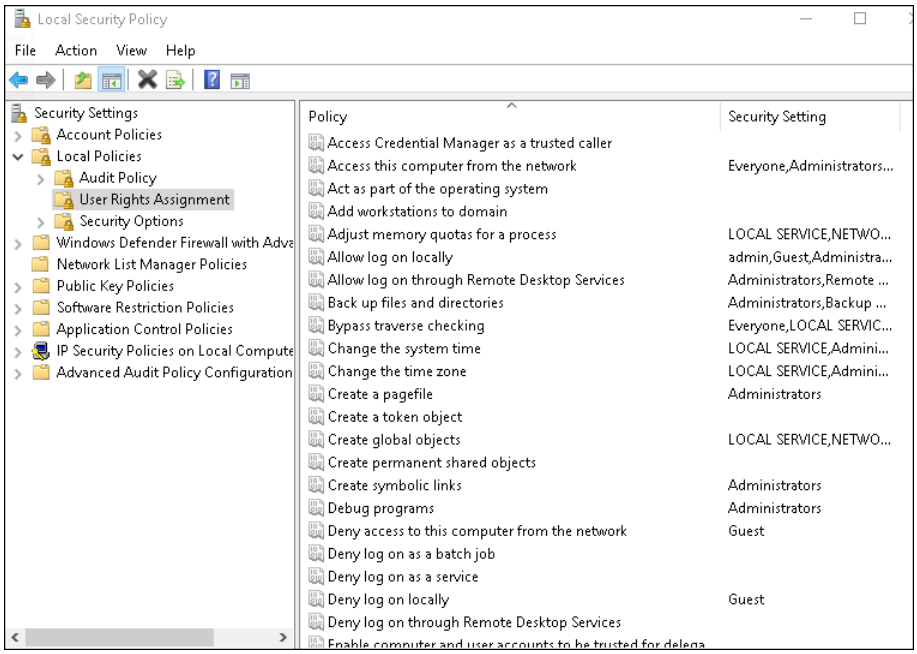
6. **How to check the Logon locally permission for a user?**

a. In the start menu, search for **Local Security Policy**.

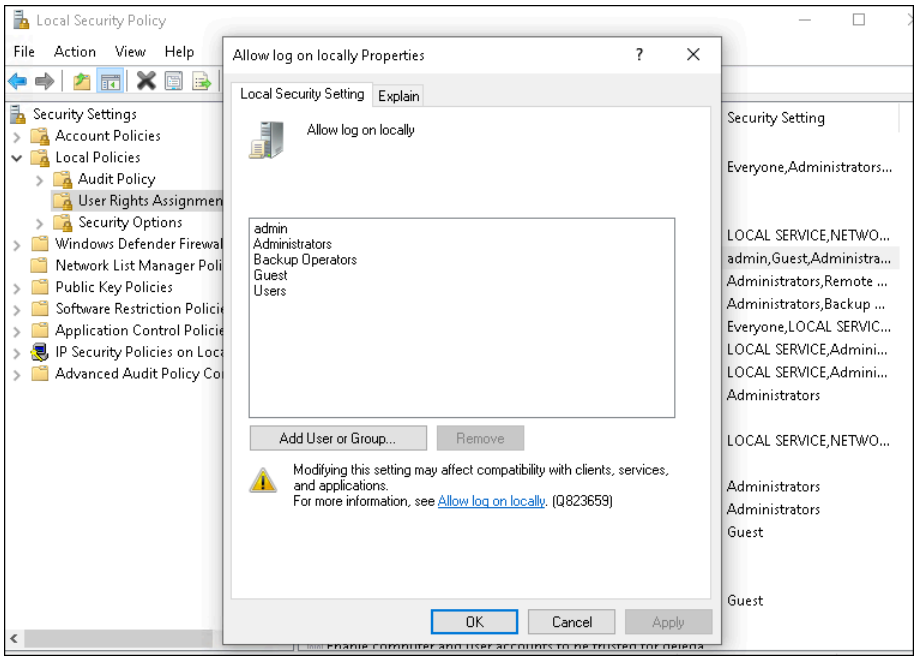


b. Open the **Local Security Policy** > **User Rights assignment**.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



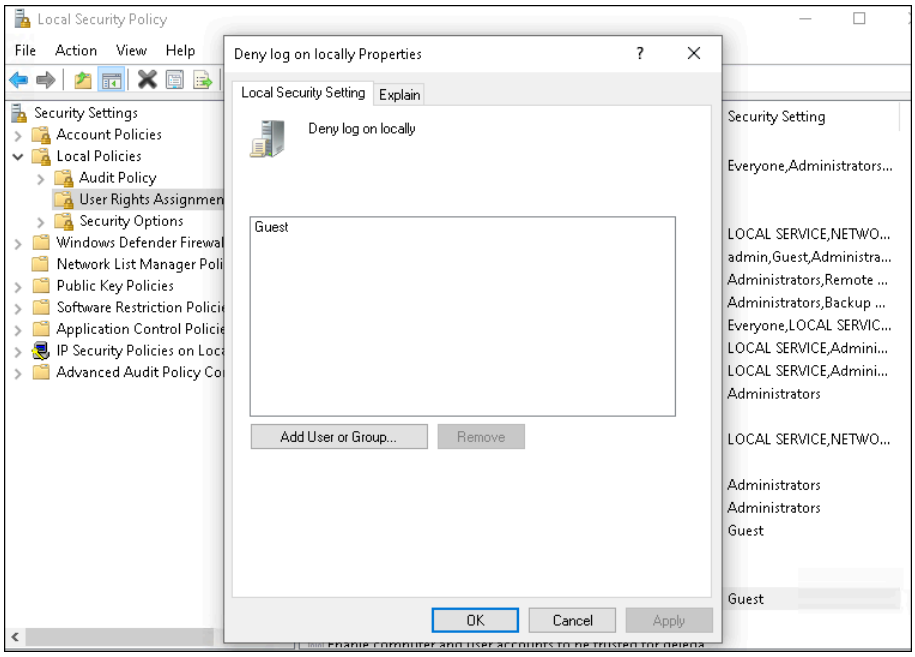
c. Double click on **Allow logon locally**.



d. Check if the service account user is present in the list as seen in the image above.

e. Close the **Allow logon locally** properties, and double click on **Deny log on locally**.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



f. Check if the user is in the **Deny logon locally** policy list.

> **Note:**
> - Deny logon locally takes precedence over the allow logon locally. So if the account is in deny logon locally, it will take precedence and logon locally won't be present for the account.
> - This setting can be disabled or greyed out due to Group Policy Settings. Check the Group Policy Settings or try providing the permission from the Domain Controller Machine.

7. **How to enable two-way trust between two different domains?**

   Refer to secion Enabling Two-way trust between Two Different Domains.

8. **How to check if any port is blocked?**

   Run the command below in Windows PowerShell. Replace *<port>* and *<hostname>* with actual values.

```
Test-NetConnection -ComputerName <hostname> -Port <port>
```

   Sample expected output:

```
ComputerName : computername

RemoteAddress : 93.184.216.34

RemotePort : 80

InterfaceAlias : Ethernet

SourceAddress : 192.168.1.10

PingSucceeded : True

TcpTestSucceeded : True
```

Copyright © 2025 AppViewX Inc. All Rights Reserved.

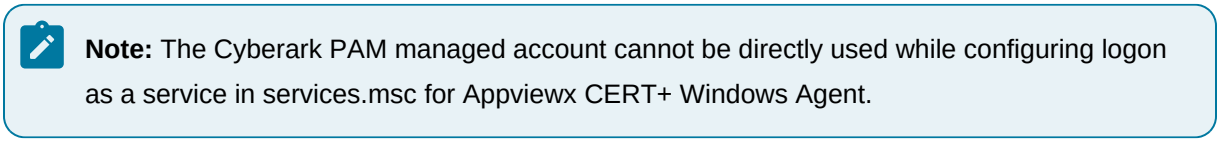
9. **What are the types of users supported by AppViewX Windows Gateway Agent?**

   The Windows Gateway Agent supports the following types of users for managing the devices:

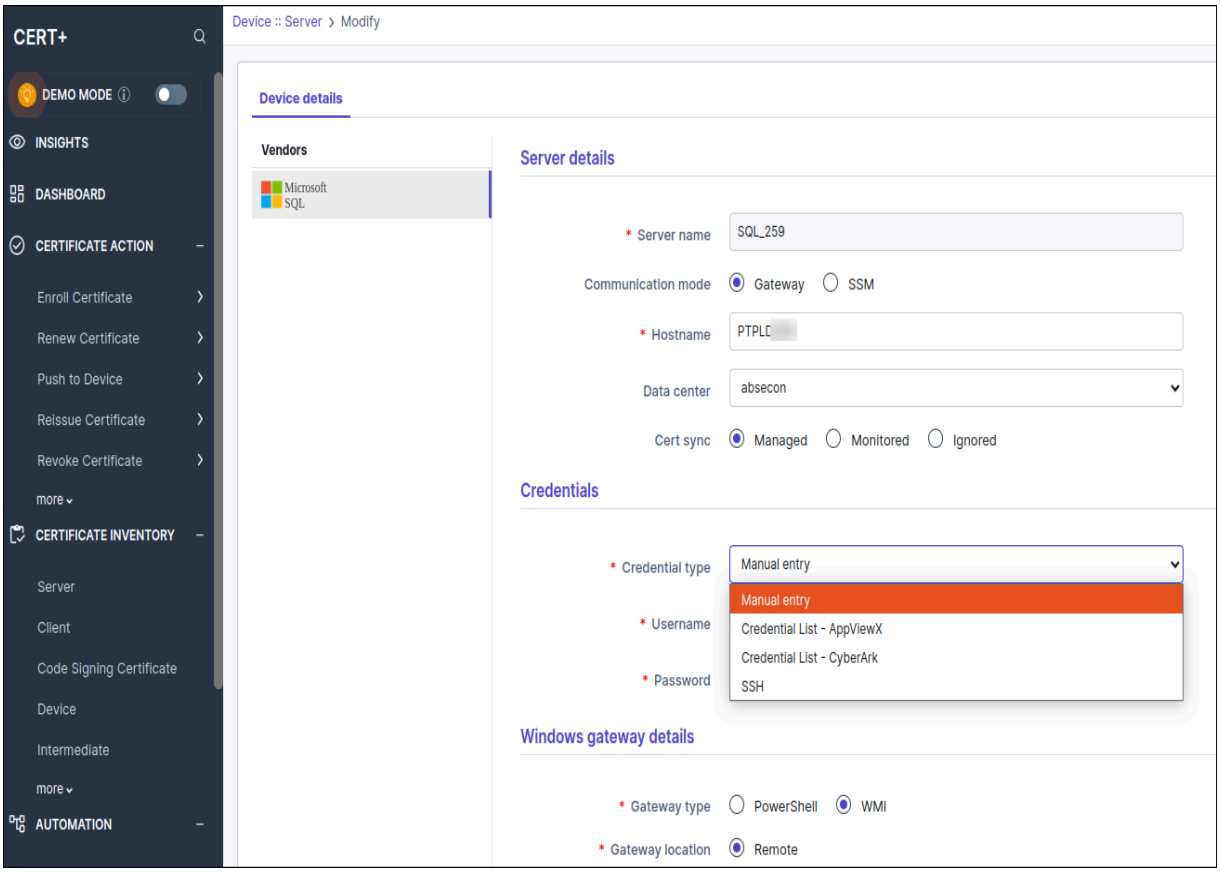
   - **Domain User with Local Admin Permissions**
     - A domain user account must belong to the organization's Active Directory domain.
     - The account should have **local administrator privileges** on the target machine and the machine where the agent is installed.
   - **Service User Account with Local Admin Permissions**
     - It must be assigned **local administrator permissions** on the target machine and the windows agent machine for executing privileged operations required by the agent.
   - **Group Managed Service Accounts (gMSAs)**
     - These are specialized **managed service accounts** designed for services running on domain-joined machines.
     - gMSAs automatically manage passwords and do not require manual password management.
     - They must have **local administrator permissions** on the windows agent machine and the target machine to perform the required operations for the agent.
     - For more info about GMSA: Group Managed Service Accounts Overview | Microsoft Learn
10. **Can CyberArk managed accounts be used for onboarding devices with AppViewX Windows Gateway Agent?**

    Yes, it is supported.

    - To use CyberArk-managed accounts for onboarding devices with the AppViewX Windows Gateway Agent, the CyberArk PAM module needs to be configured within the AppViewX platform.
    - While onboarding the device online, select the **CyberArk** option and choose the corresponding CyberArk-managed account. This ensures secure and streamlined access during the onboarding process.



Copyright © 2025 AppViewX Inc. All Rights Reserved.



11. **Why are CAs from other domains not listed when fetching CA names, even after enabling two-way trust between domains?**
    Even with two-way trust enabled between domains, CA details from any domain other than the one where the gateway is installed cannot be retrieved. This limitation arises because the built-in Windows tool, **certutil**, by default does not support fetching CA details from a domain other than where the gateway is installed.

Quick References

1. PowerShell Remoting Requirements| Microsoft Learn
2. User Account Control and WMI - Win32 apps | Microsoft Learn
3. Allow log on locally - security policy setting - Windows 10 | Microsoft Learn
4. AppViewX Helpcenter - Windows Gateway Service Installation and Pre-requisites
5. AppViewX Helpcenter - Windows Gateway Powershell Mode Requirements
6. AppViewX Helpcenter - Windows Gateway WMI Mode Requirements
7. AppViewX Helpcenter - Windows Gateway Native API Mode Requirements

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Chapter 15: 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

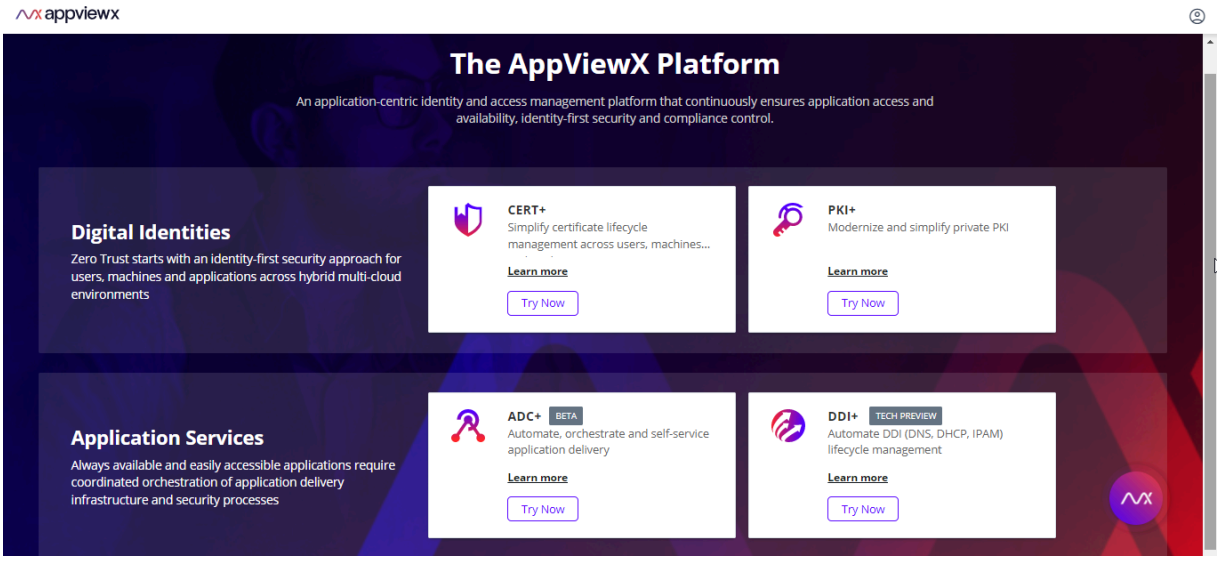
• AVXpert AI Assistant

Using the AppViewX Chatbot

To access the chatbot:

1. Log in to your SaaS account.

   The AppViewX Platform landing page is displayed.

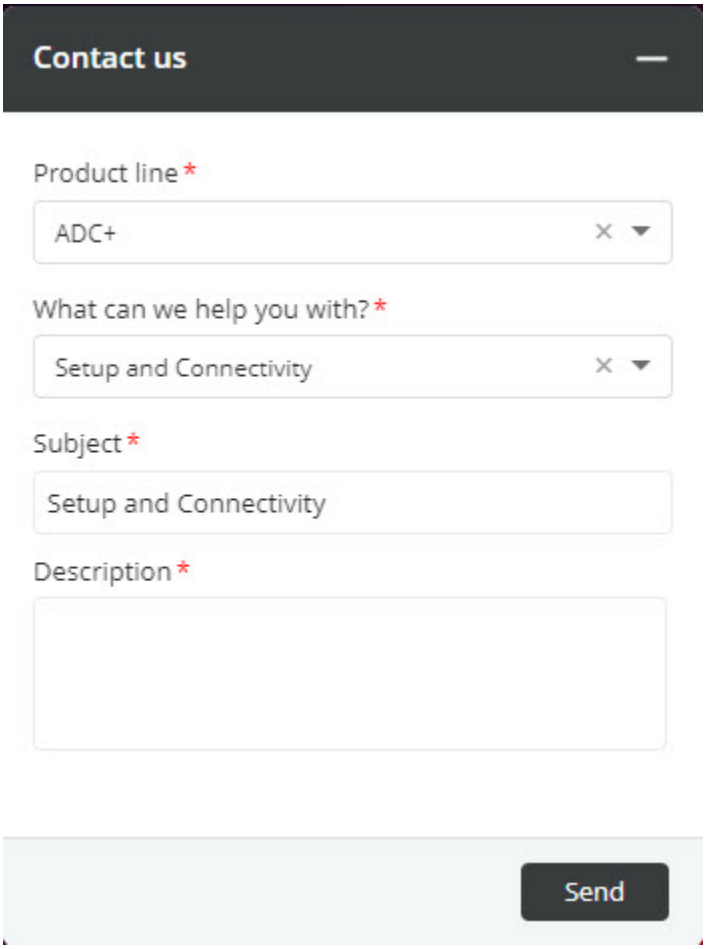


2. To access the AppViewX chatbot, click  from the bottom-right corner of the screen.

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

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



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

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

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Field	Description
	• 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

AVXpert AI Assistant

AVXpert provides the AI-based smart Help, a conversational search feature that helps you find relevant content from AppViewX documentation.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

You can ask your queries and get answers in an informative way. The answer to your query is based on the content from the product documentation. This search is entirely conversational. Based on the response, you can also ask further questions. The response is from source documentation, which you can refer to for details on helpcenter.

Note: AVXpert AI assistant is disabled by default.

When you enable AVXpert AI Assistant from RBAC the 

(AVXpert) icon appears on the bottom right. The **Doc Hub** tab opens by default, under **Actions** tab you can explore for **Smart search** and **Quick navigation** capabilities, under **History** tab you have all the list of previous search queries on AVXpert.

Enable ACF Permission for AVXpert

To enable the ACF permission for roles to access the AVXpert AI assistant, follow these steps:

1. Go to Platform module **IDENTITY** > **Role**.
   You will be redirected to the **Role** page.
2. Click on the role name to enable the ACF permission.
   You will be redirected to the **Modify :: [RoleName]** page, with the **Information** tab open by default.
3. Switch to the **Authorized Functions** tab and expand the **AppViewX Buddy** by clicking **(Expand)** icon.
4. To assign an AVXpert AI assistant, select the checkbox beside the **AppViewX Buddy**.
5. Click **Save**.
   The ACF permission will be successfully updated and AVXpert is enabled.

Doc Hub

View the ***AVXpert*.**



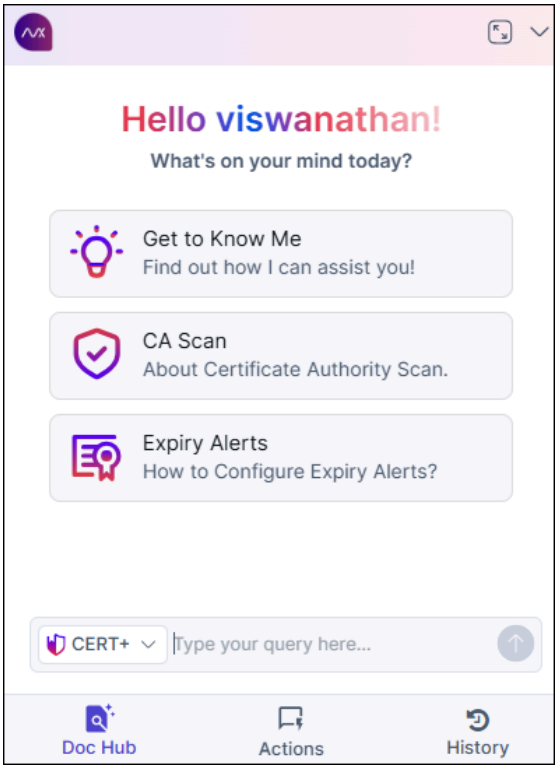
Perform the following steps to use the **AVXpert** to find appropriate content and resolve your queries:

1. Click the **AVXpert** icon to open the **Help panel**.

Note: AVXpert icon is displayed, only if the AVXpert ACF permission is enabled for the **user/ role**.

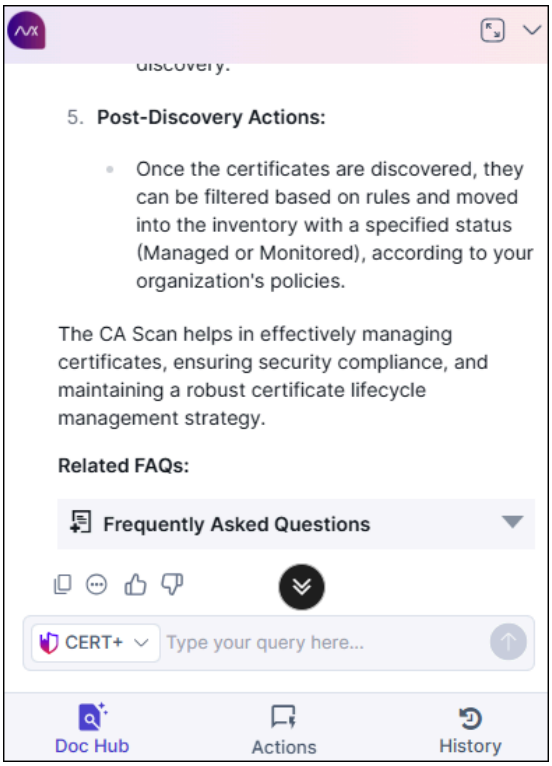
Copyright © 2025 AppViewX Inc. All Rights Reserved.

2. There are few predefined queries on how AVXpert can assist. Select the **product** from the dropdown. You can select the default question in the panel or type your question in the text box to find the related content in the **AVXpert** documentation.



3. Select **Send** icon or press **Enter** to view the response to your questions.
   Depending on your question, you may view the content, applicable images, and links to the **Frequently Asked Questions** articles.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



Expand the **Frequently Asked Questions** and view the related content links in response.

4. Select the **History** tab to view the conversation history from the panel. You can click on the previous conversation and find relevant content.

   Instead of searching through user guides and reference documents, you can use the **AVXpert** feature to quickly find relevant answers to your queries. This helps save time and allows you to focus on content, resulting in enhanced productivity and efficiency.

Actions

View the ***AVXpert*.**

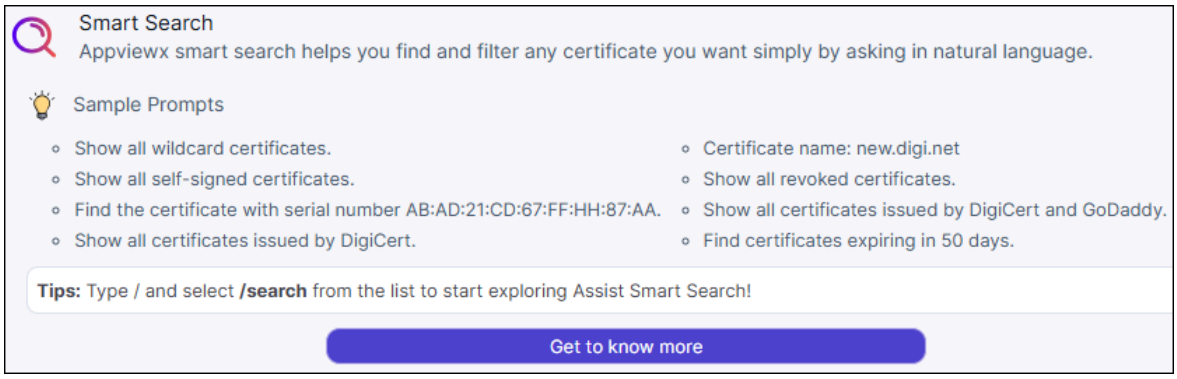


Perform the following steps to use the **AVXpert** to find appropriate Smart search and Quick navigation:

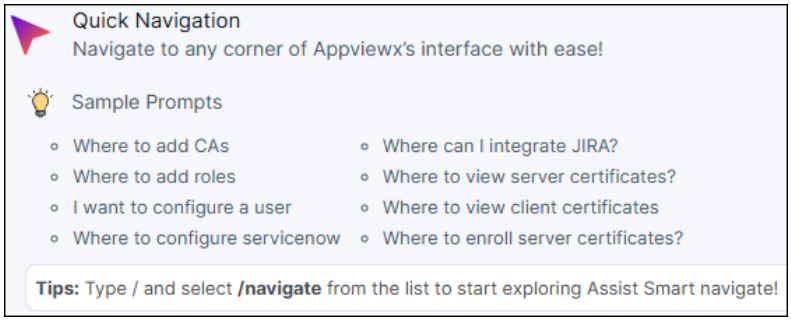
1. Click the **AVXpert** icon to open the **Help panel**.
   The **Doc Hub** tab opens by default.
2. Click the **Actions** tab.
   You can Experience effortless certificate search and seamless navigation using **Smart Search** and **Quick Navigation** within the system.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

- **Smart Search:** Appviewx smart search helps you find and filter any certificate you want simply by asking in natural language.

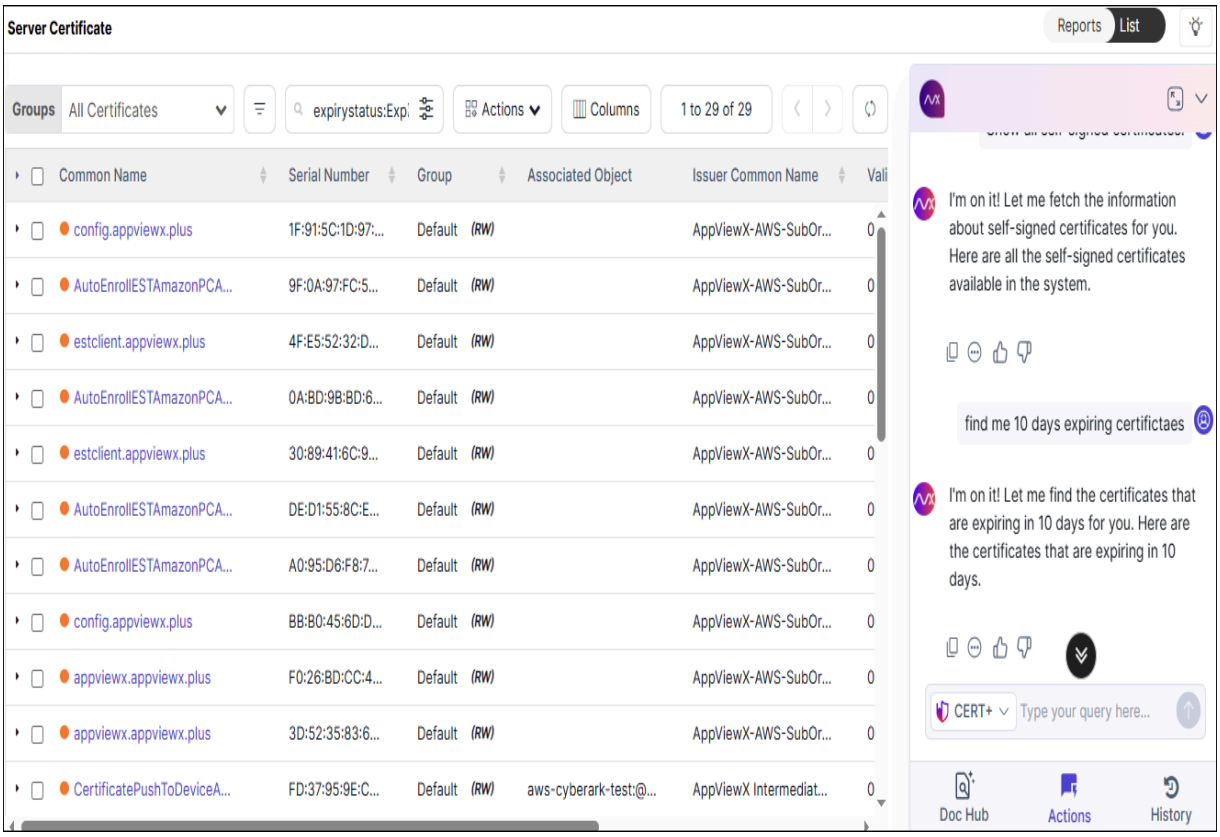


- **Quick Navigation:** Navigate to any corner of Appviewx's interface with ease!



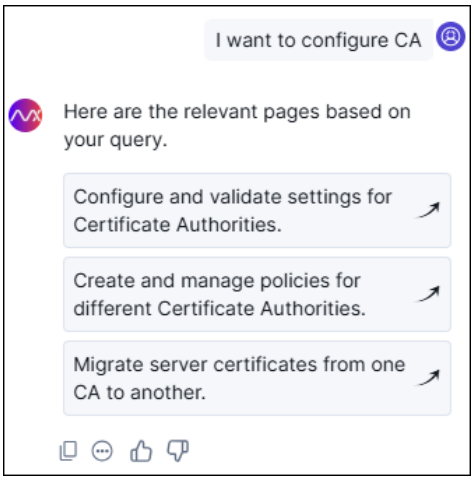
3. Type your question in the text box to find the related smart search or quick navigation in the AVXpert **Actions**.
4. Select **Send** icon or press **Enter** to view the response to your questions.
   Depending on your question, you may view the quick navigation route links or Smart Search results displayed on the main screen.
5. The system effectively interprets user requests, executes the appropriate search query, and directs users to the relevant inventory information.

Copyright © 2025 AppViewX Inc. All Rights Reserved.



Or

Navigation search results provide links to relevant routes.



6. Click the corresponding route link to navigate to the desired location.
7. To view previous conversations and relevant searches, select the **History** tab on the panel.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

History

The **History** tab is a useful tool that keeps a record of your previous conversations and searches within the platform. This allows you to reference past activity, promoting a more informed and efficient user experience. To effectively manage storage resources, search history is retained for 7 days.

Copyright © 2025 AppViewX Inc. All Rights Reserved.

Chapter 16: Glossary

An explanation of the terms used in this guide:

Term	Description
SaaS	Software as a Service
EKS	Elastic Kubernetes Service
TLS	Transport Layer Security
AES	Advanced Encryption Standard
AZ	Availability Zone
VPN	Virtual Private Network
VPC	Virtual Private Cloud
EC2	Elastic compute
AWS	Amazon Web services
HA	High Availability
DR	Disaster Recovery
mTLS	Mutual TLS Authentication

Copyright © 2025 AppViewX Inc. All Rights Reserved.