



Install, Upgrade, and Maintenance Guide

Version: 2020.3.0

Copyright AppViewX, Inc.

Copyright © 2022 AppViewX, Inc. All Rights Reserved.

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

Trademarks

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

External Reference Links

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

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

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

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

Contact Information

AppViewX, Inc.

222 Broadway, FL 19

New York, NY 10038

Email: info@appviewx.com

Web: www.appviewx.com

Contents

Preface.....	8
Revision History.....	8
About this Guide	8
Audience.....	8
Text Conventions.....	8
Chapter 1. Overview.....	10
Introduction.....	10
What's New.....	11
AppViewX Architecture.....	11
Benefits of AppViewX.....	12
Supported Deployment Methods and Types.....	13
Understanding the Installation Steps.....	19
Chapter 2. Working with Prerequisites.....	20
Understanding Requirements.....	20
Understanding Hardware Requirements.....	20
Understanding Software Requirements.....	21
Working with Prerequisites.....	21
Configuring Elevated Access.....	22
Configuring Firewall Ports.....	22
Downloading Linux Packages.....	25
Configuring YUM.....	27
Configuring SELinux.....	29
Configuring NTP.....	30
Configuring Ulimit.....	30
Increasing vm.max_map_count.....	30
Enabling IP Forwarding.....	31
Enabling Bridging.....	31

Configuring Calico before Deployment.....	31
Enabling the IP in IP Protocol.....	32
Downloading AppViewX Packages.....	33
Running the Prerequisite Tool.....	34
Chapter 3. Deploying the AppViewX Virtual Appliance.....	37
Deploying the AppViewX Virtual Appliance.....	37
Download the Release Package.....	37
Install the AppViewX OVA.....	38
Chapter 4. Deploying Appviewx in AWS using Appviewx provided AMI.....	49
Deploying Appviewx in AWS using Appviewx provided AMI.....	49
Create AWS Instance Using Appviewx AMI.....	49
Mount Additional Storage for Worker Node.....	52
Chapter 5. Installing AppViewX.....	60
Installing AppViewX.....	60
Performing a Single Node or Standalone Installation.....	61
Performing a Multi-node or High Availability Installation.....	63
Configuring the appviewx.conf File to Install AppViewX.....	63
Configuring POD and Service IP CIDR.....	71
Installation Support for 3 Nodes and 2 Datacenters.....	73
Enabling the Load Balancer for the Kube API Server.....	73
Verifying the Installation.....	76
Uploading the License Key.....	77
Adding Third-party Libraries.....	78
iControl F5 Integration.....	79
Thales.....	80
Safenet/Gemalto.....	80
Accessing the AppViewX Graphical User Interface.....	81
Installing a Fix Pack.....	83
Chapter 6. Monitoring and Maintaining AppViewX.....	85

Monitoring and Maintaining AppViewX.....	86
Installing ELK Components.....	87
Executing Commands for Maintenance.....	88
Installing Trusted Certificate for GUI/API Access.....	90
Enabling Strict Data Center Routing.....	92
Enabling Device Syslog Processing.....	93
Enabling the Insight Module.....	95
Understanding Commands Executed during Installation.....	98
Enabling Sudo Access.....	99
Creating a New Sudo User.....	99
Adding Users to the Sudo Group.....	99
Verifying if the Wheel Group is Enabled.....	100
Adding a User to the Wheel Group.....	100
Switching to the Sudo User.....	100
Adding a Node to the Cluster.....	101
Understanding the Best Practices on Reboot Sequence.....	104
Working with Alerts.....	105
Enabling an email Alert.....	105
Troubleshooting Alerts.....	106
Working with Backup and Restore.....	106
Downloading the Scripts.....	106
Performing a Backup for MongoDB and Vault.....	107
Restoring a MongoDB Backup.....	108
Restoring the Vault Backup.....	108
Troubleshooting Backup and Restore Operations.....	108
Working with Logs.....	108
Managing Logs using Kibana.....	109
Managing Logs using AppViewX Nodes.....	119
Offline Patching for CentOS.....	120

Working with Plugins.....	121
Adding a New Plugin.....	121
Removing a Plugin.....	122
Restarting a Plugin.....	123
Scaling a Plugin.....	124
Changing the Memory for a Plugin.....	125
Working with the Management Console.....	126
Accessing the Management Console.....	126
Viewing the POD Status.....	128
Accessing the POD Console.....	129
Accessing the Database Command Line.....	129
Exporting a Database Collection.....	131
Deleting a Node from the Cluster.....	132
Chapter 7. Upgrading AppViewX.....	134
Infra Readiness.....	134
Backups and VMSnapshots.....	134
Upgrading AppViewX.....	135
Chapter 8. Uninstalling AppViewX.....	141
Troubleshooting Uninstall Issues.....	141
Uninstalling AppViewX.....	141
Chapter 9. Troubleshooting.....	142
Troubleshooting.....	142
AppViewX Installation Failed.....	142
Common Installation Errors.....	142
Frequently Faced Errors.....	142
General Troubleshooting.....	143
Pods not started after installation.....	144
Unable to login.....	144
Error while downloading certificates.....	145

Preface

Revision History

Revision	Description	Date
1.0	Initial release of document for Release 2020.3.0	September 2020

About this Guide

This document covers the installation, upgrade, and maintenance activities for AppViewX. The document is divided into the following sections:

- **Overview** - provides an introduction to the product and explains the basic features.
- **Working with Prerequisites** - explains the prerequisites needed to install AppViewX.
- **Deploying the AppViewX Virtual Appliance** - describes the procedure to install AppViewX using the default settings.
- **Installing AppViewX** - describes the procedure to install AppViewX using the command line interface.
- **Monitoring and Maintaining AppViewX** - explains the functions available to maintain and monitor AppViewX.
- **Uninstalling AppViewX** - describes the procedure to safely remove AppViewX from the system.
- **Troubleshooting Installation** - explains the steps to be performed in case of issues encountered during the installation of AppViewX.

Audience

This guide is intended for the following audience:

- Network Engineers
- Service Engineers
- Customer Support Executives
- System Administrators

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.

Convention	Description
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
<code>codeblock</code>	Indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Chapter 1: Overview

- [Introduction](#)
- [What's New](#)
- [AppViewX Architecture](#)
- [Benefits of AppViewX](#)
- [Supported Deployment Methods and Types](#)
- [Understanding the Installation Steps](#)

Introduction

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 helps users:

- Manage ADC devices
- Manage certificates

In order to perform the above functions, AppViewx provides the following modules:

- ADC
- CERT+
- Platform
- Security
- Automation

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.

What's New

This section provides information about the features and the enhancements in 2020.3.0. The features are described in detail in the Release Notes. For more information, refer to the Release Notes.

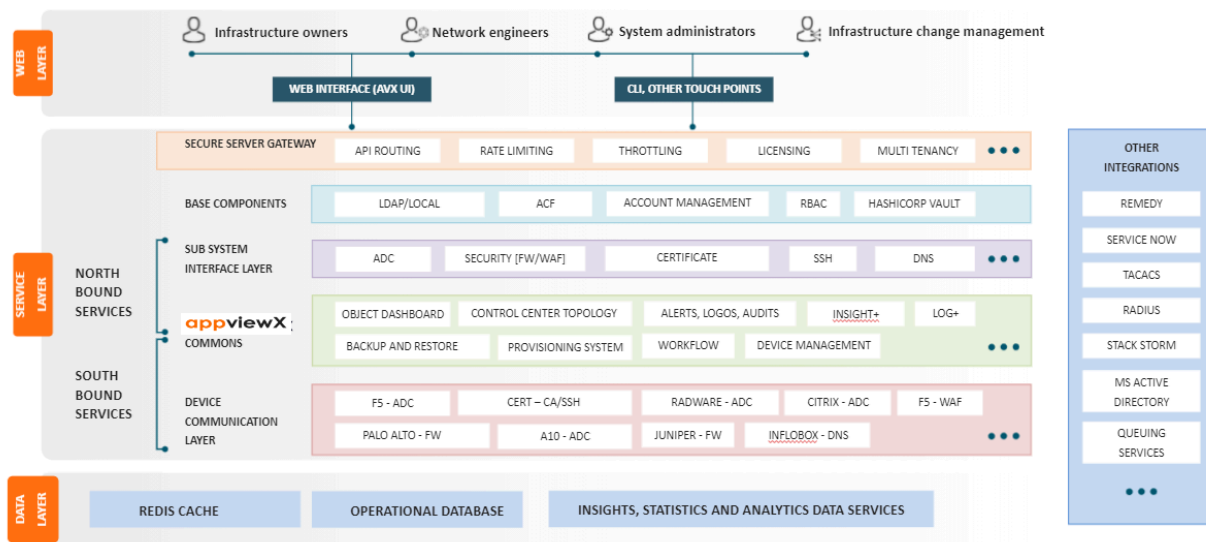
AppViewX Architecture

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.

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:

Architecture - Explained

appviewX



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

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.

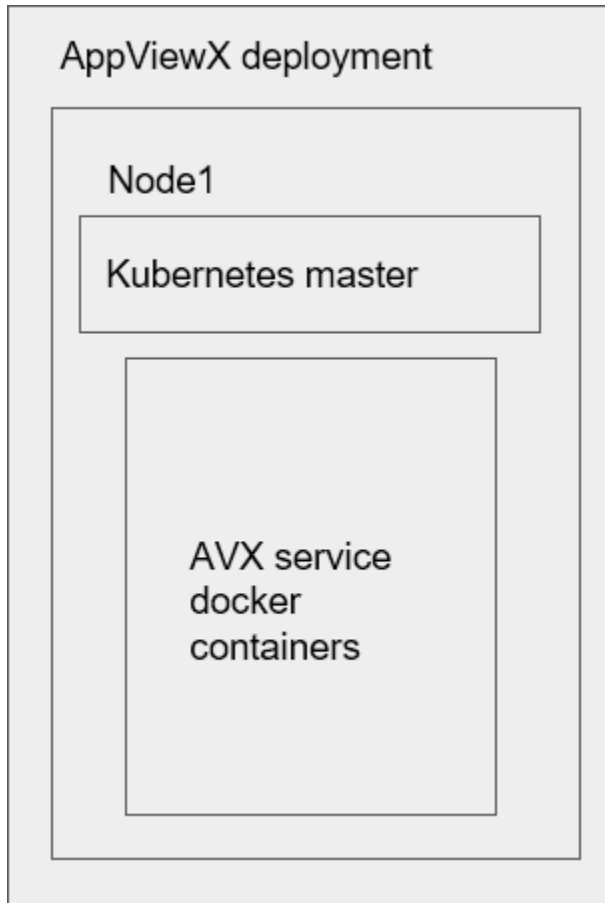


Warning: Hybrid cloud management deployment is not supported in AppViewX.

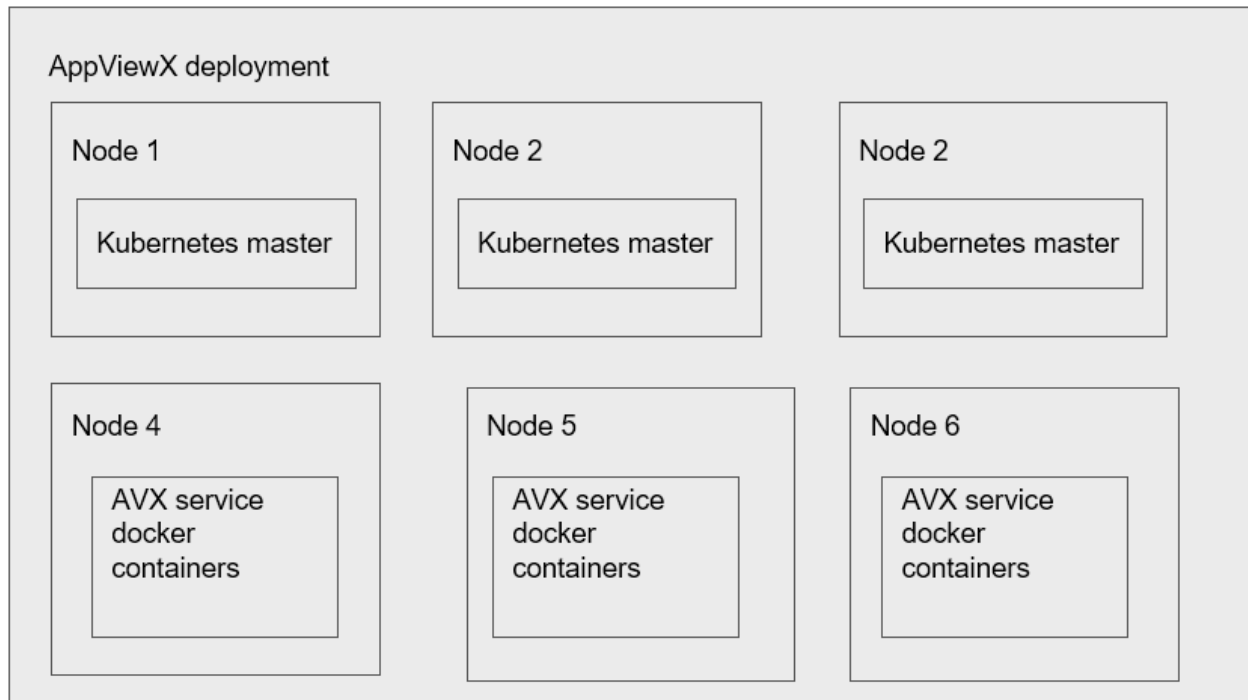
First, AppViewX can be deployed in the following modes:

- **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.
- **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:



Single Node Deployment



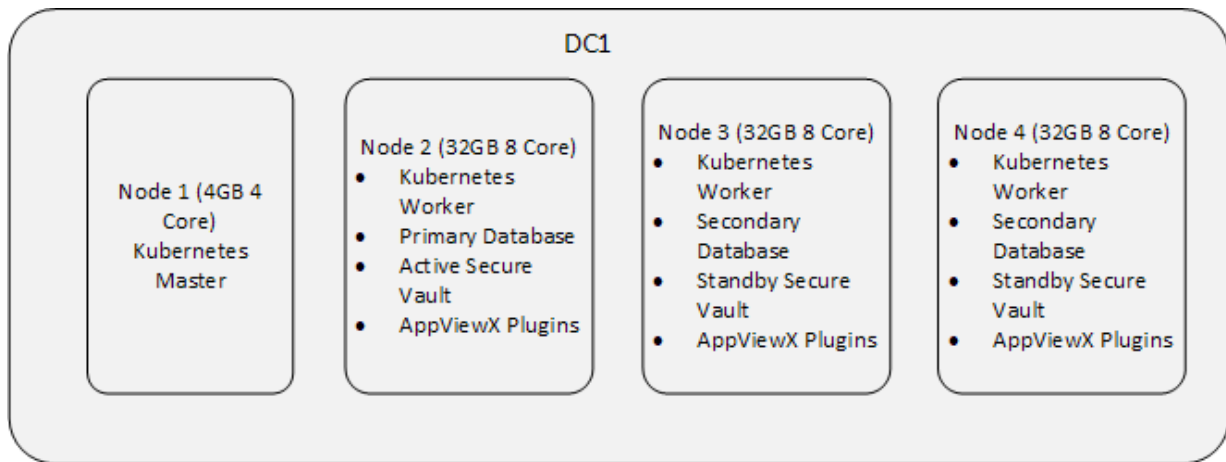
Multi Node Deployment

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

- **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.

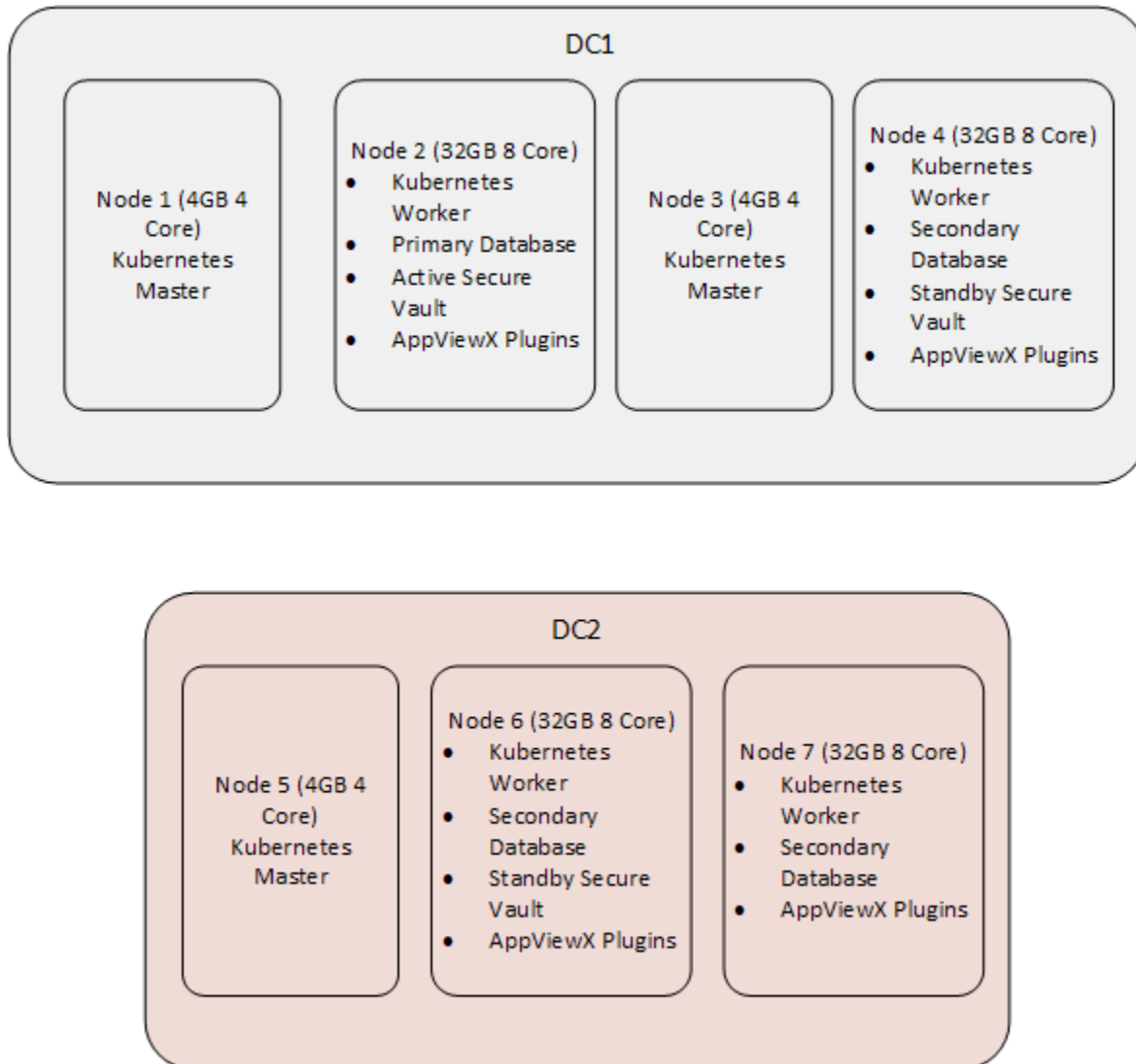
AppViewX supports the following deployment types/scenarios:

- One Data Center and Four Nodes



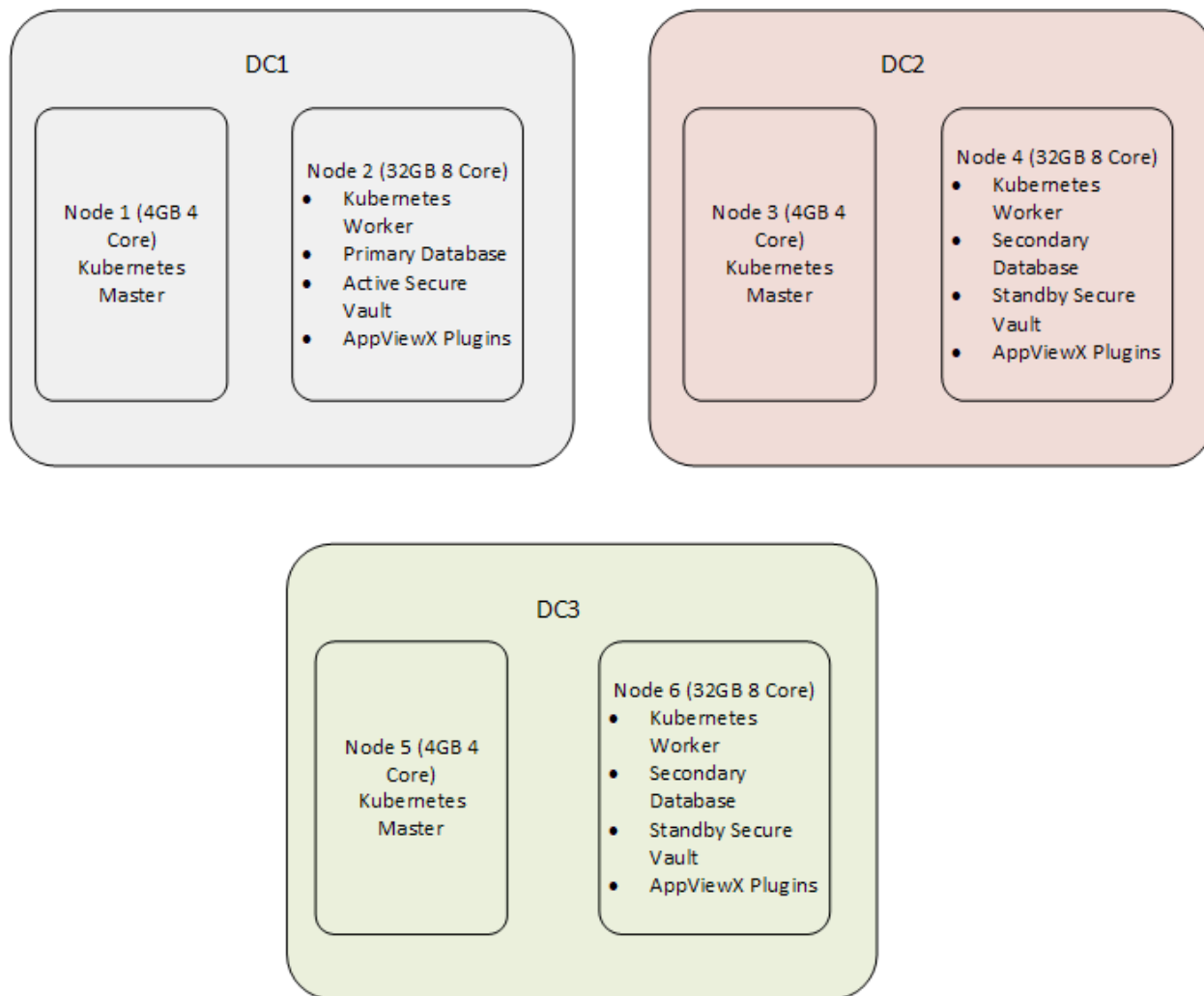
This deployment is recommended for customers who need only HA at the application level. This deployment does not support HA; neither for the Kube nor for the DC. This deployment is best suited for less than 50 ADC devices having a total of 100,000 objects and 10,000 certificates.

- Two Data Centers and Seven Nodes



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



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.

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

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.

Understanding the Installation Steps

This section outlines the various mandatory and optional steps in the process of installing AppViewX.

Step No	Step Name	Mandatory	Optional
1	Working with Prerequisites	Yes	No
2	Configuring Firewall Ports	Yes	No
3	Configuring Elevated Access	Yes	No
4	Downloading Linux Packages	Yes	No
5	Downloading AppViewX Packages	Yes	No
6	Running the Prerequisite Tool	Yes	No
7	Deploying the AppViewX Virtual Appliance	No	Yes
8	Performing a Single Node or Standalone Installation	No	Yes
9	Performing a Multi-node or High Availability Installation	No	Yes
10	Configuring the Cluster	No	Yes
11	Configuring the POD and Service IP CIDR	No	Yes
12	Verifying the Installation	Yes	No
13	Uploading the License Key	Yes	No
14	Accessing the AppViewX Graphical User Interface	Yes	No
15	Adding Third-party Libraries	No	Yes

Chapter 2: Working with Prerequisites

- [Understanding Requirements](#)
- [Working with Prerequisites](#)

Understanding Requirements

- [Understanding Hardware Requirements](#)
- [Understanding Software Requirements](#)

Understanding Hardware Requirements


Ensure that you have, at minimum, the following hardware with the given specifications before proceeding with the installation:

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

Understanding Software Requirements

Both single node and multi-node installations of AppViewX are supported on the following operating systems:

- CentOS 7.X
- RHEL 7.X
- RHEL 8.5
- RHEL 8.6
- RHEL 8.7
- RHEL 8.8

Working with Prerequisites

- [Understanding Requirements](#)
- [Working with Prerequisites](#)
- [Configuring Elevated Access](#)
- [Configuring Firewall Ports](#)
- [Downloading Linux Packages](#)
- [Configuring YUM](#)
- [Configuring SELinux](#)
- [Configuring NTP](#)
- [Configuring Ulimit](#)
- [Increasing vm.max_map_count](#)
- [Enabling IP Forwarding](#)
- [Enabling Bridging](#)
- [Configuring Calico before Deployment](#)
- [Enabling the IP in IP Protocol](#)
- [Downloading AppViewX Packages](#)
- [Running the Prerequisite Tool](#)

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.

S.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.
4	All Nodes	Any	All Nodes*	10250	HTTPS	TCP	Used by Kubelet Agent which exposes Rest endpoints for the Kubernetes API Server.
5	All Nodes	Any	All Nodes*	4243	HTTP	TCP	Required during installation and scaling up. Used to load docker images when spinning up a new container. Triggered by the node where the install process is started.No sensitive data is being transferred through this port.
6	Load Balancer (for ex, F5, GCP, etc.)	Any	ISTIO Ingress Proxy IP (Kube Worker)	31443	HTTPS	TCP	To access the AppViewX web user interface.
7	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).
8	All Nodes	Any	F5 VIP	6443		TCP	To allow all the nodes to communicate with the Kube Master for Kubernetes Control plane traffic.

S.No	Source		Destination		Protocol Used	TCP/UDP	Type of Information Communicated
9	AppViewX Admin network #	Any	ISTIO Ingress Proxy IP (Kube Worker)	30190	HTTPS	TCP	To access the AppViewX management console.
10	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.
11	Master	Any	Kube Master	2379	HTTPS	TCP	Required for etcd server communication in a multi-master setup.
12	Master	Any	Kube Master	2380	HTTPS	TCP	Required for etcd server communication in a multi-master setup.
13	All Nodes	Any	All Nodes*	9100	HTTP	TCP	Required for monitoring the node metrics.
<p>* - indicates all the nodes present in the cluster ie. master nodes, secondary master nodes, and worker nodes.</p> <p># - 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).</p>							

**Note:**

- IPs required - 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.



Note: Ensure that the external endpoints that you want to access from the AppViewX worker nodes are accessible. For example, Microsoft CA. Ensure that the corresponding ports and URLs are opened for communication.

- [Configuring Firewall Ports for External Integrations](#)

Configuring 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

Downloading Linux Packages

The following packages are required to install the application.

S.No	Package	Description	Version
1	nmap	Nmap is a utility for network exploration or security auditing.	nmap-ncat-6.40-19.el7.x86_64

S.No	Package	Description	Version
2	curl	curl is a command line tool for transferring data with URL syntax, supporting FTP, FTPS, HTTP, HTTPS, SCP, SFTP, TFTP, TELNET, DICT, LDAP, LDAPS, FILE, IMAP, SMTP, POP3 and RTSP.	curl-7.29.0-54.el7.x86_64
3	ebrates	Ethernet bridge tables is a firewalling tool to transparently filter network traffic passing a bridge. The filtering possibilities are limited to link layer filtering and some basic filtering on higher network layers.	ebrates-2.0.10-16.el7.x86_64
4	sysstat	The sysstat package contains sar, sadf, mpstat, iostat, pidstat, nfsiostat-sysstat, tapestat, cifsioat and sa tools for Linux.	sysstat-10.1.5-19.el7.x86_64
5	zip	The zip program is a compression and file packaging utility.	zip-3.0-11.el7.x86_64
6	unzip	The unzip utility is used to list, test, or extract files from a zip archive.	unzip-6.0-21.el7.x86_64
7	tcpdump	Tcpdump is a command-line tool for monitoring network traffic. Tcpdump can capture and display the packet headers on a particular network interface or on all interfaces.	tcpdump-4.9.2-4.el7_7.1.x86_64
8	rsync	Rsync uses a reliable algorithm to bring remote and host files into sync very quickly.	rsync-3.1.2-4.el7.x86_64
9	openssl	The OpenSSL toolkit provides support for secure communications between	openssl-1.0.2k-21.el7_9.x86_64

S.No	Package	Description	Version
		machines.	
10	bind-utils	Bind-utils contains a collection of utilities for querying DNS (Domain Name System) name servers to find out information about Internet hosts.	bind-utils-9.11.4-26.P2.el7_9.3.x86_64
11	font-config	Fontconfig is designed to locate fonts within the system and select them according to requirements specified by applications.	fontconfig-2.13.0-4.3.el7.x86_64
12	git	Git is a fast, scalable, distributed revision control system with an unusually rich command set that provides both high-level operations and full access to internals.	git-1.8.3.1-23.el7_8.x86_64
13	git-lfs	Git Large File Storage (LFS) replaces large files such as audio samples, videos, datasets, and graphics with text pointers inside Git, while storing the file contents on a remote server.	git-lfs-2.10.0-1.el7.x86_64
14	rsnapshot	This is a remote backup program that uses rsync to take backup snapshots of filesystems. It uses hard links to save space on the disk.	rsnapshot-1.4.3-1.el7.noarch

Configuring YUM

This section guides users to configure AppViewX (CENTOS 7.x) 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.



Note: For information regarding the best practices on rebooting the operating system after security patching, refer to the [Understanding the Best Practices on Reboot Sequence](#).



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:

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

```
rsyslog x86_64 8.24.0-37.el7_9 updates 621 k
sed x86_64 4.2.2-7.el7 base 231 k
selinux-policy noarch 3.13.1-268.el7 base 497 k
selinux-policy-targeted noarch 3.13.1-268.el7 base 7.0 M
setup noarch 2.8.71-11.el7 base 166 k
shared-mime-info x86_64 1.8-5.el7 base 312 k
sqlite x86_64 3.7.17-8.el7_7.1 base 394 k
sudo x86_64 1.8.23-10.el7 base 842 k
systemd x86_64 219-78.el7 base 5.1 M
systemd-libs x86_64 219-78.el7 base 418 k
systemd-sysv x86_64 219-78.el7 base 96 k
teamd x86_64 1.29-3.el7 base 116 k
tuned noarch 2.11.0-9.el7 base 268 k
tzdata noarch 2020d-2.el7 updates 499 k
util-linux x86_64 2.23.2-65.el7 base 2.0 M
vim-minimal x86_64 2:7.4.629-7.el7 base 443 k
xfsprogs x86_64 4.5.0-22.el7 base 897 k
yum noarch 3.4.3-168.el7.centos base 1.2 M
yum-plugin-fastestmirror noarch 1.1.31-54.el7_8 base 34 k
Installing for dependencies:
bc x86_64 1.06.95-13.el7 base 115 k
postgresql-libs x86_64 9.2.24-4.el7_8 base 234 k

Transaction Summary
=====
Install 2 Packages (+2 Dependent packages)
Upgrade 165 Packages

Total size: 272 M
Total download size: 272 M
Is this ok [y/d/N]:
```

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:

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
```

```
sudo sysctl -p
```

3. Verify the bridging using the command

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

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

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.

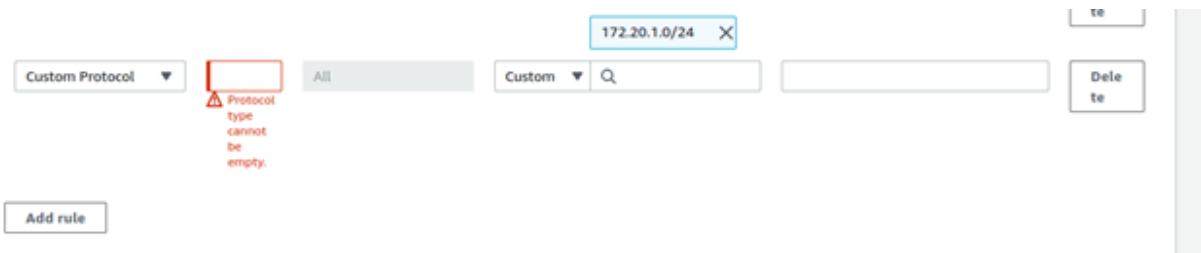
1. Navigate to the `/home/appviewx/appviewx_kubernetes/configs/kube` directory.
2. Open the `calico.yaml` file in edit mode.
3. Change the value of the `CALICO_IPV4POOL_VXLAN` parameter from `CrossSubnet` to `Always`.
4. Change the value of the `CALICO_IPV4POOL_IPIP` parameter from `Always` to `Never`.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

Enabling the IP in IP Protocol

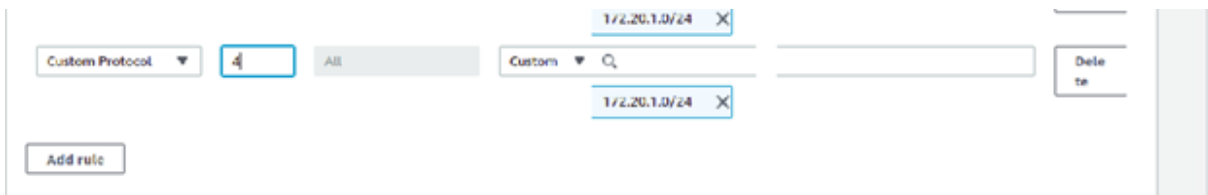
Warning: Follow these steps ONLY if you want to deploy AppViewX on AWS.

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**.

The protocol automatically changes to IP in IP.

The screenshot shows a configuration interface for a network rule. The 'Protocol' dropdown is set to 'IP-In-IP'. The 'Source' and 'Destination' fields both contain '172.20.1.0/24'. There are 'Add rule' and 'Delete' buttons. The interface also shows a search bar and a 'Custom' dropdown menu.

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_2020.3.0.tar.gz	Yes	AppViewX core installer	Core installer that has the AppViewX package from which the installation is triggered.
appviewx_kubernetes_addons_2020.3.0.tar.gz	Yes	To install AppViewX addons	Additional software to support the functionalities of AppViewX. This is mandatory for the installation.
appviewx_kubernetes_elk_2020.3.0.tar.gz	Optional	ELK stack to monitor logs	Additional package to install a GUI-based log collector to troubleshoot and Grafana-based UI to monitor the application performance.
appviewx_kubernetes_insight_2020.3.0.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 managed by

File Name	Mandatory	Description	Purpose
			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 2020.3.0.
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 checks whether all the required prerequisites are present on the system. Sudo permissions are required to execute the tool. This utility can be executed from any of the nodes; either worker or master. The prerequisites are available at https://github.com/AppViewX/prerequisite_utility/.

To run the prerequisite tool:

1. Download and extract the prerequisite_utils.tar.gz file.
2. Copy the updated appviewx.conf file to the location where you have extracted the contents of the prerequisite.tar.gz file.
3. Specify the appviewx IP address of the VMs (master and worker nodes), DNS servers and gateway address, and users in the hosts_template file.
4. Execute the following command: `sudo ./prerequisite`

```

[Thu Mar 25 05:20:01 GMT 2021 ~/abhishek/repo/prerequisite_utility]
[RPK-appviewx@192.168.1.100]$ sudo ./prerequisite
[sudo] password for appviewx:
Verifying archive integrity... 100% MD5 checksums are OK. All good.
Uncompressing prerequisite 100%
Enter password for appviewx@192.168.1.100:

PLAY [db] *****
skipping: no hosts matched

PLAY [worker] *****
[started TASK: Gathering Facts on 192.168.1.100]
[started TASK: worker : debug on 192.168.1.100]
[started TASK: worker : Validating the system architecture on 192.168.1.100]
[started TASK: worker : Validating OS on 192.168.1.100]
[started TASK: worker : Validating OS version on 192.168.1.100]
[started TASK: Validating RAM in worker node on 192.168.1.100]
[started TASK: Validating CPU cores in worker node on 192.168.1.100]
[started TASK: worker : Getting kube path on 192.168.1.100]
[started TASK: worker : Creating kube path on 192.168.1.100]
[started TASK: worker : Checking free space on 192.168.1.100]
[started TASK: worker : Calculating disk space on 192.168.1.100]
[started TASK: worker : Validating disk space on 192.168.1.100]
[started TASK: worker : Getting mongo data size on 192.168.1.100]
[started TASK: Validating disk space in worker node on 192.168.1.100]

PLAY [master] *****
[started TASK: Gathering Facts on 192.168.1.100]
[started TASK: master : Validating the system architecture on 192.168.1.100]
[started TASK: master : Validating OS on 192.168.1.100]

```

```

TASK [Validating disk space in master node] *****
fatal: [192.168.1.100]: FAILED! => ["changed": false, "msg": "Not enough disk space available in master node. At least 30GB is required."]
...ignoring
[started TASK: master : Preparing to check port communication between the servers on 192.168.1.100 ]
[started TASK: master : Verifying ports communication on 192.168.1.100 ]
[started TASK: master : Cleaning up on 192.168.1.100 ]

PLAY [nodes] *****
[started TASK: common : Gather the rpm package facts on 192.168.1.100 ]
[started TASK: common : Copy package.tar.gz on 192.168.1.100 ]
[started TASK: common : Validating rpm dependencies on 192.168.1.100 ]
[started TASK: common : Validating the RPM packages on 192.168.1.100 ]

TASK [common : Validating the RPM packages] *****
fatal: [192.168.1.100]: FAILED! => ["changed": false, "msg": "Error!! Could not install docker RPMs. Please do a yum update and try again. 2. You could see this error if 2020.3.0 or later version is already installed."]
...ignoring
[started TASK: common : Cleaning up RPM packages on 192.168.1.100 ]
[started TASK: common : Validating the RPM packages on 192.168.1.100 ]
[started TASK: common : Preparing to check port communication between the servers on 192.168.1.100 ]
[started TASK: common : Verifying ports communication on 192.168.1.100 ]
[started TASK: common : Cleaning up on 192.168.1.100 ]
[started TASK: common : Getting User ID from 192.168.1.100 on 192.168.1.100 ]
[started TASK: common : Getting user id details on 192.168.1.100 ]
[started TASK: common : Validating user id value on 192.168.1.100 ]
[started TASK: common : Getting umask details on 192.168.1.100 ]
[started TASK: common : Validating umask value on 192.168.1.100 ]
[started TASK: common : Getting then openssl version on 192.168.1.100 ]
[started TASK: common : Validating Openssl version on 192.168.1.100 ]

[started TASK: common : Validating user id value on 192.168.1.100 ]
[started TASK: common : Getting umask details on 192.168.1.100 ]
[started TASK: common : Validating umask value on 192.168.1.100 ]
[started TASK: common : Getting then openssl version on 192.168.1.100 ]
[started TASK: common : Validating Openssl version on 192.168.1.100 ]
[started TASK: common : Getting the time from all nodes on 192.168.1.100 ]
[started TASK: common : Displaying results of time check on 192.168.1.100 ]
[started TASK: common : Creating files for time check on 192.168.1.100 ]
[started TASK: common : Preparing for the time validation on 192.168.1.100 ]
[started TASK: common : Time sync - Processing the data on 192.168.1.100 ]
[started TASK: common : Validating time difference between the servers on 192.168.1.100 ]
[started TASK: common : Getting temp space on 192.168.1.100 ]
[started TASK: common : Validating temp space on 192.168.1.100 ]
[started TASK: common : Validating temp space on 192.168.1.100 ]
[started TASK: common : Collecting ftype info in xfs_info on 192.168.1.100 ]
[started TASK: common : Validating ftype in xfs_info on 192.168.1.100 ]
[started TASK: common : Collecting mount details on 192.168.1.100 ]
[started TASK: common : Validating noexec for /tmp on 192.168.1.100 ]
[started TASK: common : Checking IPV6 is disabled or not on 192.168.1.100 ]
[started TASK: common : Validating IPV6 on 192.168.1.100 ]

PLAY RECAP *****
192.168.1.100 : ok=37  changed=0  unreachable=0  failed=0  skipped=22

```

The command displays the failures in red text and at the end displays a summary of the tasks.

Chapter 3: Deploying the AppViewX Virtual Appliance

- [Deploying the AppViewX Virtual Appliance](#)
- [Download the Release Package](#)
- [Install the AppViewX OVA](#)

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_2020.3.0.tar.gz`
- `appviewx_kubernetes_elk_2020.3.0.tar.gz`
- `appviewx_kubernetes_insight_2020.3.0.tar.gz`
- `appviewx_kubernetes_addons_2020.3.0.tar.gz`
- `upgrade.tar.gz`

To deploy an OVA file,

- [Download the Release Package](#)
- [Install the AppViewX OVA](#)
- [Deploying the AppViewX Virtual Appliance](#)
- [Download the Release Package](#)
- [Install the AppViewX OVA](#)

Download the Release Package

This section covers the procedures for downloading the release package.

To download the release package,

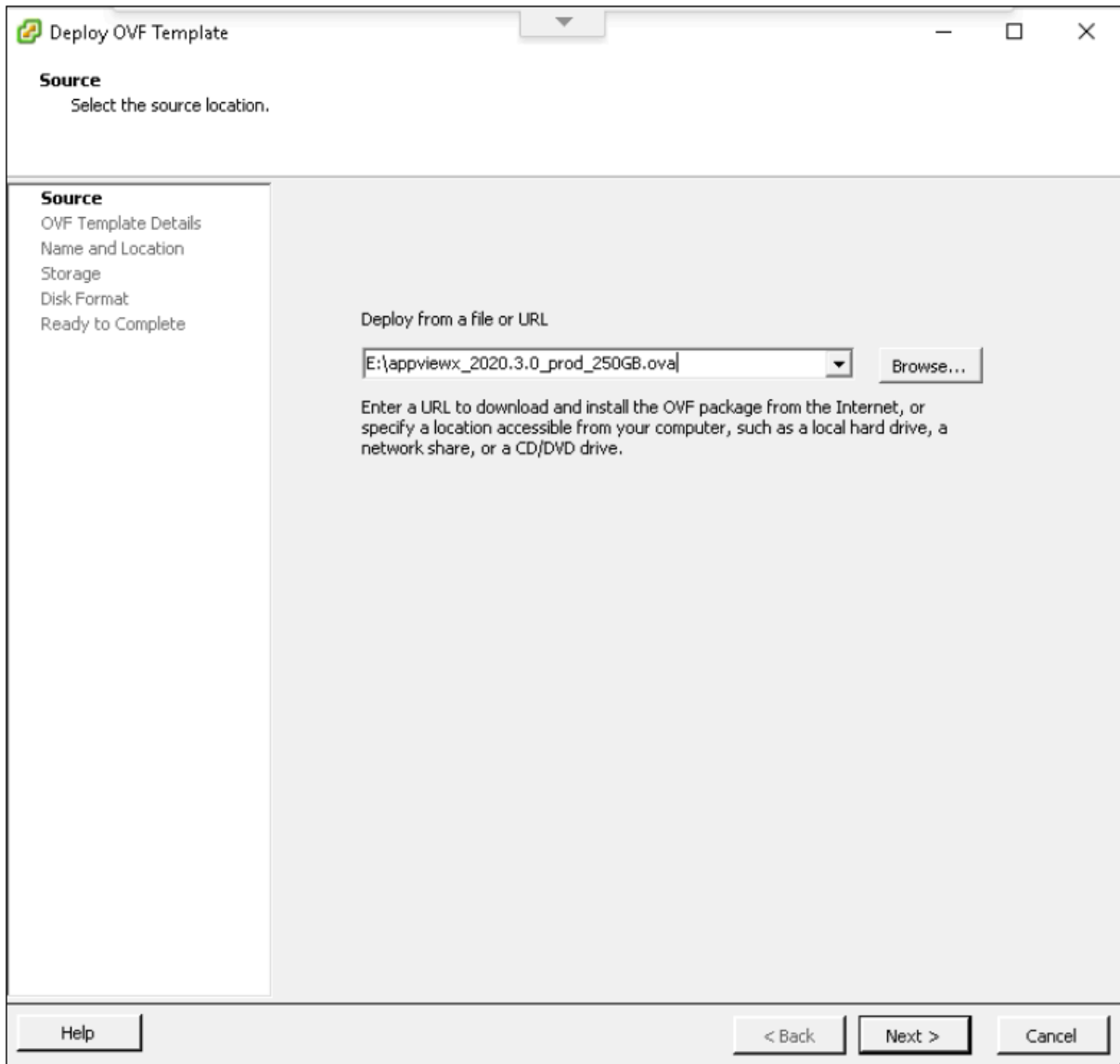
1. Visit the AppViewX download URL at <https://release.appviewx.com>.
2. Download the release package in <.ova> format into the Downloads folder or the Desktop in your environment.
3. Validate the md5sum of the downloaded file
4. Open a terminal window.
5. To display the md5sum value of the downloaded 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.

Install the AppViewX OVA

This section covers the procedures for installing the AppViewX OVA.

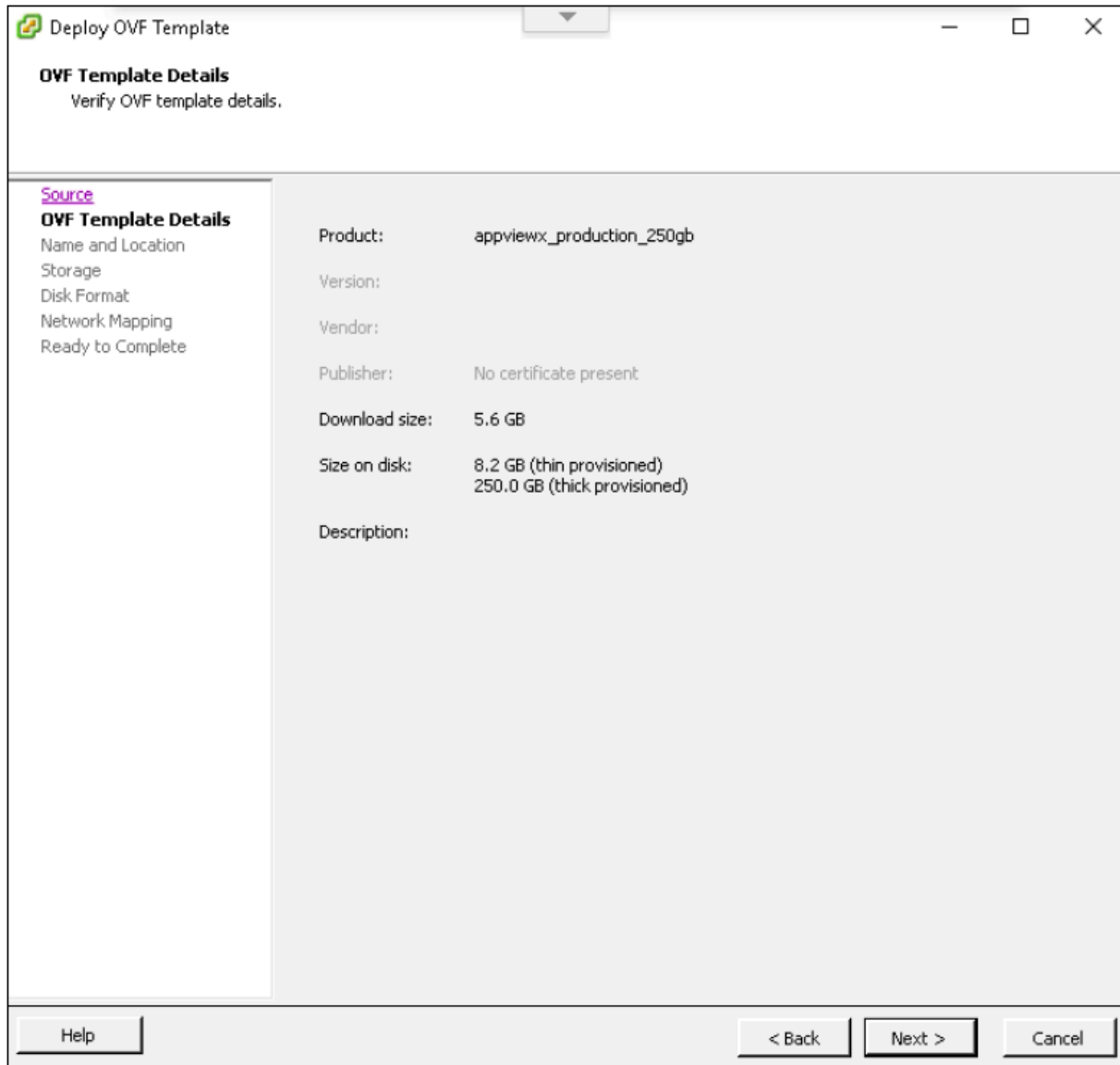
To install the AppViewX OVA,

1. Log in to the vSphere Client.
2. Select **File > Deploy OVF Template**.
The **Name and Location** screen is displayed.



3. Click **Next**.

The **Source** screen of the Deploy OVF Template wizard is displayed.



4. Click **Next**.

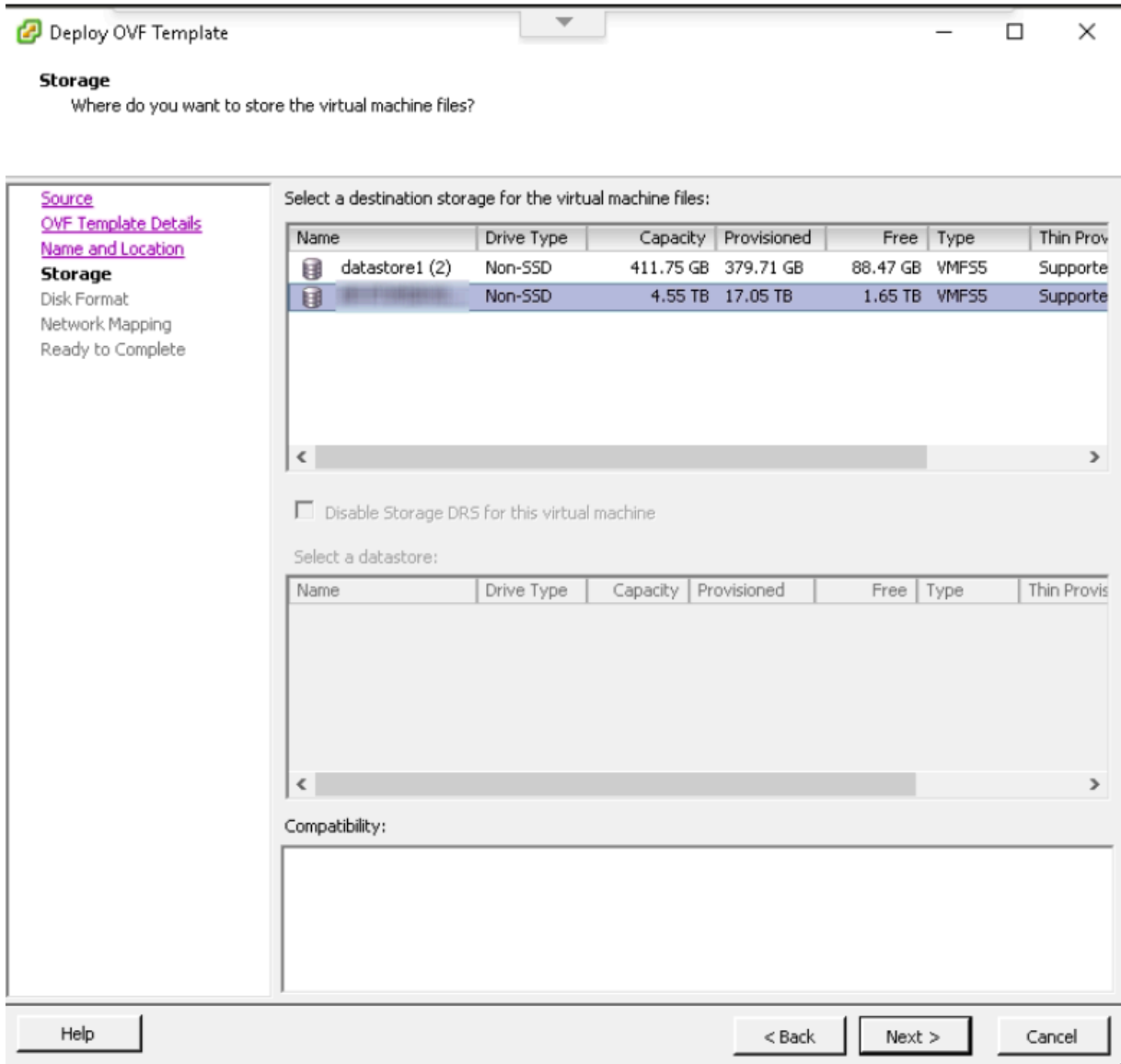
Name and Location screen is displayed.

The screenshot shows a window titled "Deploy OVF Template" with a standard Windows-style title bar (minimize, maximize, close buttons). The main content area is titled "Name and Location" with the instruction "Specify a name and location for the deployed template". On the left side, there is a vertical navigation pane with the following items: "Source" (underlined), "OVF Template Details" (underlined), "Name and Location" (bold), "Storage" (underlined), "Disk Format", "Network Mapping", and "Ready to Complete". The main area on the right has a "Name:" label above a text input field containing "appviewx_production_server". Below the input field, a note states: "The name can contain up to 80 characters and it must be unique within the inventory folder." At the bottom of the window, there are four buttons: "Help", "< Back", "Next >", and "Cancel".

5. (Optional) In the **Name and Location** screen, change the server name to display.

6. Click **Next**.

The **Storage** screen is displayed.



7. In the **Storage** screen, select a destination storage for the VM files.

8. Click **Next**.

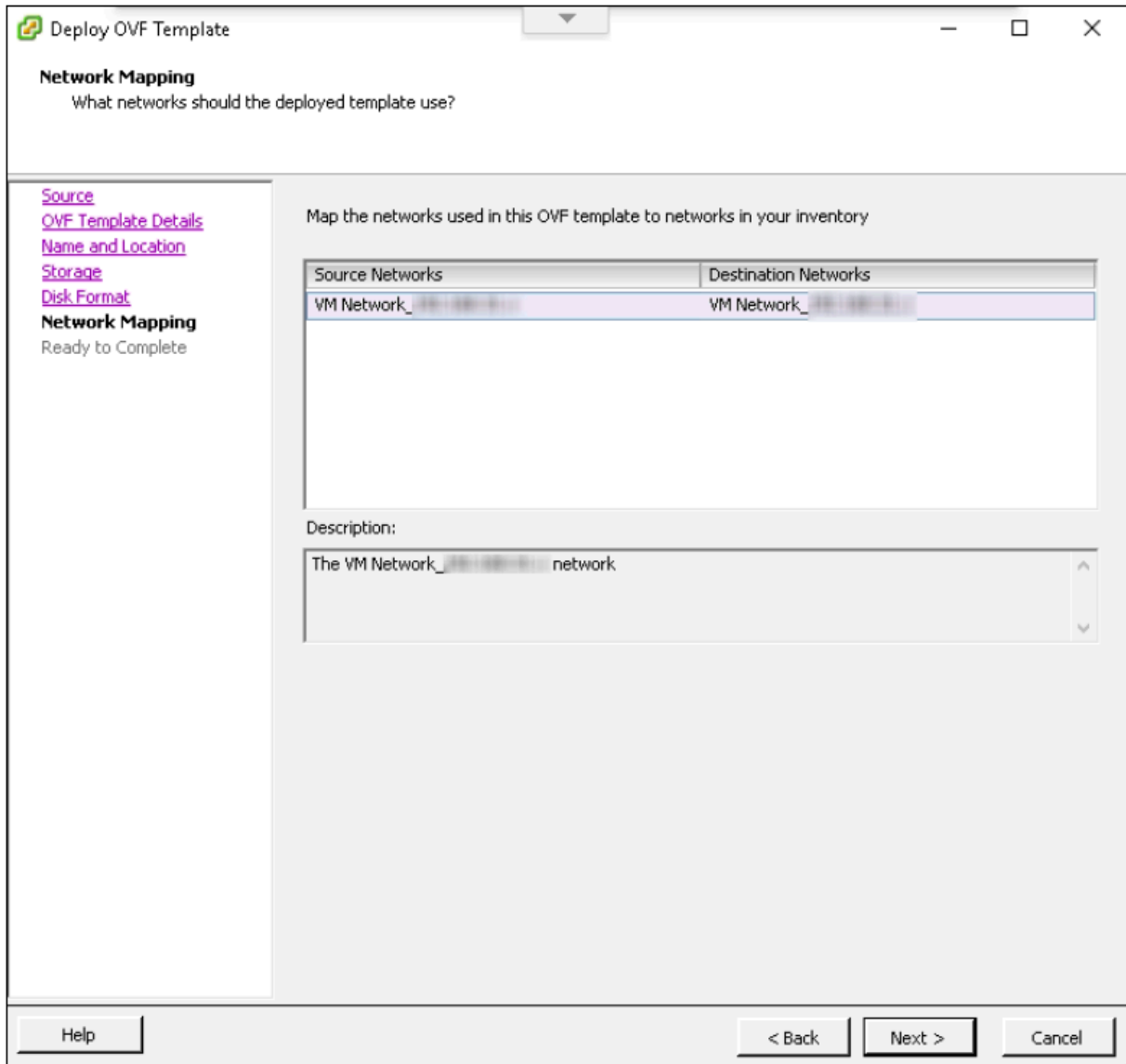
The **Disk Format** screen is displayed.

The screenshot shows a window titled "Deploy OVF Template" with a standard Windows-style title bar (minimize, maximize, close). The main content area is titled "Disk Format" and contains the question "In which format do you want to store the virtual disks?". On the left side, there is a vertical navigation pane with links for "Source", "OVF Template Details", "Name and Location", "Storage", "Disk Format" (which is highlighted), "Network Mapping", and "Ready to Complete". The main area contains the following fields and options:

- Datastore:** A text box containing the value "datastore1".
- Available space (GB):** A text box containing the value "1691.5".
- Provisioning Options:** Three radio button options:
 - Thick Provision Lazy Zeroed
 - Thick Provision Eager Zeroed
 - Thin Provision

At the bottom of the window, there are three buttons: "Help", "< Back", and "Next >", and a "Cancel" button on the far right.

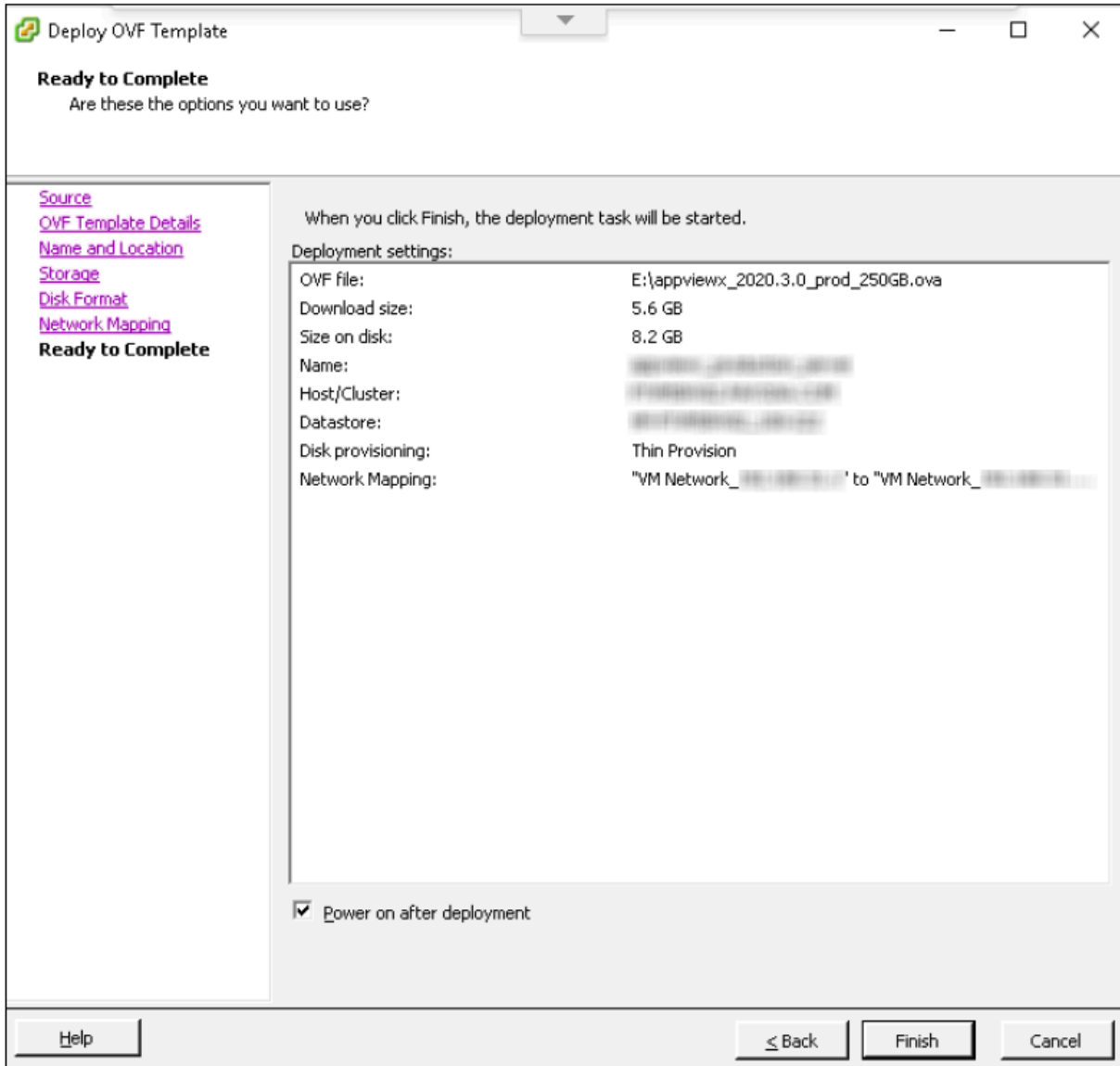
9. In the **Disk Format** screen, select a disk type.
10. Click **Next**.
The **Network Mapping** screen is displayed.



11. In the **Network Mapping** screen, choose a network adapter.

12. Click **Next**.

The **Ready to Complete** screen is displayed.



13. In the **Ready to Complete** screen, verify all the details.
14. Click **Finish**.
15. After the deployment, access the AppViewX VM console.



Note: Contact the account rep for the root credentials as well as the credentials to access AppViewX.

16. To navigate to the root folder as a root user, execute the command: `$ cd /root`

```
[root@pesrv03-regression02-98-13 ~]# cd /root/
[root@pesrv03-regression02-98-13 ~]# pwd
/root
```

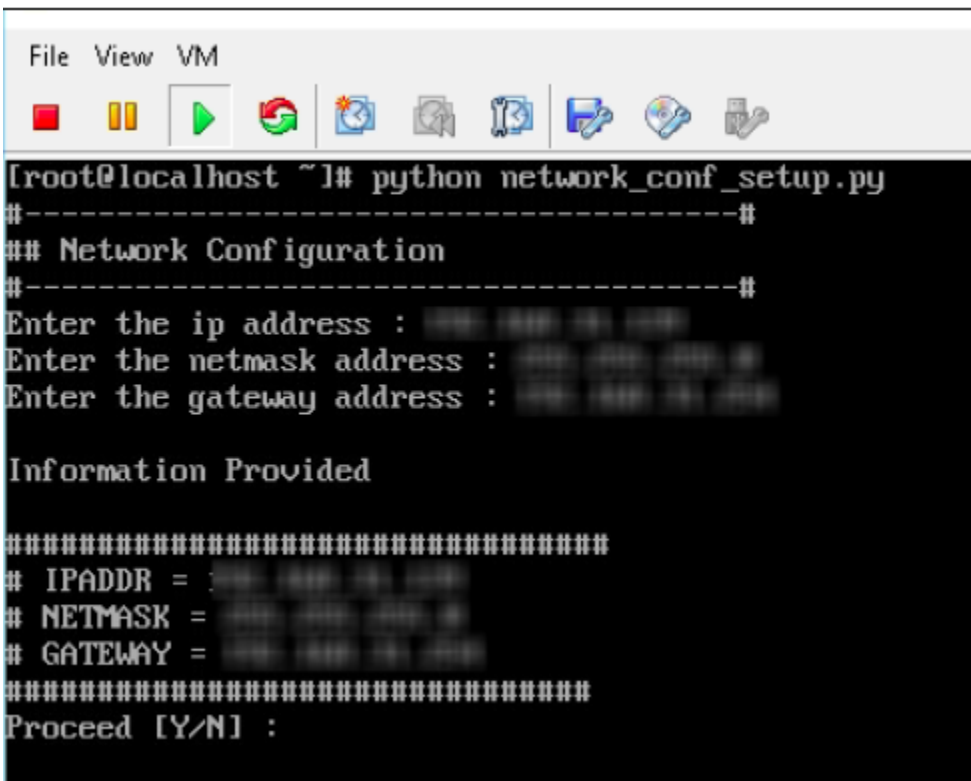
17. Execute the following command: `$ python network_conf_setup.py`

The command starts the console and prompts you to enter the network configuration for the node.

18. Enter the following details at the prompt:

- IP address
- Netmask address
- Gateway address

A prompt to proceed is displayed.



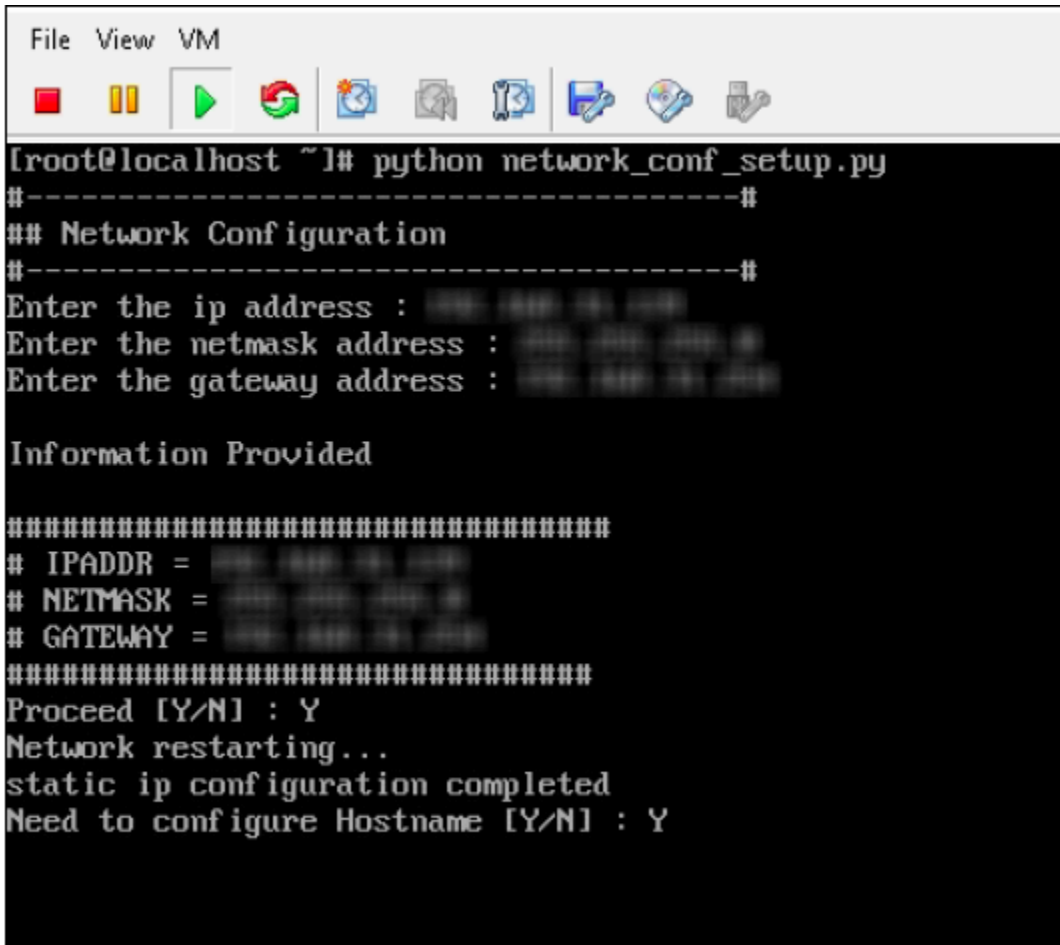
```
File View VM
[root@localhost ~]# python network_conf_setup.py
#-----#
## Network Configuration
#-----#
Enter the ip address : 
Enter the netmask address : 
Enter the gateway address : 

Information Provided

#####
# IPADDR = 
# NETMASK = 
# GATEWAY = 
#####
Proceed [Y/N] :
```

19. Type `Y` to proceed.

20. At the **Need to Configure Hostname [Y/N]** prompt, type `Y` to proceed.



```

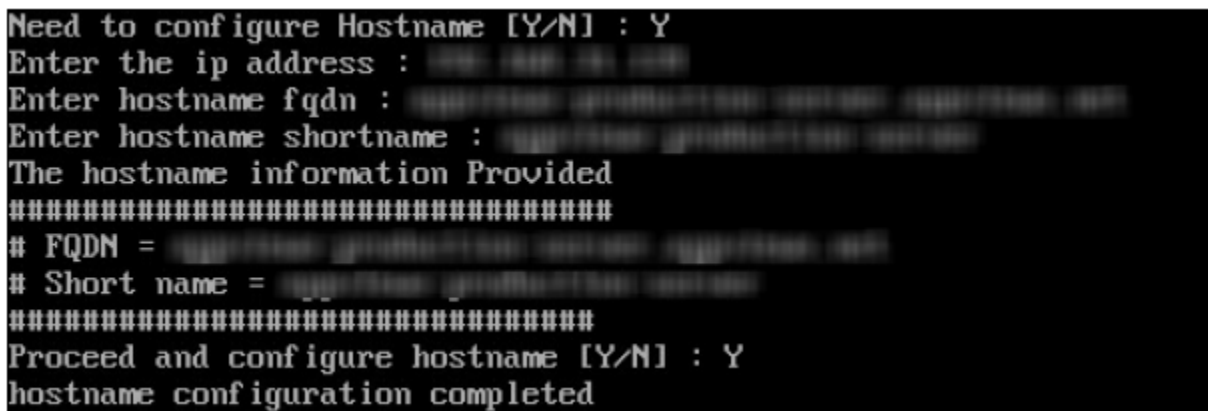
File View VM
[root@localhost ~]# python network_conf_setup.py
#-----#
## Network Configuration
#-----#
Enter the ip address : 192.168.1.100
Enter the netmask address : 255.255.255.0
Enter the gateway address : 192.168.1.1

Information Provided

#####
# IPADDR = 192.168.1.100
# NETMASK = 255.255.255.0
# GATEWAY = 192.168.1.1
#####
Proceed [Y/N] : Y
Network restarting...
static ip configuration completed
Need to configure Hostname [Y/N] : Y

```

21. Enter the **IP address**, desired **hostname**, and a **short name** for the hostname.



```

Need to configure Hostname [Y/N] : Y
Enter the ip address : 192.168.1.100
Enter hostname fqdn : appviewx-192-168-1-100
Enter hostname shortname : appviewx-192-168-1-100
The hostname information Provided
#####
# FQDN = appviewx-192-168-1-100
# Short name = appviewx-192-168-1-100
#####
Proceed and configure hostname [Y/N] : Y
hostname configuration completed

```

22. At the Proceed and Configure the Hostname [Y/N] prompt, type Y to proceed with the node configuration.

```

## Network Configuration
#-----#
Enter the ip address : 192.168.1.100
Enter the netmask address : 255.255.255.0
Enter the gateway address : 192.168.1.100

Information Provided

#####
# IPADDR = 192.168.1.100
# NETMASK = 255.255.255.0
# GATEWAY = 192.168.1.100
#####
Proceed [Y/N] : Y
Network restarting...
static ip configuration completed
Need to configure Hostname [Y/N] : Y
Enter the ip address : 192.168.1.100
Enter hostname fqdn : appviewx-192.168.1.100
Enter hostname shortname : appviewx
The hostname information Provided
#####
# FQDN = appviewx-192.168.1.100
# Short name = appviewx
#####
Proceed and configure hostname [Y/N] : Y
hostname configuration completed
Need to configure /etc/resolv.conf [Y/N] : Y
Enter search domain : appviewx
Enter the no of name servers : 1
Enter the name server 1
192.168.1.100
The dns information provided
#####
# search domain= appviewx
name_servers:
192.168.1.100
#####
Proceed and change resolv.conf [Y/N] : Y
/etc/resolv.conf configuration completed
System configuration completed
Please reboot the system..

```

23. Repeat the steps from 2 to 21 to configure the other nodes.
24. After you complete the OVA and network configuration steps across all nodes, SSH into any one of the nodes as an AppViewX user to start the manual installation process.

Chapter 4: Deploying Appviewx in AWS using Appviewx provided AMI

- [Deploying Appviewx in AWS using Appviewx provided AMI](#)

Deploying Appviewx in AWS using Appviewx provided AMI

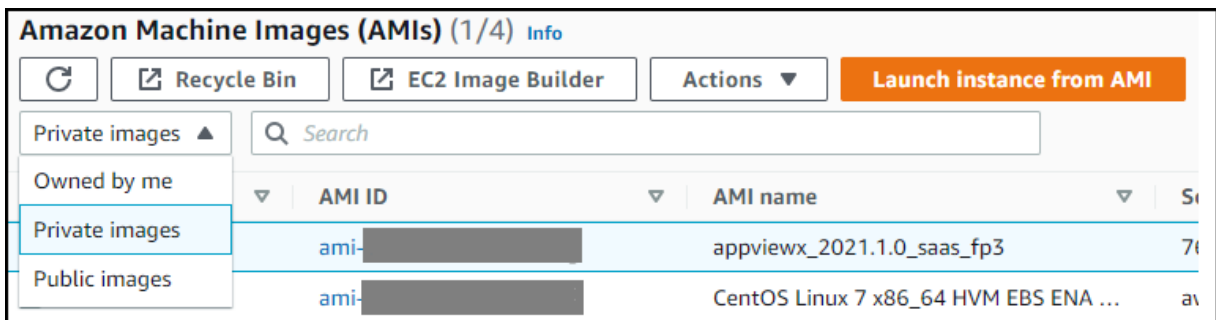
- [Create AWS Instance Using Appviewx AMI](#)
- [Mount Additional Storage for Worker Node](#)

Create AWS Instance Using Appviewx AMI

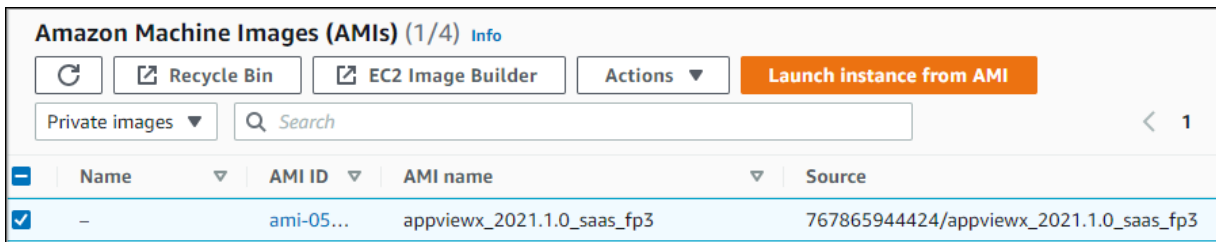
Customers need to approach Appviewx Architect and finalize the deployment model. Then they must share their AWS account number and region so that Appviewx will process the sharing of the AMI to the given AWS account and region.

The steps to creating the AWS instance using Appviewx AMI are as follows:

1. Once the Appviewx AMI is successfully shared to the customer account, login to AWS management console and look for EC2 service > Navigate to Images > AMI > choose “Private images” as shown below.



2. Choose the AMI shared by Appviewx and Launch instances from AMI.



3. Select the hardware specification for master node as below and no need to mount additional storage (Appviewx provided AMI is built with 250GB storage mounted).

Instance type

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

▼ Configure storage [Info](#)

1x GiB **Root volume**

4. Select the hardware specification below for the worker node and increase the default storage from 250GB to 1000GB.

Instance type

t2.2xlarge
 Family: t2 8 vCPU 32 GiB Memory
 On-Demand Linux pricing: 0.3712 USD per Hour
 On-Demand Windows pricing: 0.4332 USD per Hour

Increase the default storage from 250GB to 1000GB as below:

▼ Configure storage [Info](#)

1x GiB **Root volume**

5. Create a new keypair or use an existing keypair to connect to AWS EC2 instance securely.

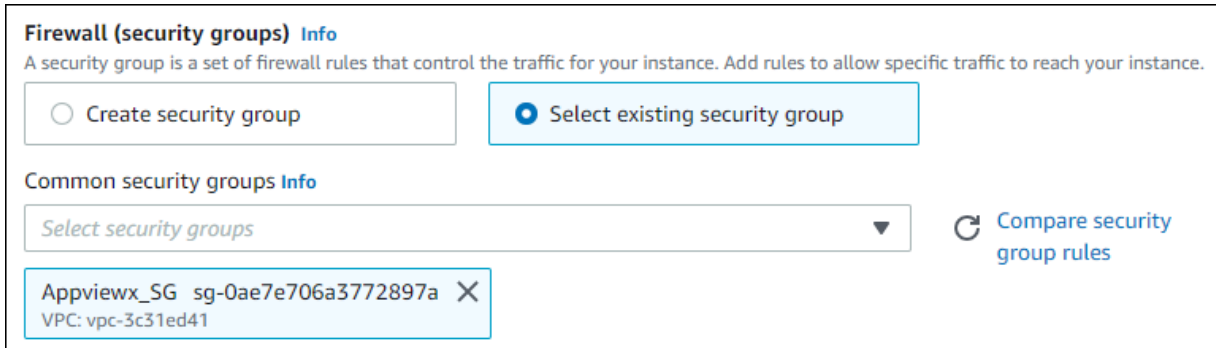
▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

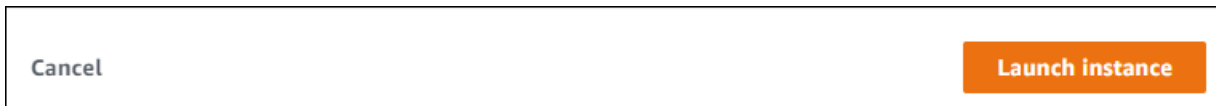
Key pair name - *required*

▼
↻ [Create new key pair](#)

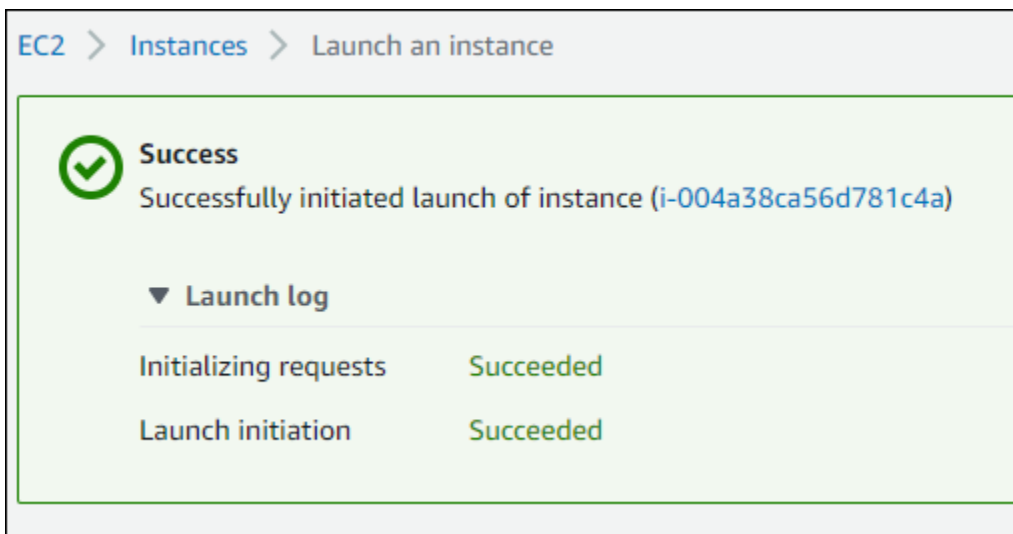
6. Create a new Firewall “Security group” or choose the existing “Security group” and make the required changes.



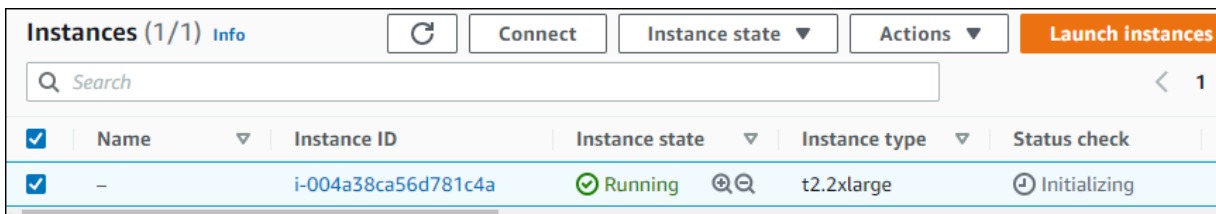
7. Click on **Launch Instance** at the bottom of the screen.



8. Ensure the instance is successfully created.



9. Switch back to **All Instance** page and choose the newly created instance.



10. Now login to AWS instance using the keypair .pem file.

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

- Switch to Appviewx user by executing the below command:

```
sudo su - appviewx
```

- Get the default password for “appviewx” user from Appviewx on-boarding engineer.

Mount Additional Storage for Worker Node

Appviewx AMI comes with a default storage of 250GB. When we increase the storage from 250GB to 1000GB it is required to follow the below instructions to extend the storage mount point from 250GB to 1000GB.



Note: If you are planning for a single-node deployment, please continue to follow the steps. In the case of multi-node deployment, the steps below are to be followed only for worker node/instance.

To mount additional storage for worker node,

- Switch from appviewx user to root by executing the command below and enter appviewx default password.

```
sudo -i
```

- Check the existing available disk space in all mount points.

```
df -h
```

```
[root@ip-172-31-62-178 ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        16G   0    16G   0% /dev
tmpfs           16G   0    16G   0% /dev/shm
tmpfs           16G 584K   16G   1% /run
tmpfs           16G   0    16G   0% /sys/fs/cgroup
/dev/mapper/centos-root 50G  1.7G   49G   4% /
/dev/xvda1      2.0G 232M  1.8G  12% /boot
/dev/mapper/centos-home 100G  5.9G   95G   6% /home
/dev/mapper/centos-tmp  10G   33M   10G   1% /tmp
/dev/mapper/centos-var  20G  533M   20G   3% /var
/dev/mapper/centos-var_log 20G  197M   20G   1% /var/log
/dev/mapper/centos-var_tmp 20G   33M   20G   1% /var/tmp
/dev/mapper/centos-var_log_audit 20G  114M   20G   1% /var/log/audit
tmpfs           3.2G   0    3.2G   0% /run/user/1001
```

- Execute the below command to check the logical/physical size under root mount point.

```
fdisk -l
```

```
[root@ip-172-31-62-178 ~]# fdisk -l

Disk /dev/xvda: 1073.7 GB, 1073741824000 bytes, 2097152000 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x0002820a

   Device Boot      Start         End      Blocks   Id  System
/dev/xvda1    *          2048     4196351     2097152   83  Linux
/dev/xvda2                4196352    507529215    251666432   8e  Linux LVM

Disk /dev/mapper/centos-root: 53.7 GB, 53687091200 bytes, 104857600 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mapper/centos-var: 21.5 GB, 21474836480 bytes, 41943040 sectors
```

- List the mount points.

```
lsblk
```

```
[root@ip-172-31-62-178 ~]# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda                                 202:0    0 1000G  0 disk
├─xvda1                              202:1    0    2G  0 part /boot
└─xvda2                              202:2    0  240G  0 part
   ├─centos-root                    253:0    0   50G  0 lvm  /
   ├─centos-var                    253:1    0   20G  0 lvm  /var
   ├─centos-tmp                    253:2    0   10G  0 lvm  /tmp
   ├─centos-var_log                253:3    0   20G  0 lvm  /var/log
   ├─centos-var_tmp                253:4    0   20G  0 lvm  /var/tmp
   ├─centos-var_log_audit          253:5    0   20G  0 lvm  /var/log/audit
   └─centos-home                    253:6    0  100G  0 lvm  /home
```

- Get into fdisk by executing the below command, type “n” and hit Enter.

```
fdisk /dev/xvda
```

```
[root@ip-172-31-62-178 ~]# fdisk /dev/xvda
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): n
```

- a. Now type “p” for partition type and hit Enter

```
Partition type:
  p   primary (2 primary, 0 extended, 2 free)
  e   extended
Select (default p): p
```

- b. Choose Partition 3 and hit Enter.
c. For the First sector, leave it blank and hit Enter.

```
First sector (507529216-2097151999, default 507529216):
```

- d. For the Last sector, leave it blank and hit Enter.

```
Using default value 507529216
Last sector, +sectors or +size{K,M,G} (507529216-2097151999, default 2097151999):
```

- e. To choose the type of partition, type “t” to execute the below command and hit Enter.

```
Command (m for help): t
```

- f. Choose the partition number as “3”.

```
Partition number (1-3, default 3): 3
```

- g. To list all codes, type “L”.

```
Hex code (type L to list all codes): L
 0 Empty                24 NEC DOS              81 Minix / old Lin    bf Solaris
 1 FAT12                27 Hidden NTFS win    82 Linux swap / So   c1 DRDOS/sec (FAT-
 2 XENIX root           39 Plan 9              83 Linux              c4 DRDOS/sec (FAT-
 3 XENIX usr            3c PartitionMagic     84 OS/2 hidden C:    c6 DRDOS/sec (FAT-
 4 FAT16 <32M          40 Venix 80286        85 Linux extended    c7 Syrix
 5 Extended            41 PPC PREP Boot      86 NTFS volume set   da Non-FS data
 6 FAT16               42 SFS                 87 NTFS volume set   db CP/M / CTOS / .
 7 HPFS/NTFS/exFAT    4d QNX4.x              88 Linux plaintext   de Dell Utility
 8 AIX                 4e QNX4.x 2nd part    8e Linux LVM         df BootIt
 9 AIX bootable       4f QNX4.x 3rd part    93 Amoeba            e1 DOS access
 a OS/2 Boot Manag   50 OnTrack DM         94 Amoeba BBT        e3 DOS R/O
 b w95 FAT32         51 OnTrack DM6 Aux   9f BSD/OS            e4 SpeedStor
 c w95 FAT32 (LBA)  52 CP/M              a0 IBM Thinkpad hi  eb BeOS fs
 e w95 FAT16 (LBA)  53 OnTrack DM6 Aux   a5 FreeBSD          ee GPT
 f w95 Ext'd (LBA)  54 OnTrackDM6        a6 OpenBSD          ef EFI (FAT-12/16/
10 OPUS              55 EZ-Drive          a7 NeXTSTEP         f0 Linux/PA-RISC b
11 Hidden FAT12      56 Golden Bow        a8 Darwin UFS       f1 SpeedStor
12 Compaq diagnost  5c Priam Edisk        a9 NetBSD           f4 SpeedStor
14 Hidden FAT16 <3  61 SpeedStor        ab Darwin boot     f2 DOS secondary
16 Hidden FAT16     63 GNU HURD or Sys   af HFS / HFS+       fb VMware VMFS
17 Hidden HPFS/NTF  64 Novell Netware    b7 BSDI fs          fc VMware VMKCORE
18 AST SmartSleep   65 Novell Netware    b8 BSDI swap         fd Linux raid auto
1b Hidden w95 FAT3  70 DiskSecure Mult   bb Boot Wizard hid  fe LANstep
1c Hidden w95 FAT3  75 PC/IX             be Solaris boot     ff BBT
1e Hidden w95 FAT1  80 Old Minix
```

6. Choose the type of partition “8e” for “Linux LVM”.

```
Hex code (type L to list all codes): 8e
Changed type of partition 'Linux' to 'Linux LVM'
```

- a. To write, type “w”.

```
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
```

- b. Execute partprobe and volume display commands.

```
[root@ip-172-31-49-7 ~]# partprobe
[root@ip-172-31-49-7 ~]# vgsdisplay
--- Volume group ---
VG Name                centos
System ID
Format                 lvm2
Metadata Areas         1
Metadata Sequence No  8
VG Access              read/write
VG Status              resizable
MAX LV                 0
Cur LV                7
Open LV               7
Max PV                 0
Cur PV                1
Act PV                1
VG Size                240.00 GiB
PE Size                4.00 MiB
Total PE              61441
Alloc PE / Size       61440 / 240.00 GiB
Free PE / Size        1 / 4.00 MiB
VG UUID                V79Svw-Qq1g-IZ8J-sw07-TVHj-TWcx-lyWVUx
```

7. Create physical volume by executing the below command.

```
pvcreeate /dev/xvda3
```

```
[root@ip-172-31-49-7 ~]# pvcreate /dev/xvda3
Physical volume "/dev/xvda3" successfully created.
```

8. Extend the volume group by executing the below command.

```
vgextend centos /dev/xvda3
```

```
[root@ip-172-31-49-7 ~]# vgextend centos /dev/xvda3
Volume group "centos" successfully extended
```

9. Check the volume group by executing the command.

vgdisplay

```
[root@ip-172-31-49-7 ~]# vgdisplay
--- Volume group ---
VG Name                centos
System ID
Format                 lvm2
Metadata Areas         2
Metadata Sequence No   9
VG Access               read/write
VG Status               resizable
MAX LV                 0
Cur LV                 7
Open LV                 7
Max PV                 0
Cur PV                 2
Act PV                  2
VG Size                 997.99 GiB
PE Size                 4.00 MiB
Total PE                255486
Alloc PE / Size        61440 / 240.00 GiB
Free PE / Size         194046 / 757.99 GiB
VG UUID                 V79Svw-Qqlg-IZ8J-sw07-TVHj-TWcx-lywVUX
```

10. Check the available disk space of all the Filesystem before extending the volume for centos-home.

```
[root@ip-172-31-49-7 ~]# df -kh
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        16G   0    16G   0% /dev
tmpfs           16G   0    16G   0% /dev/shm
tmpfs           16G 556K   16G   1% /run
tmpfs           16G   0    16G   0% /sys/fs/cgroup
/dev/mapper/centos-root 50G  1.7G   49G   4% /
/dev/xvda1       2.0G 232M  1.8G  12% /boot
/dev/mapper/centos-home 100G  5.9G   95G   6% /home
/dev/mapper/centos-var  20G 533M   20G   3% /var
/dev/mapper/centos-tmp  10G  33M   10G   1% /tmp
/dev/mapper/centos-var_log 20G 181M   20G   1% /var/log
/dev/mapper/centos-var_tmp 20G  33M   20G   1% /var/tmp
/dev/mapper/centos-var_log_audit 20G 114M   20G   1% /var/log/audit
tmpfs           3.2G   0    3.2G   0% /run/user/1001
```

- Extend the Volume for centos-tmp
- Extend the Volume for cetos-home
- Enable password auth and Bypass .pem auth

Extend the Volume for centos-tmp

To extend the volume for centos-tmp,

1. Extend 10GB for `/dev/mapper/centos-tmp` by executing the following commands:

```
lvextend -L +10G /dev/mapper/centos-tmp
```

```
[root@ip-172-31-49-7 ~]# lvextend -L +10G /dev/mapper/centos-tmp
Size of logical volume centos/tmp changed from 10.00 GiB (2560 extents) to 20.00 GiB (5120 extents).
Logical volume centos/tmp successfully resized.
```

```
xfs_growfs /dev/mapper/centos-tmp
```

```
[root@ip-172-31-49-7 ~]# xfs_growfs /dev/mapper/centos-tmp
meta-data=/dev/mapper/centos-tmp isize=512    agcount=4, agsize=655360 blks
          =                               sectsz=512   attr=2, projid32bit=1
          =                               crc=1      finobt=0 spinodes=0
data      =                               bsize=4096 blocks=2621440, imaxpct=25
          =                               sunit=0    swidth=0 blks
naming    =version 2                       bsize=4096 ascii-ci=0 ftype=1
log       =internal                        bsize=4096 blocks=2560, version=2
          =                               sectsz=512 sunit=0 blks, lazy-count=1
realtime  =none                             extsz=4096  blocks=0, rtextents=0
data blocks changed from 2621440 to 5242880
```

2. Check if the volume increases from 10G to 20G.

```
df -h
```

```
[root@ip-172-31-49-7 ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        16G   0    16G   0% /dev
tmpfs           16G   0    16G   0% /dev/shm
tmpfs           16G 588K   16G   1% /run
tmpfs           16G   0    16G   0% /sys/fs/cgroup
/dev/mapper/centos-root 50G  1.7G   49G   4% /
/dev/xvda1      2.0G 232M  1.8G  12% /boot
/dev/mapper/centos-home 840G  5.9G  835G   1% /home
/dev/mapper/centos-var  20G  533M   20G   3% /var
/dev/mapper/centos-tmp  20G   33M   20G   1% /tmp
/dev/mapper/centos-var_log 20G  181M   20G   1% /var/log
/dev/mapper/centos-var_tmp 20G   33M   20G   1% /var/tmp
/dev/mapper/centos-var_log_audit 20G  114M   20G   1% /var/log/audit
tmpfs           3.2G   0    3.2G   0% /run/user/1001
```

3. Now exit from the root user and switch back to the “appviewx” user.



Note: Make sure these steps are executed in all worker node AWS instance created using Appviewx provided AMI.

- Once completed, get support from the on-boarding engineer to start the Appviewx deployment.

Extend the Volume for centos-home

To extend the volume for centos-home,

- Extend the volume for `/dev/mapper/centos-home` by +740GB using the following command:

```
lvextend -L +740G /dev/mapper/centos-home
```

```
[root@ip-172-31-49-7 ~]# lvextend -L +740G /dev/mapper/centos-home
Size of logical volume centos/home changed from 100.00 GiB (25600 extents) to 840.00 GiB (215040 extents).
Logical volume centos/home successfully resized.
You have new mail in /var/spool/mail/root
```

- Execute the next command,

```
xfs_growfs /dev/mapper/centos-home
```

```
[root@ip-172-31-49-7 ~]# xfs_growfs /dev/mapper/centos-home
meta-data=/dev/mapper/centos-home isize=512    agcount=4, agsize=6553600 blks
         =                       sectsz=512    attr=2, projid32bit=1
         =                       crc=1        finobt=0 spinodes=0
data     =                       bsize=4096  blocks=26214400, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096  ascii-ci=0 ftype=1
log      =internal              bsize=4096  blocks=12800, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                  extsz=4096  blocks=0, rtextents=0
data blocks changed from 26214400 to 220200960
```

- Now check the increase in disk volume for `/dev/mapper/centos-home`

```
[root@ip-172-31-49-7 ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        16G   0    16G   0% /dev
tmpfs           16G   0    16G   0% /dev/shm
tmpfs           16G  556K   16G   1% /run
tmpfs           16G   0    16G   0% /sys/fs/cgroup
/dev/mapper/centos-root  50G  1.7G   49G   4% /
/dev/xvda1      2.0G  232M  1.8G  12% /boot
/dev/mapper/centos-home 840G  5.9G  835G   1% /home
/dev/mapper/centos-var  20G  533M   20G   3% /var
/dev/mapper/centos-tmp  10G  33M   10G   1% /tmp
/dev/mapper/centos-var_log  20G  181M   20G   1% /var/log
/dev/mapper/centos-var_tmp  20G  33M   20G   1% /var/tmp
/dev/mapper/centos-var_log_audit  20G  114M   20G   1% /var/log/audit
tmpfs           3.2G   0    3.2G   0% /run/user/1001
```

Enable password auth and Bypass .pem auth

Bypassing the pem auth is required if the Appviewx install script has to successfully execute the installation.

Execute the commands below:

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

After the successful completion of all the above steps, the instances are now ready to run the Appviewx prerequisite tool. This helps to validate that the instances are now ready to start the Appviewx installation.

Chapter 5: Installing AppViewX

- Installing AppViewX
- Performing a Single Node or Standalone Installation
- Performing a Multi-node or High Availability Installation
- Installation Support for 3 Nodes and 2 Datacenters
- Enabling the Load Balancer for the Kube API Server
- Verifying the Installation
- Uploading the License Key
- Adding Third-party Libraries
- Accessing the AppViewX Graphical User Interface
- Installing a Fix Pack

Installing AppViewX

This section covers the process to install AppViewX on Linux servers in a single node as well as a multi-node environment. Once AppViewX is installed, users can verify the installation, upload the license key and integrate third party libraries with AppViewX.



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



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 (\).
- Upgrading from earlier versions is not supported in 2020.3.0. A new install is the only option.

- Installing AppViewX
- Performing a Single Node or Standalone Installation
- Performing a Multi-node or High Availability Installation

- [Installation Support for 3 Nodes and 2 Datacenters](#)
- [Enabling the Load Balancer for the Kube API Server](#)
- [Verifying the Installation](#)
- [Uploading the License Key](#)
- [Adding Third-party Libraries](#)
- [Accessing the AppViewX Graphical User Interface](#)
- [Installing a Fix Pack](#)

Performing a Single Node or Standalone Installation

Prior to performing the installation, ensure the prerequisites success result is received after running the prerequisite tools. For running the prerequisites tool, see section [Running the Prerequisite Tool](#).

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_2020.3.0.tar.gz file, execute the following command: `tar -xvf appviewx_kubernetes_2020.3.0.tar.gz`
5. To move the appviewx_kubernetes_addons_2020.3.0.tar.gz file to the appviewx_kubernetes folder, execute the following command: `mv appviewx_kubernetes_addons_2020.3.0.tar.gz appviewx_kubernetes/`



Note: Refer to the [Configuring POD and Service IP CIDR](#) section before proceeding with the install to change the IP addresses/range used for pods and services.

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

```
[appviewx@pesrv07- ~]$ cp appviewx.conf /home/appviewx/appviewx_kubernetes/scripts/
[appviewx@pesrv07- ~]$
```



Note: If you have received the `appviewx.conf` file already from AppViewX support, you can skip steps 6 through 9. Copy the provided `appviewx.conf` file into `InstallerLocation/appviewx_kubernetes/scripts/` and continue to Step 10.

7. To copy the `appviewx.conf.template` file to the `appviewx.conf` file, execute the following command:
`cp appviewx.conf.template appviewx.conf`



Note: The entire installation process is driven by the values mentioned in the `appviewx.conf` file.

8. To open the `appviewx.conf` file, execute the following command: `vi appviewx.conf`
 9. Enter the configuration values.



Note: For more information, refer to the `Configuring the Cluster` section. Refer to the deployment diagram provided from `help@appviewx.com` or use the reference architecture provided by AppViewX

10. Save the changes to the file and exit the editor.
 11. In the `<InstallerLocation>/appviewx_kubernetes/scripts/` directory, execute the following command `/install.sh`
 12. Enter the user credentials for the respective nodes.

```
[appviewx@appviewx-kube scripts]$ vi appviewx.conf
[appviewx@appviewx-kube scripts]$ ./install.sh
Please enter appviewx password of absecon:appviewx-kube :
```



Note: The `installerLocation` is the path where the installer file is extracted. After you enter the credentials, the installation process starts and takes about 15 to 20 minutes to complete.

After the AppViewX installation is complete, a success message is displayed on the command prompt with the Web and Gateway URLs.



- Note:**
- Users can also find the AppViewX Web and Gateway URLs in the `appviewx.conf` file in the installation location.
 - Users can



- verify the installation by following the instructions provided in the section [Verifying the Installation](#)
- upload the license by referring to the instructions provided in the section
- For troubleshooting issues, please refer to the [Troubleshooting.ditamap](#) section.

Performing a Multi-node or High Availability Installation

This section explains the procedure to install AppViewX in a multi-node environment. The installation procedure is identical to the single node installation with the only difference being the cluster configuration and the POD and Service IP CIDR configuration.

Prior to performing the installation, ensure the prerequisites success result is received after running the prerequisite tools. For running the prerequisites tool, see section [Running the Prerequisite Tool](#).

Recommendations:


- MongoDB is CPU and disk intensive. Therefore, it is recommended to run MongoDB on a worker node.
- The hostnames or IP addresses present in the configuration should be a subset of `SSH_HOSTS`.
- The items in the `SSH` list and the `SSH_HOSTS` list should be in the same order. In other words, if the index of the IP address is 3 in the `SSH` list, it should also be 3 in the `SSH_HOSTS` list.
- It is recommended to assign a data center to a plugin once it is enabled.
- For production environments, a single node deployment is **NOT** recommended because:
 - Single node does not support log monitoring using Kibana and Grafana.
 - Unavailability of HA.
 - Syslog and statistics not available.
- [Configuring the appviewx.conf File to Install AppViewX](#)
- [Configuring POD and Service IP CIDR](#)





Configuring the appviewx.conf File to Install AppViewX


The installation of the application is driven by the `appviewx.conf` file available within the release package. For more information refer to the configuration file available in the following location:




```
<InstallerLocation>/appviewx__kubernetes/scripts/
```





The following parameters must be configured to install the application:





Parameter	Description
MULTINODE	<p>Specifies the boolean value to describe if the installation is in a single node/multi-node environment.</p> <p>Example:</p> <pre>MULTINODE=TRUE (For multi-node)</pre> <pre>MULTINODE=FALSE (For single node)</pre>
SSH	<p>Specifies the comma (,) separated values of node IPs in which the application is set to be deployed.</p> <p>Example:</p> <pre>SSH=192.168.XXX.XXX, 192.168.XXX.XXX, 192.168.XXX.XXX</pre>
SSH_HOST	<p>Specifies the comma (,) separated values of node hostnames in which the application is set to be deployed.</p> <div data-bbox="570 1119 1417 1383" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note: Execute the command <code>hostname</code> in the node and add that output to this field. The hostname of a node must be the output of the command "hostname". Ensure to give the IPs provided in the SSH and host name provided in the SSH_HOST must be in the same order.</p> </div> <p>Example:</p> <pre>SSH_HOST=master:appviewx- kube-95.214.appviewx.net,master:appviewx- kube-95.215.appviewx.net,master:appviewx- kube-95.216.appviewx.net,dc1:appviewx- kube-95.217.appviewx.net,appviewx- kube-95.218.appviewx.net,appviewx-kube-95.219.appviewx.net</pre>

Parameter	Description
	<p> Note: For the master nodes, the recommendation is to have the hostname as master:hostname. Ensure that the SSH_HOST and SSH are in the same order.</p>
INGRESS_HOST	<p>To access AppViewX's Web UI, the INGRESS_HOST parameter must be configured. It can be configured with comma (,) separated values of Kubernetes worker node IP addresses where AppViewX needs to be accessed.</p> <p> Note: For single node AppViewX deployments, ensure that it is the IP address of the instance. To ensure high availability of the multiple DC deployments, it is recommended to add a minimum of one host per DC.</p> <p>Example:</p> <pre>INGRESS_HOST=192.168.XXX.XXX,192.168.XXX.XXX,192.168.XXX.XXX</pre> <p> Note: It is recommended to add the Kubernetes worker node IP addresses in this field.</p> <p> Warning: If the INGRESS_HOST parameter does not contain a host IP address, the AppViewX UI will not be accessible.</p>
ENABLE_IPV6	<p>Specifies whether IP v6 is enabled.</p> <p>Example:</p> <pre>ENABLE_IPV6=False</pre>
INSTALLATION_PATH	<p>Specifies the path in which AppViewX is installed.</p> <p>Example:</p> <pre>INSTALLATION_PATH=/home/appviewx/appviewx/</pre>


Parameter	Description
ENABLED_PLUGINS	<p>Specifies the list of plugins that needs to be enabled in the AppViewX installation.</p> <p>Example:</p> <pre>ENABLED_PLUGINS=appviewx_dependencies, avx_config_server, avx_commons, avx_platform_core, avx_patform_queue, avx_platform_gateway, avx_platform_web, avx_subsystems, avx_vendors</pre>
PLUGINS	<p>Specifies the plugins to be installed in the datacenters.</p> <p>Example:</p> <pre>avx_config_server=absecon:appviewx-kube-150-146.appviewx.net, absecon:appviewx-kube-150-147.appviewx.net avx_platform_core=absecon:appviewx-kube-150-146.appviewx.net, absecon:appviewx-kube-150-147.appviewx.net avx_platform_queue=absecon:appviewx-kube-150-146.appviewx.net, absecon:appviewx-kube-150-147.appviewx.net avx_subsystems=absecon:appviewx-kube-150-146.appviewx.net, absecon:appviewx-kube-150-147.appviewx.net avx_subsystems_sync=absecon:appviewx-kube-150-146.appviewx.net avx_vendors=absecon:appviewx-kube-150-146.appviewx.net, absecon:appviewx-kube-150-147.appviewx.net, absecon:appviewx-kube-150-148.appviewx.net avx_platform_gateway=absecon:appviewx-kube-150-146.appviewx.net, absecon:appviewx-kube-150-147.appviewx.net avx_platform_web=absecon:appviewx-kube-150-146.appviewx.net, absecon:appviewx-kube-150-147.appviewx.net</pre>
SSH_OTHER_USER	<p>Specifies the Linux user account with which AppViewX is installed.</p> <div data-bbox="570 1570 1419 1791" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin: 10px 0;">  Note: AppViewX can be installed only as a Sudo user. Refer to the document Commands executed during AppViewX installation to get the details of commands that the Sudo user needs access to. </div> <p>Example:</p>

Parameter	Description
MONGODB_HOST	<p data-bbox="565 275 867 300">SSH_OTHER_USER=appviewx</p> <p data-bbox="565 359 1409 478">Specifies the comma (,) separated values of node hostnames in which the MongoDB is set to be deployed. This parameter is applicable only in a multi-node installation.</p> <div data-bbox="570 516 1419 688" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px;"> <p data-bbox="581 531 1377 659"> Note: Add the output of hostname command in each node in this field. Do not add the output of hostname -f. A minimum of three nodes must be added.</p> </div> <p data-bbox="565 726 675 751">Example:</p> <div data-bbox="570 779 1419 877" style="border: 1px solid #ccc; border-radius: 5px; padding: 5px; background-color: #f9f9f9;"> <pre data-bbox="581 793 1305 858">MONGODB_HOST=appviewx-kube-95.217.appviewx.net, appviewx-kube-95.218.appviewx.net, appviewx-kube-95.219.appviewx.net</pre> </div> <div data-bbox="570 905 1419 1077" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px;"> <p data-bbox="581 919 1398 1050"> Note: A minimum of three nodes for MongoDB across three data centers are required to achieve HA at the data center level. It is recommended to run MongoDB only in the worker nodes.</p> </div>
ARBITER_HOST	<p data-bbox="565 1161 1398 1457">This parameter is applicable only when AppViewX is deployed with two data centers. Arbiters are MongoDB instances that are part of a replica set but do not hold data. Arbiters participate in elections to break ties. Recommended to enable Arbiters only in AppViewX deployment with two data centers (DC) for high availability. In two DC environments, select the DC that has one Kubernetes master node, configure one of the Kubernetes worker nodes as an Arbiter node.</p> <div data-bbox="570 1495 1419 1625" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px;"> <p data-bbox="581 1509 1377 1596"> Note: If AppViewX deployment is not in two DC environments, this parameter can be blank.</p> </div> <p data-bbox="565 1663 675 1688">Example:</p> <div data-bbox="570 1715 915 1749" style="border: 1px solid #ccc; border-radius: 5px; padding: 5px; background-color: #f9f9f9;"> <pre data-bbox="581 1730 915 1749">ARBITER_HOST=192.168.XXX.XXX</pre> </div>

Parameter	Description
	<p> Note: Do not add multiple IP addresses. Only one IP address is allowed.</p>
VAULT_HOST	<p>This parameter is valid only in multi-node installations. This parameter is comma (,) separated values of node hostnames in which the vault is set to be installed.</p> <p> Note: Add the output of <code>hostname</code> command in each node to this field. A minimum of three nodes must be added.</p> <p>Example:</p> <pre>VAULT_HOST=appviewx-kube-95.217.appviewx.net, appviewx-kube-95.218.appviewx.net, appviewx-kube-95.219.appviewx.net</pre> <p> Note: A minimum of three nodes for a vault across three data centers is required to achieve HA at the data center level. It is recommended to run the vault only in the worker hosts.</p>
MASTER_HOST	<p>Specifies the hostname of the node which you want to run as a Kubernetes Master. The total number of masters can be 1, 3, 5, 7, and so on. For example, for a three-master installation, enter one node in the master host and the other two nodes in the secondary master_host.</p> <p> Note: Add the output of <code><hostname></code> command in this parameter.</p> <p>Example:</p> <pre>MASTER_HOST=appviewx-kube-install-94-179</pre>
SECONDARY_MASTER_HOST	<p>Specifies the list of nodes that are designated to run as secondary masters. The total number of masters can be 1, 3, 5, 7, and so on. For example, for a three-master installation, enter one node in the</p>

Parameter	Description
	<p>master host and the other two nodes in the secondary master_host. This parameter is applicable only in multi-node installations.</p> <div data-bbox="570 380 1419 510" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px;">  Note: For deployments with a single master, comment out the SECONDARY_MASTER_HOST section. </div> <p>Example:</p> <pre>SECONDARY_MASTER_HOST=appviewx-kube-install-94-180, appviewx-kube-install-94-181</pre>
WORKER_HOST	<p>Specifies the hostname of the list of Kubernetes worker nodes.</p> <div data-bbox="570 804 1419 890" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px;">  Note: This parameter can be empty in a three-node setup. </div> <p>Example:</p> <pre>WORKER_HOST=appviewx-kube-install-94-180, appviewx-kube-install-94-181</pre> <div data-bbox="570 1052 1419 1226" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px;">  Note: Do not add the value given in the master_host in the worker_host. The worker and master nodes cannot be the same. This is again applicable only in multi-node installations. </div>
USER_GENERATED_PEM and PRIVATE_KEY_FILE_PATH	<p>Set the value of the USER_GENERATED_PEM variable to TRUE if you want to perform a password-less installation.</p> <div data-bbox="570 1415 1419 1843" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px;">  Note: <ul style="list-style-type: none"> • Update the value of the PRIVATE_KEY_FILE_PATH and set the value of the USER_GENERATED_PEM variable to TRUE. Otherwise, leave it empty. • Ensure that the value of the PRIVATE_KEY_FILE_PATH variable points to the private key file and not the directory. For example: /tmp/user_generated_private.pem. </div>

Parameter	Description
EST_SERVER_ACCESS_CERT	Specifies the location for the digital enrollment certificate.
EST_SERVER_ACCESS_KEY	Specifies the location of the access key for the digital enrollment certificate.
EST_TRUSTED_CA_CERTS	Specifies the location of trusted certificate authorities for the EST server.
avx_crontab	Specifies whether the crontab feature is enabled or not.
MONITORING	<p>Specifies whether monitoring is enabled or not. When you set the value to <code>TRUE</code>, set the value of the <code>PROMETHEUS_HOST</code> and <code>GRAFANA_HOST</code> to one of the worker node for multinodes.</p> <p>Example:</p> <pre>MONITORING=TRUE</pre>
PROMETHEUS_HOST	Specifies the hostname or IP address of the Prometheus node.
GRAFANA_HOST	Specifies the hostname or IP address of the Grafana node.
ELK	<p>Specifies whether the <code>ELK</code> stash is enabled or not. You must specify a value for the <code>ELASTICSEARCH_HOST</code> parameter when you set <code>ELK</code> to <code>TRUE</code> for multinodes.</p> <p>Example:</p> <pre>ELK=FALSE</pre>
ELASTICSEARCH_HOST	Specifies the hostname or IP address of the elastic search host node.
API_ADDRESS	Specifies the hostname of the API server.
INSIGHT	<p>Specifies whether the Insight module is enabled or not.</p> <p>Example:</p> <pre>INSIGHT=TRUE</pre>
SYSLOG	<p>Specifies whether the Syslog module is enabled or not.</p> <p>Example:</p> <pre>SYSLOG=TRUE</pre>

Parameter	Description
<code>INSIGHT_ELASTICSEARCH_HOST</code>	Specifies the hostname or IP address of the insight elastic search host node.
<code>USER_GENERATED_PEM</code> and <code>PRIVATE_KEY_FILE_PATH</code>	<p>Set the value of the <code>USER_GENERATED_PEM</code> variable to <code>TRUE</code> if you want to perform a password-less installation.</p> <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note:</p> <ul style="list-style-type: none"> • Update the value of the <code>PRIVATE_KEY_FILE_PATH</code> and set the value of the <code>USER_GENERATED_PEM</code> variable to <code>TRUE</code>. Otherwise, leave it empty. • Ensure that the value of the <code>PRIVATE_KEY_FILE_PATH</code> variable points to the private key file and not the directory. For example: <code>/tmp/user_generated_private.pem</code>. </div>
Sudoers file configuration	<ul style="list-style-type: none"> • <code>sudo vi /etc/sudoers</code> • Go to the last line and add the line below. <pre>appviewx ALL=(ALL) NOPASSWD: ALL</pre>
<code>ISTIO_SECRET_TT</code>	<p>The istio secret TTL value is used to extend the workload certificates. The secret TTL value should always be in minutes. Execute the utility command to the configuration:</p> <pre>./appviewx.sh --update-secret-ttl</pre> <p>Example:</p> <pre>ISTIO_SECRET_TTL=8640m</pre>

Configuring POD and Service IP CIDR

This section explains how to configure the number of POD/Service IP CIDRs that can run on a node. The Pods that run on a node are allocated an IP address from the node's Pod CIDR range.



Note: It is recommended to use the default settings for the POD and Service IP CIDR.

To configure POD/Service IP CIDRs:

1. Navigate to the `<InstallerLocation>/appviewx_kubernetes/configs/kube/` directory.
2. To open the file, execute the following command: `vi kubeadm-config.yaml.tpl`

```
-bash-4.2$ cd /home/appviewx/appviewx_kubernetes/configs/kube/
-bash-4.2$ vi kubeadm-config.yaml.tpl
-bash-4.2$
```

3. Under the networking section, check for serviceSubnet and change it as per requirements.CIDR
networking: serviceSubnet: <value> <change this default value to the desired CIDR>
4. Under the networking section, check for podSubnet and change it as per requirements.podSubnet:
<value> <change this default value to the desired CIDR>

```
apiVersion: kubeadm.k8s.io/v1beta2
kind: ClusterConfiguration
kubernetesVersion: v1.18.1
controlPlaneEndpoint: "${api_address}:6443"
networking:
  serviceSubnet: "10.20.0.0/24"
  podSubnet: "10.20.0.0/24"
  dnsDomain: "cluster.local"
apiServer:
  certSANS:
  - "${api_address}"
  extraArgs:
    service-account-signing-key-file: /etc/kubernetes/pki/sa.key
    service-account-key-file: /etc/kubernetes/pki/sa.pub
    service-account-issuer: api
    service-account-api-audiences: api,vault,factors
    authorization-mode: "Node,RBAC"
```

5. Save the changes and close the editor.
6. Once the above steps are complete, proceed with the AppViewX installation as mentioned in the [installing_appviewx.ditamap](#) section.



Note:

- After the successful installation, you can access the `.appviewx_configuration` file by following the procedure given in the Accessing the Management Console section.
- Users can upload the license by referring to the instructions provided in the section [Uploading the License Key](#).

Installation Support for 3 Nodes and 2 Datacenters

1. In the `appviewx.conf` file, set the value for the `Multinode` parameter as "TRUE".
2. Update the `SSH` and `SSH_HOST` parameters with the 1 master and min 2 workers as shown below.

```
# Comma separated values of node IPs in which the application is to be deployed
# For single node add this node ip
SSH=192.168.1.1,192.168.1.2,192.168.1.3,192.168.1.4,192.168.1.5,192.168.1.6,192.168.1.7,192.168.1.8,192.168.1.9,192.168.1.10,192.168.1.11,192.168.1.12,192.168.1.13,192.168.1.14,192.168.1.15,192.168.1.16,192.168.1.17,192.168.1.18,192.168.1.19,192.168.1.20,192.168.1.21,192.168.1.22,192.168.1.23,192.168.1.24,192.168.1.25,192.168.1.26,192.168.1.27,192.168.1.28,192.168.1.29,192.168.1.30,192.168.1.31,192.168.1.32,192.168.1.33,192.168.1.34,192.168.1.35,192.168.1.36,192.168.1.37,192.168.1.38,192.168.1.39,192.168.1.40,192.168.1.41,192.168.1.42,192.168.1.43,192.168.1.44,192.168.1.45,192.168.1.46,192.168.1.47,192.168.1.48,192.168.1.49,192.168.1.50,192.168.1.51,192.168.1.52,192.168.1.53,192.168.1.54,192.168.1.55,192.168.1.56,192.168.1.57,192.168.1.58,192.168.1.59,192.168.1.60,192.168.1.61,192.168.1.62,192.168.1.63,192.168.1.64,192.168.1.65,192.168.1.66,192.168.1.67,192.168.1.68,192.168.1.69,192.168.1.70,192.168.1.71,192.168.1.72,192.168.1.73,192.168.1.74,192.168.1.75,192.168.1.76,192.168.1.77,192.168.1.78,192.168.1.79,192.168.1.80,192.168.1.81,192.168.1.82,192.168.1.83,192.168.1.84,192.168.1.85,192.168.1.86,192.168.1.87,192.168.1.88,192.168.1.89,192.168.1.90,192.168.1.91,192.168.1.92,192.168.1.93,192.168.1.94,192.168.1.95,192.168.1.96,192.168.1.97,192.168.1.98,192.168.1.99,192.168.1.100,192.168.1.101,192.168.1.102,192.168.1.103,192.168.1.104,192.168.1.105,192.168.1.106,192.168.1.107,192.168.1.108,192.168.1.109,192.168.1.110,192.168.1.111,192.168.1.112,192.168.1.113,192.168.1.114,192.168.1.115,192.168.1.116,192.168.1.117,192.168.1.118,192.168.1.119,192.168.1.120,192.168.1.121,192.168.1.122,192.168.1.123,192.168.1.124,192.168.1.125,192.168.1.126,192.168.1.127,192.168.1.128,192.168.1.129,192.168.1.130,192.168.1.131,192.168.1.132,192.168.1.133,192.168.1.134,192.168.1.135,192.168.1.136,192.168.1.137,192.168.1.138,192.168.1.139,192.168.1.140,192.168.1.141,192.168.1.142,192.168.1.143,192.168.1.144,192.168.1.145,192.168.1.146,192.168.1.147,192.168.1.148,192.168.1.149,192.168.1.150,192.168.1.151,192.168.1.152,192.168.1.153,192.168.1.154,192.168.1.155,192.168.1.156,192.168.1.157,192.168.1.158,192.168.1.159,192.168.1.160,192.168.1.161,192.168.1.162,192.168.1.163,192.168.1.164,192.168.1.165,192.168.1.166,192.168.1.167,192.168.1.168,192.168.1.169,192.168.1.170,192.168.1.171,192.168.1.172,192.168.1.173,192.168.1.174,192.168.1.175,192.168.1.176,192.168.1.177,192.168.1.178,192.168.1.179,192.168.1.180,192.168.1.181,192.168.1.182,192.168.1.183,192.168.1.184,192.168.1.185,192.168.1.186,192.168.1.187,192.168.1.188,192.168.1.189,192.168.1.190,192.168.1.191,192.168.1.192,192.168.1.193,192.168.1.194,192.168.1.195,192.168.1.196,192.168.1.197,192.168.1.198,192.168.1.199,192.168.1.200,192.168.1.201,192.168.1.202,192.168.1.203,192.168.1.204,192.168.1.205,192.168.1.206,192.168.1.207,192.168.1.208,192.168.1.209,192.168.1.210,192.168.1.211,192.168.1.212,192.168.1.213,192.168.1.214,192.168.1.215,192.168.1.216,192.168.1.217,192.168.1.218,192.168.1.219,192.168.1.220,192.168.1.221,192.168.1.222,192.168.1.223,192.168.1.224,192.168.1.225,192.168.1.226,192.168.1.227,192.168.1.228,192.168.1.229,192.168.1.230,192.168.1.231,192.168.1.232,192.168.1.233,192.168.1.234,192.168.1.235,192.168.1.236,192.168.1.237,192.168.1.238,192.168.1.239,192.168.1.240,192.168.1.241,192.168.1.242,192.168.1.243,192.168.1.244,192.168.1.245,192.168.1.246,192.168.1.247,192.168.1.248,192.168.1.249,192.168.1.250,192.168.1.251,192.168.1.252,192.168.1.253,192.168.1.254,192.168.1.255
SSH_HOST=master:192.168.1.1,192.168.1.2,192.168.1.3,192.168.1.4,192.168.1.5,192.168.1.6,192.168.1.7,192.168.1.8,192.168.1.9,192.168.1.10,192.168.1.11,192.168.1.12,192.168.1.13,192.168.1.14,192.168.1.15,192.168.1.16,192.168.1.17,192.168.1.18,192.168.1.19,192.168.1.20,192.168.1.21,192.168.1.22,192.168.1.23,192.168.1.24,192.168.1.25,192.168.1.26,192.168.1.27,192.168.1.28,192.168.1.29,192.168.1.30,192.168.1.31,192.168.1.32,192.168.1.33,192.168.1.34,192.168.1.35,192.168.1.36,192.168.1.37,192.168.1.38,192.168.1.39,192.168.1.40,192.168.1.41,192.168.1.42,192.168.1.43,192.168.1.44,192.168.1.45,192.168.1.46,192.168.1.47,192.168.1.48,192.168.1.49,192.168.1.50,192.168.1.51,192.168.1.52,192.168.1.53,192.168.1.54,192.168.1.55,192.168.1.56,192.168.1.57,192.168.1.58,192.168.1.59,192.168.1.60,192.168.1.61,192.168.1.62,192.168.1.63,192.168.1.64,192.168.1.65,192.168.1.66,192.168.1.67,192.168.1.68,192.168.1.69,192.168.1.70,192.168.1.71,192.168.1.72,192.168.1.73,192.168.1.74,192.168.1.75,192.168.1.76,192.168.1.77,192.168.1.78,192.168.1.79,192.168.1.80,192.168.1.81,192.168.1.82,192.168.1.83,192.168.1.84,192.168.1.85,192.168.1.86,192.168.1.87,192.168.1.88,192.168.1.89,192.168.1.90,192.168.1.91,192.168.1.92,192.168.1.93,192.168.1.94,192.168.1.95,192.168.1.96,192.168.1.97,192.168.1.98,192.168.1.99,192.168.1.100,192.168.1.101,192.168.1.102,192.168.1.103,192.168.1.104,192.168.1.105,192.168.1.106,192.168.1.107,192.168.1.108,192.168.1.109,192.168.1.110,192.168.1.111,192.168.1.112,192.168.1.113,192.168.1.114,192.168.1.115,192.168.1.116,192.168.1.117,192.168.1.118,192.168.1.119,192.168.1.120,192.168.1.121,192.168.1.122,192.168.1.123,192.168.1.124,192.168.1.125,192.168.1.126,192.168.1.127,192.168.1.128,192.168.1.129,192.168.1.130,192.168.1.131,192.168.1.132,192.168.1.133,192.168.1.134,192.168.1.135,192.168.1.136,192.168.1.137,192.168.1.138,192.168.1.139,192.168.1.140,192.168.1.141,192.168.1.142,192.168.1.143,192.168.1.144,192.168.1.145,192.168.1.146,192.168.1.147,192.168.1.148,192.168.1.149,192.168.1.150,192.168.1.151,192.168.1.152,192.168.1.153,192.168.1.154,192.168.1.155,192.168.1.156,192.168.1.157,192.168.1.158,192.168.1.159,192.168.1.160,192.168.1.161,192.168.1.162,192.168.1.163,192.168.1.164,192.168.1.165,192.168.1.166,192.168.1.167,192.168.1.168,192.168.1.169,192.168.1.170,192.168.1.171,192.168.1.172,192.168.1.173,192.168.1.174,192.168.1.175,192.168.1.176,192.168.1.177,192.168.1.178,192.168.1.179,192.168.1.180,192.168.1.181,192.168.1.182,192.168.1.183,192.168.1.184,192.168.1.185,192.168.1.186,192.168.1.187,192.168.1.188,192.168.1.189,192.168.1.190,192.168.1.191,192.168.1.192,192.168.1.193,192.168.1.194,192.168.1.195,192.168.1.196,192.168.1.197,192.168.1.198,192.168.1.199,192.168.1.200,192.168.1.201,192.168.1.202,192.168.1.203,192.168.1.204,192.168.1.205,192.168.1.206,192.168.1.207,192.168.1.208,192.168.1.209,192.168.1.210,192.168.1.211,192.168.1.212,192.168.1.213,192.168.1.214,192.168.1.215,192.168.1.216,192.168.1.217,192.168.1.218,192.168.1.219,192.168.1.220,192.168.1.221,192.168.1.222,192.168.1.223,192.168.1.224,192.168.1.225,192.168.1.226,192.168.1.227,192.168.1.228,192.168.1.229,192.168.1.230,192.168.1.231,192.168.1.232,192.168.1.233,192.168.1.234,192.168.1.235,192.168.1.236,192.168.1.237,192.168.1.238,192.168.1.239,192.168.1.240,192.168.1.241,192.168.1.242,192.168.1.243,192.168.1.244,192.168.1.245,192.168.1.246,192.168.1.247,192.168.1.248,192.168.1.249,192.168.1.250,192.168.1.251,192.168.1.252,192.168.1.253,192.168.1.254,192.168.1.255
```

3. Set the value of the `MONGODB_MIN_REPLICA` parameter as TRUE.

```
MONGODB_MIN_REPLICA=TRUE
```

4. Ensure that you add a minimum of 2 hosts to the `MONGODB_HOST` parameter. It is mandatory to add any one of the IP addresses of the mongodb host to the `ARBITER_HOST` parameter.

```
MONGODB_HOST=worker1.lab.net,worker2.lab.net
```

```
ARBITER_HOST=192.168.xx.2
```

5. Retain the `VAULT_HOST` parameter as is, because the system will automatically assign a vault host from each of the datacenters.
6. Update the `MASTER_HOST` and the `WORKER_HOST` parameters appropriately with the hostnames.
7. To navigate to the `<installer location>/appviewx_kubernetes/scripts` cd `<installer location>/appviewx_kubernetes/scripts`
8. To run the installation script, execute the following command: `./install.sh`

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 apiserver.



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
  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:

1. To verify whether the load balancer is functioning normally, execute the following command: `curl -k https://loadbalancer-ip:6443/version`

```

-bash-4.2$ curl -k https://192.168.1.100:6443/version
{
  "major": "1",
  "minor": "20",
  "gitVersion": "v1.20.7",
  "gitCommit": "132a687512d7fb058d0f5890f07d4121b3f0a2e2",
  "gitTreeState": "clean",
  "buildDate": "2021-05-12T12:32:49Z",
  "goVersion": "go1.15.12",
  "compiler": "gc",
  "platform": "linux/amd64"
}-bash-4.2$
-bash-4.2$
-bash-4.2$ █

```

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.

```

#API ADDRESS - by default it will be MASTER IP; # If the cluster has a single master, use the IP
#of that master as the api_address. If the cluster has 3 masters, the api_address var needs
#to point to the IP of the load balancer
API_ADDRESS=tpsv-01.appvx.com

```

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

```

appviewx_loadbalancer.tf loadbalancer.sh sshkeyless terraform.tfstate
-bash-4.2$ ./loadbalancer.sh
Please enter appviewx password of master: pesrv02- lab.appviewx.net :
Please enter appviewx password of master: et :
Please enter appviewx password of master: 2-94-206 :
Please enter appviewx password of dc1:gs- :

```

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.

Verifying the Installation

This section provides information on verifying whether the installation of AppViewX is successful. There are a few commands that will help you verify the installation. The commands are listed below.

1. To check the status of the pods, execute the following command: `kubectl get pods -n` If any of the pods show a different status, the application might not function as expected.
2. To restart the pod, execute the following command: `kubectl delete pods -n <dcname> <podname>`

```

[appviewx@appviewx-kube scripts]$ kubectl get pods -n absecon

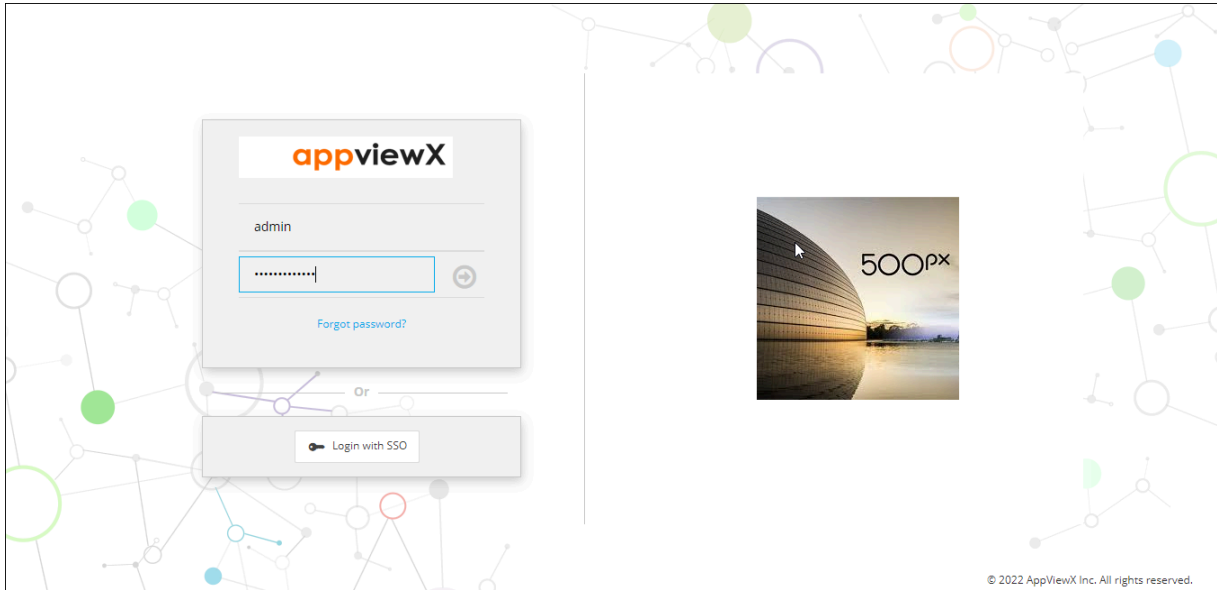
```

NAME	READY	STATUS	RESTARTS	AGE
avx-commons-7f4b5b46fc-7dplc	2/2	Running	2	23m
avx-config-server-6b59c6f67b-rdngm	2/2	Running	0	23m
avx-platform-core-5f4584cfc-5wmj8	2/2	Running	2	23m
avx-platform-queue-7c99dfc48d-txln7	2/2	Running	2	23m
avx-platform-web-d65cf7f47-m6lk8	2/2	Running	0	23m
avx-subsystems-d897946b7-rq84w	2/2	Running	2	23m
avx-subsystems-d897946b7-vp2tn	2/2	Running	2	23m
avx-subsystems-sync-bcf5d674-9d595	2/2	Running	2	23m
avx-vendors-6d574bf496-d4vhc	2/2	Running	1	23m

3. Access the GUI using the AppViewX Web URL with valid credentials. (AppViewX provides the default credentials).



Note: Refer to the `appviewx_configuration` file, available for the URL. The file is available in the `<InstallerLocation>/appviewx__kubernetes/scripts/` directory.



Note: Multi-node installations come with a Redis cluster out-of-the-box. For single-node installations, there is a single Redis instance available that is enabled for PubSub only.



Note: For troubleshooting issues, refer to the [Troubleshooting](#) section.

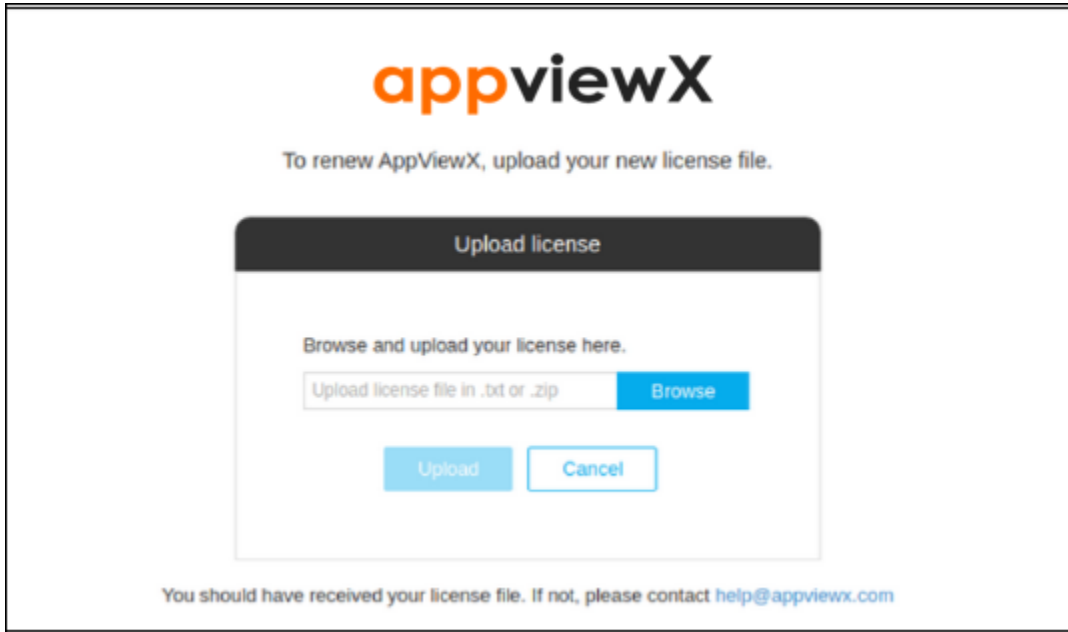
Uploading the License Key

License Management Software tracks software installed throughout the enterprise and ensures legal licensure for its usage. The software helps you to obtain the license key, upload the license key, and troubleshoot the license issues. License management is an essential element of software asset management (SAM).

To access the application, the user needs to upload a license. If you do not have a license key, send an email to help@appviewx.com with the hostname of the node in which the application is installed.

To upload the license key:

1. Log in to the application using the Web URL displayed in the success message of the installation. You are prompted to enter the username and password of the AppViewX admin account.
2. Click **Browse** and upload the license file.



A confirmation message is displayed after uploading a valid license.

Troubleshooting:

- If the license upload fails, ensure that the uploaded file is in the proper <.txt> (or) <.zip> format.
- If the license upload fails while activating the license, ensure that the output of the hostname command is provided during the generation of the license.
- If the license upload fails, trigger the following URL from your browser and try again after a few minutes.

`https://AppViewX GATEWAY URL/refresh`

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.

- 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: <https://devcentral.f5.com/s/articles/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. Copy the `iControl.jar` file from the extracted package to the external_libs directory.
6. If AppViewX is already installed or upgraded from an earlier version of AppViewX, move the `iControl-13.1.0.jar` file to `cp -r /lib/iControl-13.1.0.jar <user_home_dir>/appviewx_dependencies/external_libs/` directory.
7. If AppViewX is not installed, move the `iControl-13.1.0.jar` file to `cp -r /lib/iControl-13.1.0.jar /home/appviewx/Installer/external lib` directory.

8. In case of a multi node environment, copy the `iControl-13.1.0.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.

- To navigate to the directory where Thales client is installed, execute the following command: `cd /opt/nfast/java/classes`
- 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 lib`
- 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:

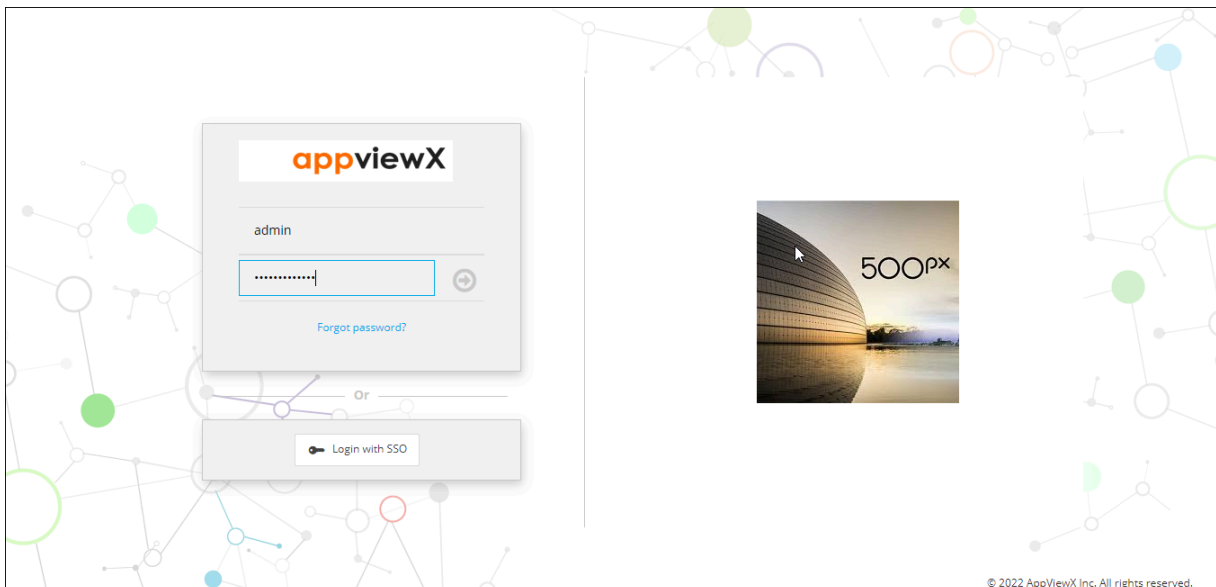

```
cp jcprov.jar /home/appviewx/Installer/external_lib
```
3. Restart the `avx_vendors` plugin followed by the gateway plugin.



Note: For more information on how to restart the plugin, refer to the [Restarting a plugin](#) section.

Accessing the AppViewX Graphical User Interface

1. Access the graphical user interface (GUI) using the AppViewX Web URL with valid credentials.

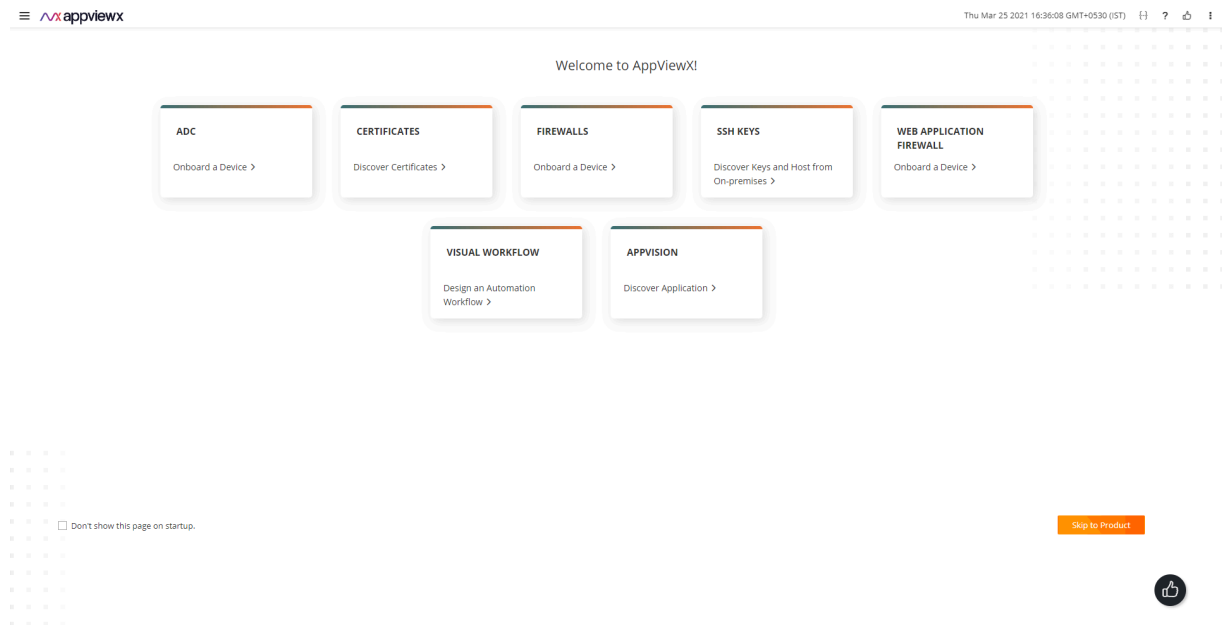


Note: AppViewX provides default credentials to access the GUI.



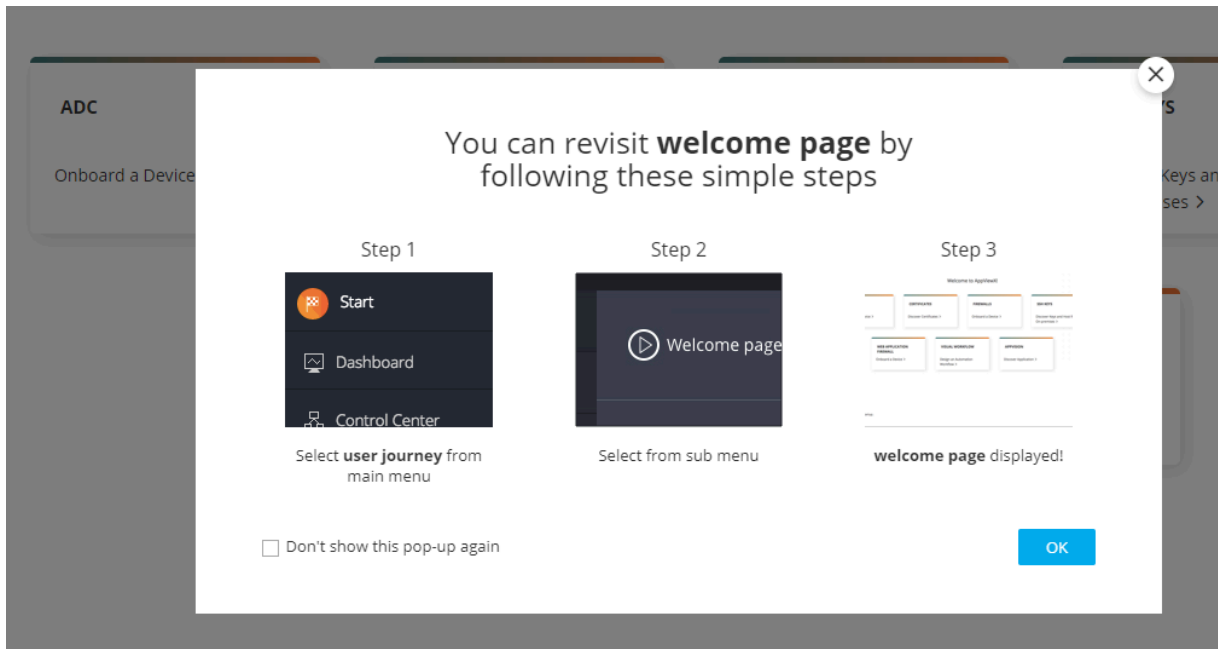
Note: Refer to the appviewx_configuration file, available for the URL. The file is available in the <InstallerLocation>/appviewx__kubernetes/scripts/ directory

Upon successful login, the **Welcome to AppViewX** page is displayed.



2. Click **Skip to Product**.

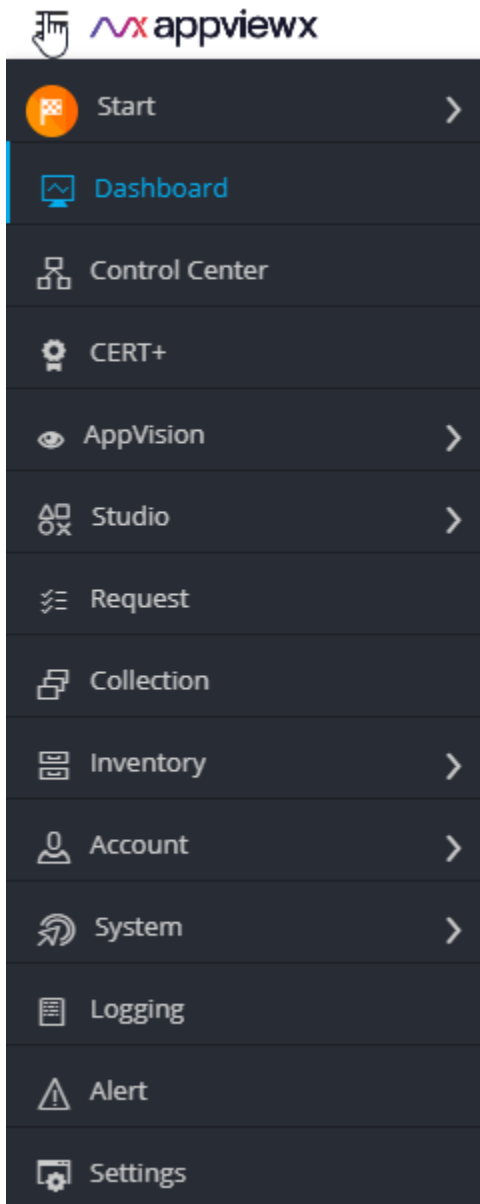
The **Revisit Welcome** page is displayed.



3. Click **OK**.

The system loads the dashboard page.

4. To access the different modules, click the icon to view the menu.



Using this menu, you can navigate to the different modules and access the features of the application.

Installing a Fix Pack

This section provides instructions for applying patches on AppViewX v2020.3.0.

Downloading the Patch

Before installing the fix pack, ensure that you have downloaded the patch plugins and addons from the release portal.

Installing the Patch

The process of installing the fix pack is executed by a script.



Note: For more information and detailed steps, please refer to the AppViewX Patch Deployment Guides.

Chapter 6: Monitoring and Maintaining AppViewX

- Monitoring and Maintaining AppViewX
- Installing ELK Components
- Executing Commands for Maintenance
- Installing Trusted Certificate for GUI/API Access
- Enabling Strict Data Center Routing
- Enabling Device Syslog Processing
- Enabling the Insight Module
- Understanding Commands Executed during Installation
- Enabling Sudo Access
- 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
- Adding a Node to the Cluster
- Understanding the Best Practices on Reboot Sequence
- Working with Alerts
- Working with Backup and Restore
- Working with Logs
- Offline Patching for CentOS
- Working with Plugins
- Working with the Management Console
- Deleting a Node from the Cluster

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.

- [Monitoring and Maintaining AppViewX](#)
- [Installing ELK Components](#)
- [Executing Commands for Maintenance](#)
- [Installing Trusted Certificate for GUI/API Access](#)
- [Enabling Strict Data Center Routing](#)
- [Enabling Device Syslog Processing](#)
- [Enabling the Insight Module](#)
- [Understanding Commands Executed during Installation](#)
- [Enabling Sudo Access](#)
- [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](#)
- [Adding a Node to the Cluster](#)
- [Understanding the Best Practices on Reboot Sequence](#)
- [Working with Alerts](#)
- [Working with Backup and Restore](#)
- [Working with Logs](#)
- [Offline Patching for CentOS](#)
- [Working with Plugins](#)
- [Working with the Management Console](#)
- [Deleting a Node from the Cluster](#)

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.

- View all the nodes in the cluster `kubectl get nodes`

```
[RPK-appviewx@... ]$ kubectl get nodes
NAME                                STATUS    ROLES    AGE     VERSION
pesrv07-devops-94-107             Ready    master   18d    v1.18.6
```

- View all the pods of AppViewX services `kubectl get pods -n avx -o wide`

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

- 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   10.100.100.100 <none>         5300/TCP         18d
avx-platform-web                   ClusterIP   10.100.100.100 <none>         5004/TCP,5555/TCP 18d
mongo-configdb-service              ClusterIP   10.100.100.100 <none>         27017/TCP        18d
mongo-routerdb-service              ClusterIP   10.100.100.100 <none>         27017/TCP        18d
mongo-shareddb-service              ClusterIP   10.100.100.100 <none>         27017/TCP        18d
vault                               ClusterIP   10.100.100.100 <none>         8200/TCP,8201/TCP 18d
vault-internal                      ClusterIP   10.100.100.100 <none>         8200/TCP,8201/TCP 18d
```

- 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#
```

- List all the namespaces `kubectl get namespaces`

```
[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

- 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

- 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

- 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
```

- 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
```

- Edit a configuration `kubectl edit configmaps -n avx avx-common-config`

```

1 # Please edit the object below. Lines beginning with a '#' will be ignored,
2 # and an empty file will abort the edit. If an error occurs while saving this file will be
3 # reopened with the relevant failures.
4 #
5 apiVersion: v1
6 data:
7   APS_MONGO_ENCRYPTED_PASSWORD: wnfzZa0MAvf0R/ULgeCMNA==
8   DATA_CENTER: avx
9   DEPENDENCY_PATH: /appviewx/dependencies
10  MONGO_ENCRYPTED_PASSWORD: vault:v1:JY40+YyoCLfcfUys7T84zWGAB/Vr9sNSk/8h9VYFNIA+jazThgggPeH49ZM5
11  MONGO_KEY: t6ehrofmwa59g3hoakjh4d79s
12  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\n

```

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

```

[RPK-appviewx@10.10.10.10] $ kubectl describe pods -n avx avx-platform-web-5c4595b87-cd2ml
Name:          avx-platform-web-5c4595b87-cd2ml
Namespace:    avx
Priority:      0
Node:         ip-10-10-10-10.us-east-1-b-1.amazonaws.com
Start Time:   Thu, 25 Feb 2021 04:53:59 +0000
Labels:       appviewx/platform-web
              pod-template-hash=5c4595b87
              security.kubernetes.io/certificate-protection=enabled
              service.kubernetes.io/canonical-name=avx-platform-web
              service.kubernetes.io/canonical-name-label=platform-web
Annotations:  com.gigamonkey.com/org.qualys: 10.10.10.10
              com.gigamonkey.com/org.qualys: 10.10.10.10

```

- Log in to the database `kubectl exec -it mongo-routerdb-0 -n avx -- /bin/sh`

```

[RPK-appviewx@10.10.10.10] $ 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

To install a trusted certificate for GUI/API access:

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

```

kubernetes --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

```
appviewx@appviewx-kube ~]$ kubectl --kubeconfig=/tmp/kube_cluster.conf create -n istio-system secret tls external-tls-credential
--key=/home/appviewx/STAR_appviewx_con-2020-comodo/appviewx.con.key --cert=/home/appviewx/STAR_appviewx_con-2020-comodo/STAR_appviewx_con.crt
secret/external-tls-credential created
```

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
```

```
[appviewx@appviewx-kube-install ~]$ sed -i 's/tls-credential/external-tls-credential/g' /home/appviewx/appviewx_kubernetes/yaml/appviewx
_plugins/avx_platform_web/chart/values.yaml
[appviewx@appviewx-kube-install ~]$
```

3. Update the Gateway to consume the latest changes:

- 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
```

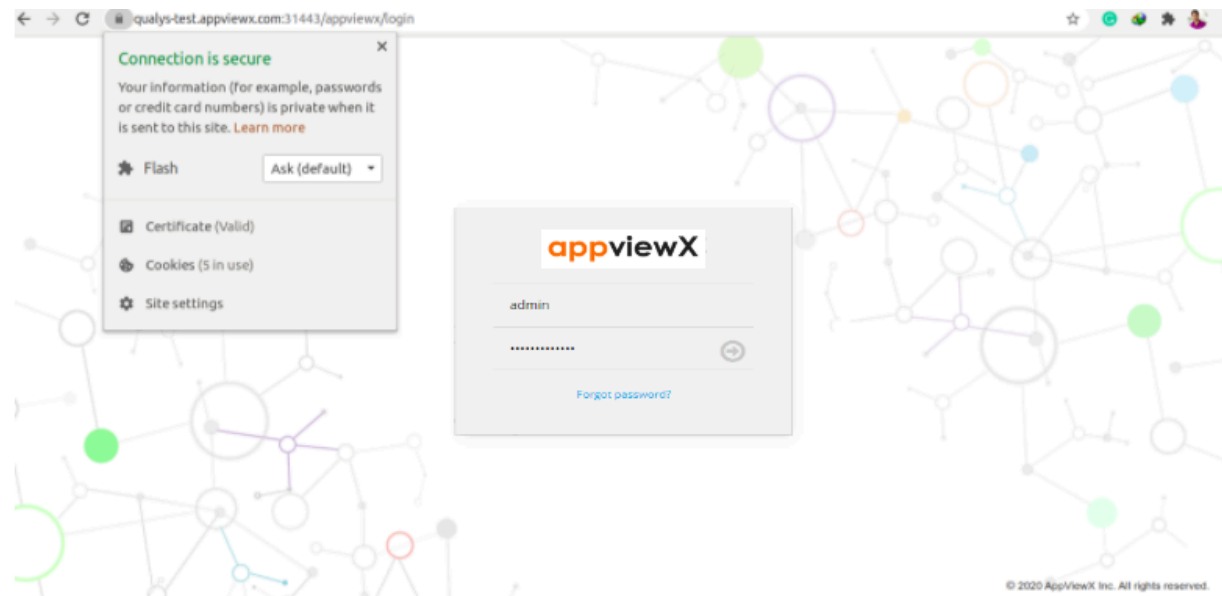
- To upgrade the `avx-platform-web` package to reflect changes, execute the following command:

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

```
[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 ~]$
```

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

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

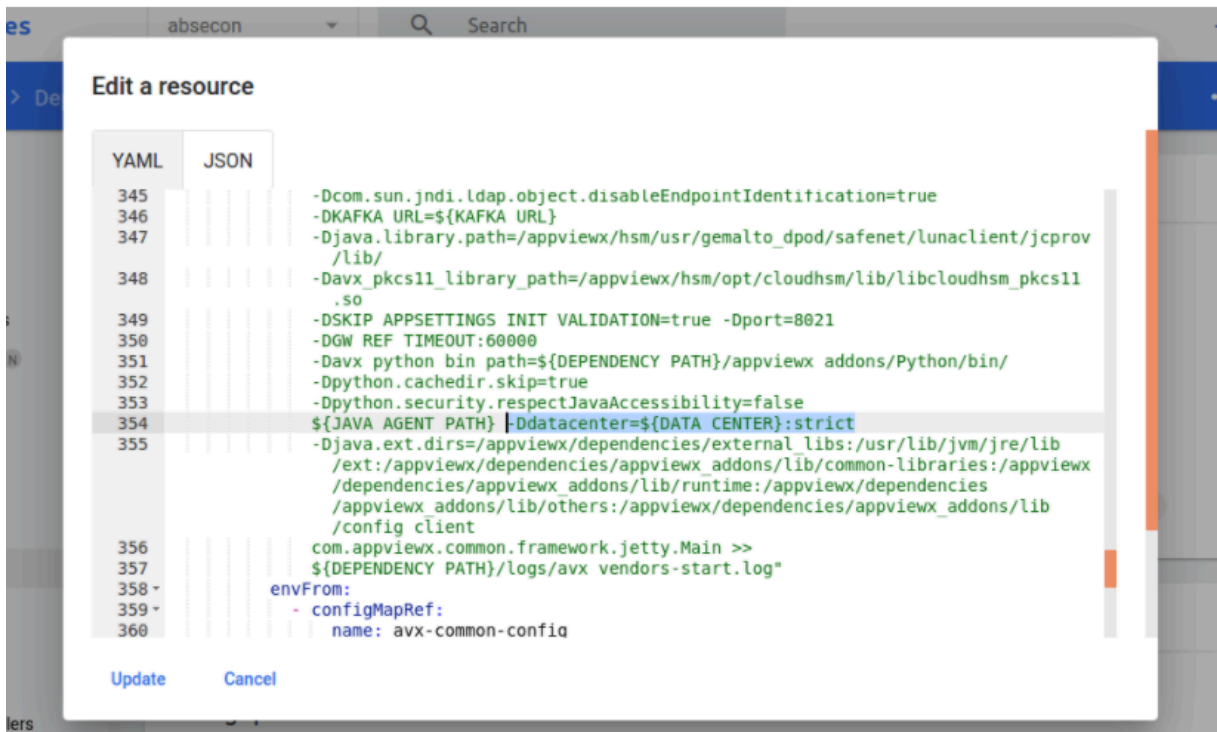


Enabling Strict Data Center Routing

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:



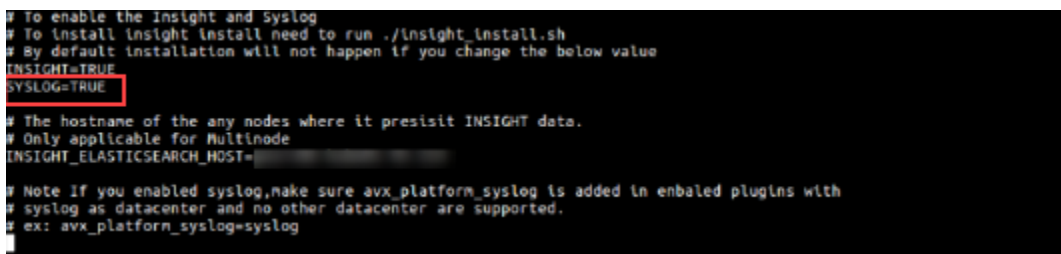
6. Click **Update**.

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.

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 **avx_platform_syslog** plugin. (avx_plaform_syslog=syslog).

```
SSH_OTHER_USER=appviewx
avx_commons=dc1
avx_config_server=dc1
avx_platform_core=dc1
avx_platform_queue=dc1
avx_subsystems=dc1
avx_subsystems_sync=dc1
avx_vendors=dc1
avx_platform_gateway=dc1
avx_platform_web=dc1
avx_insight_subsystem_adc=dc1
avx_insight_statistics_bot=dc1
avx_platform_syslog=syslog
```

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
```

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.

```
[appviewx] appviewx]$ kubectl get services -n syslog
NAME                                TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
avx-platform-syslog                 ClusterIP    10.10.10.10      <none>           5204/tcp         10d
logstash-syslog-service              NodePort    10.10.10.10      <none>           5514:30336/UDP   10d
[appviewx] appviewx]$
```

This Syslog port changes for every installation/upgrade.

12. 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)
```

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

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

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

```
kubectl delete pods "Pod name" -n "data center name"
```

For example:

```
kubectl delete pods avx-subsystems-7666cfb459-6q4rn avx-vendors-99c69cd69-jtr4w -n "data center name"
```

You can restart multiple Pods by entering the name of the Pod in the above command with space.

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_2020.3.0.tar.gz` file.
4. To extract the file, execute the following command: `tar -xvf appviewx_kubernetes_insight_2020.3.0.tar.gz`
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` The value of the `INSIGHT` parameter is set to `TRUE` as shown in the image below.

```
# To enable the insight and Syslog
# To install insight install need to run ./insight_install.sh
# By default installation will not happen if you change the below value
INSIGHT=TRUE
SYSLOG=TRUE

# The hostname of the any nodes where it presisit INSIGHT data.
# Only applicable for Multinode
INSIGHT_ELASTICSEARCH_HOST=

# Note If you enabled syslog,make sure avx_platform_syslog is added in enbaled plugins with
# syslog as datacenter and no other datacenter are supported.
# ex: avx_platform_syslog=syslog
```

7. Search for `Enabled Plugins` and add the following plugins:

- `appviewx_dependencies`
- `avx_insight_subsystem_adc`
- `avx_insight_statistics_bot`
- `avx_platform_gateway`

8. Update the data center for insight plugins as shown in the image below:

```
SSH_OTHER_USER=appviewx
avx_commons=dc1
avx_config_server=dc1
avx_platform_core=dc1
avx_platform_queue=dc1
avx_subsystems=dc1
avx_subsystems_sync=dc1
avx_vendors=dc1
avx_platform_gateway=dc1
avx_platform_web=dc1
avx_insight_subsystem_adc=dc1
avx_insight_statistics_bot=dc1
```

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

```
./insight_install.sh
```

12. Execute the following command:

```
./plugins_install.sh
```

13. Execute the following command:

```
kubectl edit configmaps -n absecon
```

14. Search for the `ELASTIC_ENABLE` parameter and set the value to `TRUE`.
15. Ensure that the following parameters have the default values as given 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`

16. Save and exit the `configmaps` file.
17. To restart subsystems and vendors, execute the following command:

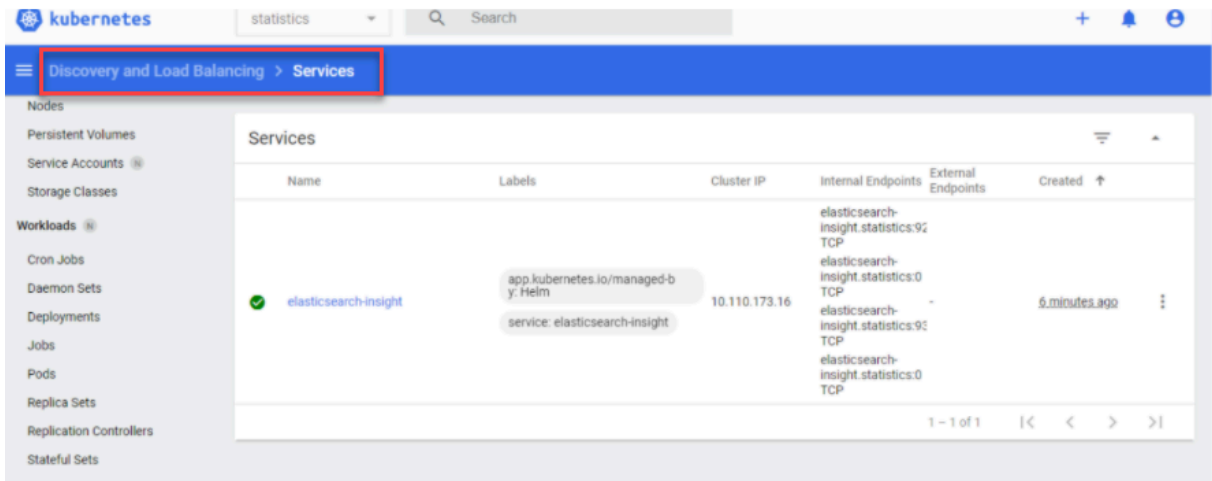
```
kubectl delete pods "Pod name" -n "datacenter name"
```

For example:

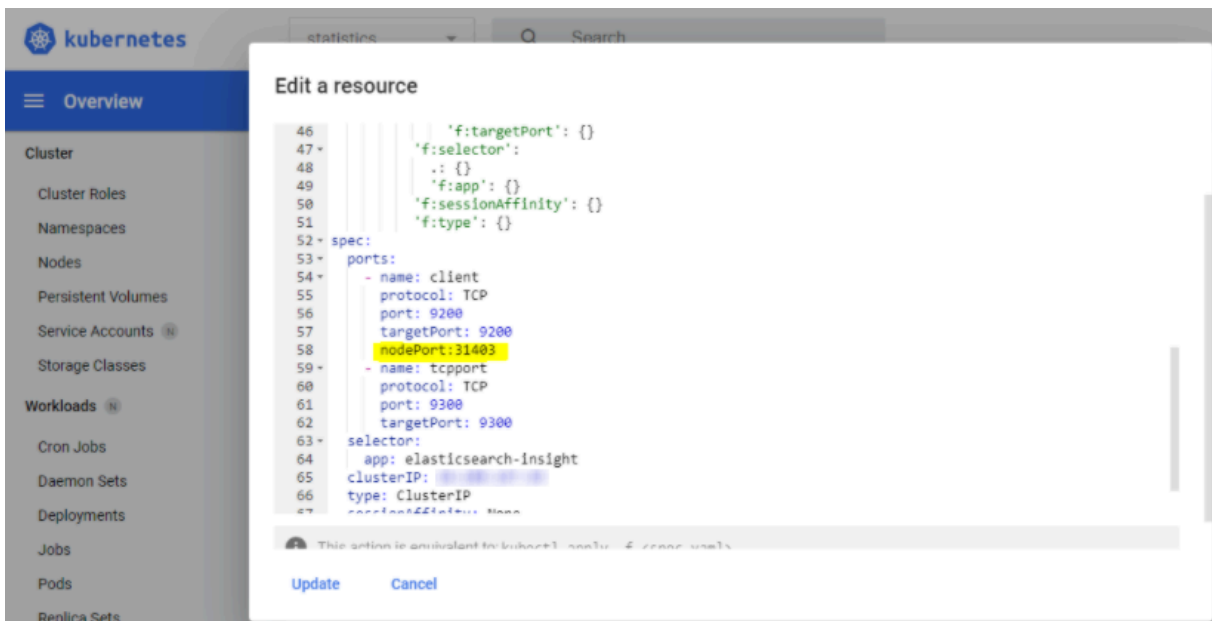
```
kubectl delete pods avx-insight-statistics-bot-3499c69cd6-4sdfs,avx-insight-subsystem-adc-4399c69ed6-4sdfs,avx-subsystems-7666cfb459-6q4m -n
absecon
```

To restart multiple Pods, enter the name of the pod in the above command with space.

18. In the case of Insight migration, continue till point 11.
19. Log in to the Kubernetes dashboard, enter statistics as the namespace.
20. Select services and search for **elasticsearch-insight**.



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



23. To restore the elastic data, go to the old installation path of AppViewX, and execute the following command: `appviewx --elastic-restore`

```
[appviewx] appviewx] appviewx --elastic-restore
*****
AppViewX 2020.3.0 elastic-restore
*****
Snapshot restored successfully
*****
```

24. 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:

```
[appviewx] scripts]$ kubectl get services -n statistics
NAME                                TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
elasticsearch-insight               NodePort    10.10.1.10       <none>           9200:31403/TCP,9300:30222/TCP 136m
[appviewx] scripts]$
```

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)

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

To create 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 UserName
```

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

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.

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.

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.

Adding a Node to the Cluster

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

To add a node to a 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:

```
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**.

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. 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
derive_configmap.py
cp -r * <installer_location>/appviewx_kubernetes/scripts/
```

10. To add the node, execute the respective command:

For FP6 upgrade installations:

- For installations upgraded to FP6, the latest FP6 installer is needed for the installation artifacts.
- Extract the FP6 installer to **/home/appviewx/FP6_installer/**
- Run the command:

```
sudo ./appviewx_add_node.sh /home/appviewx/FP6_installer/appviewx_kubernetes
```

- Go to the Prompt section for the next step.



Note: For downloading the FP6 installer, use the below-mentioned link: <https://release.appviewx.com/downloadArtifact?id=606>

For FP10 upgrade installations:

- For installations upgraded to FP10, the latest FP10 installer is needed for the installation artifacts.
- Extract the FP10 installer to **/home/appviewx/FP10_installer/**
- Run the command:

```
sudo ./appviewx_add_node.sh /home/appviewx/FP10_installer/appviewx_kubernetes
```

- Go to the Prompt section for the next step.



Note: For downloading the FP10 installer, use the below-mentioned link: <https://release.appviewx.com/downloadArtifact?id=870>

For pre-FP6, FP6, and post-FP6 fresh installations:

- For pre-FP6 (docker-based installations) or fresh FP6 installations, the artifacts are already present in the installer directory.
- Run the command :

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

- Go to the Prompt section for the next step.

Prompt

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

The command will prompt you to enter the following details:

- IP addresses of new master nodes
- datacentername:hostname of the new master nodes
- IP address of the new worker nodes
- datacentername:hostname of the new worker nodes
- Confirmation to deploy the node in the new datacenter
- Details of plugins that you want to deploy on the new nodes
- (*Optional step*) Enter the name of the new datacenter in which strict routing needs to be enabled (comma separated values)

```
[RPK-appviewx@192.168.94.98]$ sudo ./appviewx_add_node.sh
Enter the ip-addresses of the master(Control-plane) nodes you want to add(Comma-separated Values)

Enter the data-center:hostnames of the master(control-plane) nodes you want to add(Comma-separated Values)

Enter the ip-addresses of the worker(slave) nodes you want to add(Comma-separated Values)
1.2.3.4
Enter the datacenter:hostnames of the worker(slave) nodes you want to add(Comma-separated Values)
datacenter:hostname
Do you want to deploy in the new datacenter datacenter? Enter yes/no
yes
creating new datacenter datacenter
1.avx_commons
2.avx_config_server
3.avx_platform_core
4.avx_platform_queue
5.avx_platform_gateway
6.avx_platform_web
7.avx_subsystems
8.avx_vendors
9.avx_subsystems_sync
10.avx_platform_report_generator
11.avx_visual_page_builder
12.avx_platform_logforwarding
13.avx_vendor_cert_network_discovery
Enter serial number(Comma separated value) against plugins to be deployed on datacenter, press enter to deploy it on all plugins
9,14
Enter the datacenter names for which strict routing needs to be enabled(Comma-separated Values):
datacenter
```

The command will add the node to the cluster and deploy the new plugins on the node.

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

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

Understanding the Best Practices on Reboot Sequence

This section provides information on the best practices to be followed for rebooting the operating system after security patching.



Note: Before you perform these steps, ensure that all prerequisites are complied with as mentioned in the [Configuring YUM](#) section.

The steps are to be executed in the order given below.

1. Log in into the AppViewX worker node from where the installation has been initiated.
2. Navigate to `<installer directory path>/appviewx_kubernetes/scripts`
3. Take a backup of the scripts directory from `/appviewx_kubernetes/scripts`
4. Download the latest `scripts.tar.gz` from the release portal.
5. Copy the existing `appviewx.conf` file from the older scripts folder to the newly downloaded scripts folder from the release portal.
6. Execute the commands from the installer location/scripts folder.



Note: The Stop all and Start all commands are applicable only for a multi node setup.

7. To drain all the pods, execute the following command:

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

The command will drain the pods in the nodes in the following order; Worker, Secondary master(if any), Master.

8. Shut down the nodes in the order mentioned in step 7.
9. Start the nodes in the reverse order; Primary master, Secondary masters, and Workers from the primary mongo as per `appviewx.conf` entries.
10. To start all the pods in the nodes, execute the following command:

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

This command will start the pods in the nodes in the following order; Master, Secondary master(if any), Worker.

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](#)
- [Troubleshooting 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.

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
```

Troubleshooting Alerts



Note: For troubleshooting issues, please refer to the [Troubleshooting.ditamap](#) section.

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](#)
- [Performing a Backup for MongoDB and Vault](#)
- [Restoring a MongoDB Backup](#)
- [Restoring the Vault Backup](#)
- [Troubleshooting Backup and Restore Operations](#)

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`
- `vault_restore.sh`
- `mongo_restore.sh`

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

Performing a Backup for MongoDB and Vault

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

```
sh mongo_backup.sh <appviewx_installed_Path> <appviewx_installer_location_path>
```

```
sh vault_backup.sh <appviewx_installed_Path> <appviewx_installer_location_path>
```

where,

- `appviewx_installed_Path` - specifies the path where AppViewX is installed
- `appviewx_installer_location_path` - specifies the location of the AppViewX installer file.

```

-bash-4.25 ./mongo_backup.sh /home/appviewx/appviewx /home/appviewx/
script dir: /home/appviewx/appviewx_kubernetes/scripts
Initiating config server identification...

Copying backup script to all config-server nodes...
mongo_backup.sh
vault_backup.sh
File copied to: pesrv05-devops01-150-145
mongo_backup.sh
vault_backup.sh
File copied to: pesrv05-devops02-150-146
avx-config-server-5c0f08d955-sdnjr

Triggering mongo backup...
Defaulting container name to avx-config-server.
Use 'kubectl describe pod/avx-config-server-5c0f08d955-sdnjr -n dc1' to see all of the containers in this pod.
Mongo backup directory available
Vault backup directory available
Mongo Backup Script begins
Logging into Vault: http://vault-active.avx.svc.cluster.local:8200/v1/auth/approle/login

```

The command creates a backup of both MongoDB and Vault. The backup files will be available as listed below:

- Mongo Backup file path: `<Installation path>/logs/<mongo file with timestamp>`
- Vault Backup file path: `<Installation path>/logs/<vault file with timestamp>`

```

Script execution complete.
Mongo Backup File Details: pesrv05-devops01-150-145:/home/appviewx/appviewx/logs/mongo_backup_Mon_Feb_22_18_33_24_UTC_2021.tar.gz
Script execution complete.
Vault Backup File Details: pesrv05-devops01-150-145:/home/appviewx/appviewx/logs/vault_backup_Mon_Feb_22_18_33_35_UTC_2021
-----
Log file: pesrv05-devops01-150-145:/home/appviewx/appviewx/logs/mongo_backup_02222021_183311.log

```



Note: The backup is maintained for a period of five days. The backup data will be purged after this period.

Restoring a MongoDB Backup

1. Open the terminal.
2. Execute the following command with parameters like Installation Path, appviewx installer location Path, target pod name (avx-config-server), and mongo backup file name.

```
sh mongo_restore.sh <appviewx_installed_Path> <appviewx_installer_location_path> <avx-config-server> <mongo-backup-file-path>
```

```
root@4-25-45-mongo-restore-03:/home/appviewx/appviewx:/home/appviewx/avx-config-server-5c6f68d955-dvtlz:/home/appviewx/appviewx/logs/mongo_backup_Mon_Feb_22_18_33_35_UTC_2021.tar.gz
Identifying the target pod
Copying backup dir to the container...
```

Restoring the Vault Backup

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

```
sh vault_restore.sh -p <vault_backup_file_path>
```

where,

- `vault_backup_file_path` - specifies the location of the vault backup file to be restored.

Example:

```
sh vault_restore.sh -p /home/appviewx/appviewx/logs/vault_backup_Mon_Feb_22_18_33_35_UTC_2021
```

```
-bash-4.2$ sh vault_restore.sh -p /home/appviewx/appviewx/logs/vault_backup_Mon_Feb_22_18_33_35_UTC_2021
Backup file path is /home/appviewx/appviewx/logs/vault_backup_Mon_Feb_22_18_33_35_UTC_2021
Vault Restore Script begins
Namespaces: avx-config-server-5c6f68d955-dvtlz -n dc1
Additional Config is {"deletion_allowed": "True", "allow_plaintext_backup": "True", "exportable": "True"}
Restore Request Data is {"backup": "eyJwb2xpY3kiOnsibmFtZSI6InVFeW5iVWhid00iLCJrZXlziip7IjEiOnsia2V5IjoiaR1FEYtdzajJE
```

Troubleshooting Backup and Restore Operations



Note: For troubleshooting issues, please refer to the [Troubleshooting.ditamap](#) section.

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.

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](#)

Managing Logs using Kibana

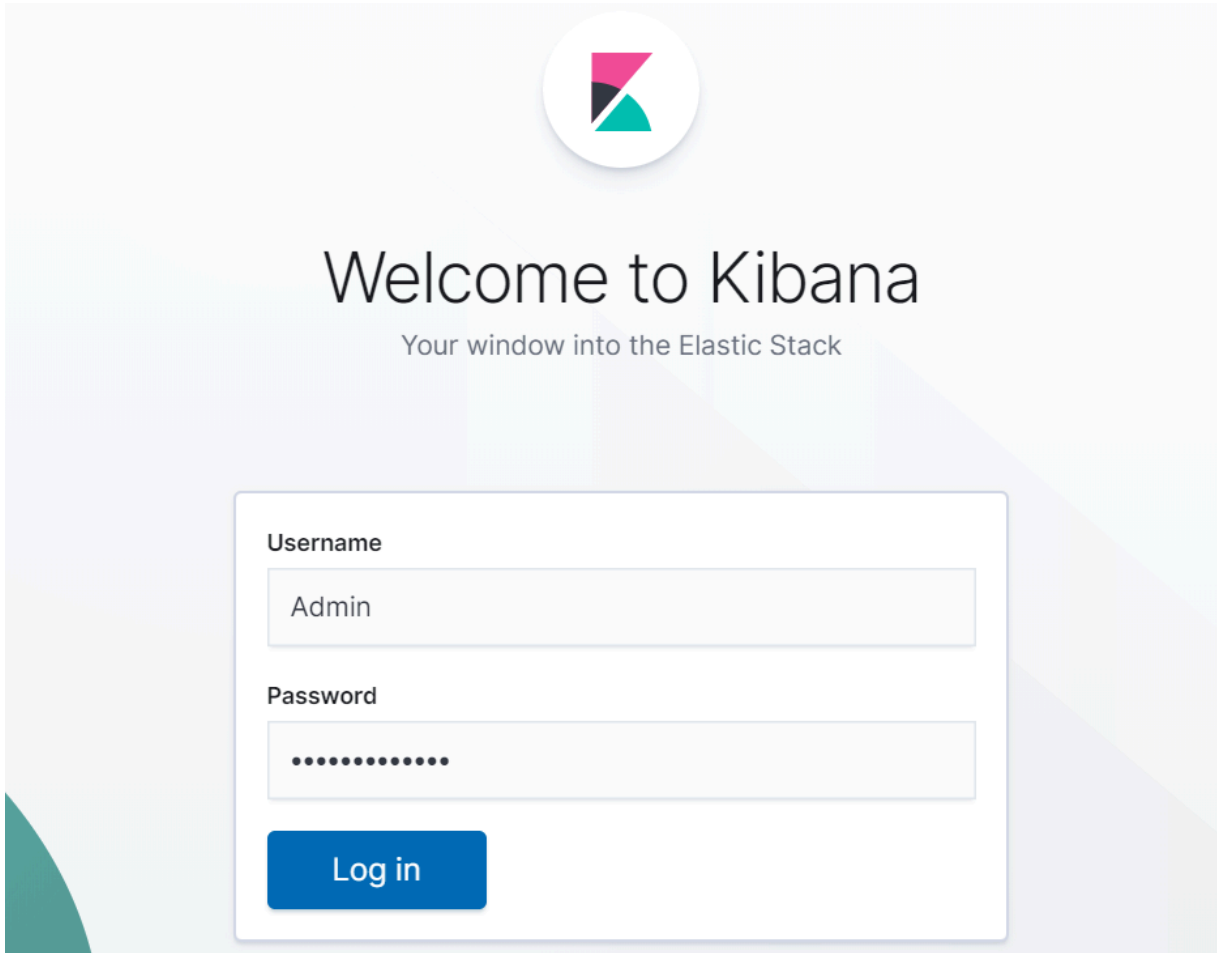
Kibana is an open user interface that enables you to graphically represent the log files and monitor system performance.

Before using Kibana, ensure that you have the following:

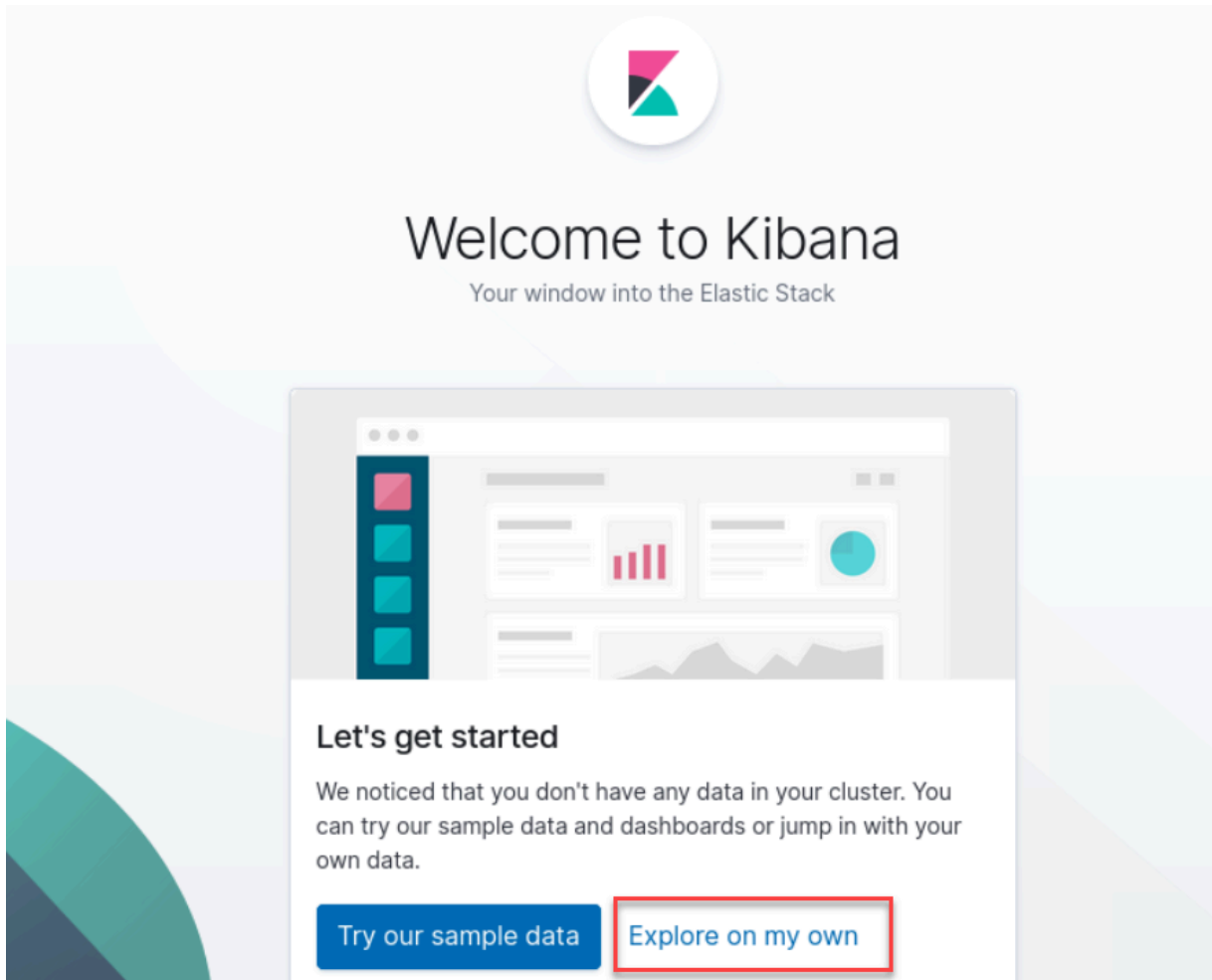
- Kibana Web URL - mentioned in the `<INSTALLATION_PATH>/appviewx_configuration` file
- Kibana Username - mentioned in the `<INSTALLATION_PATH>/appviewx_configuration` file
- Kibana Password - mentioned in the `<INSTALLATION_PATH>/appviewx_configuration` file
- [Accessing Kibana](#)
- [Creating an Index Pattern](#)
- [Viewing Logs](#)
- [Generating a Report](#)

Accessing Kibana

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

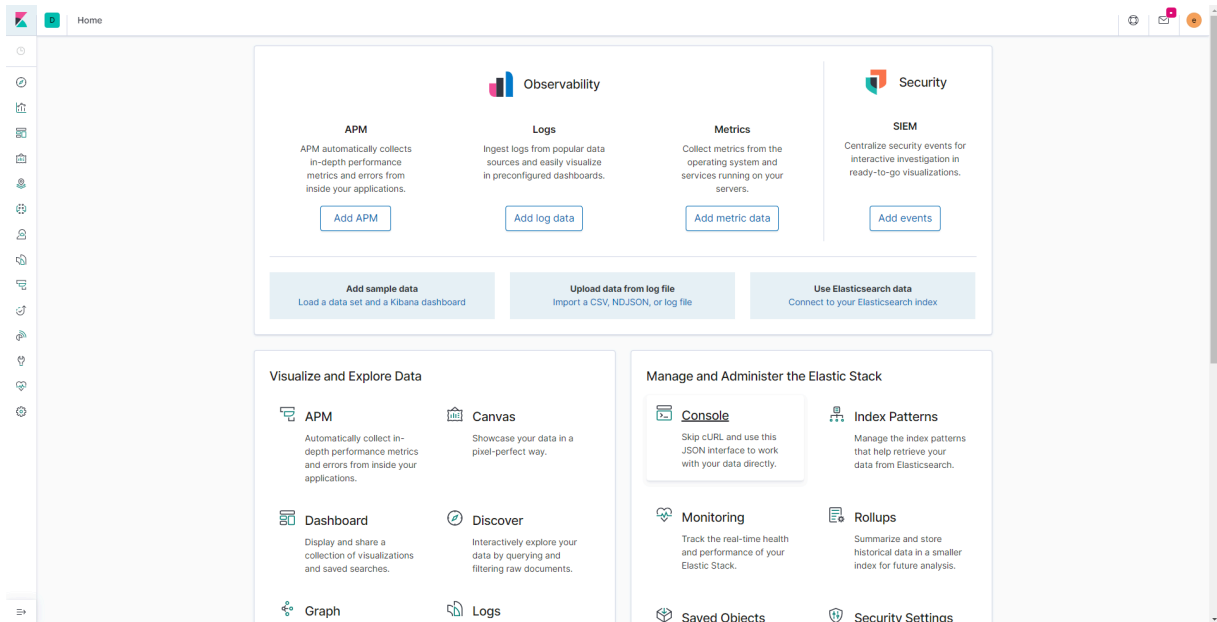


The **Let's get started** page is displayed.



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

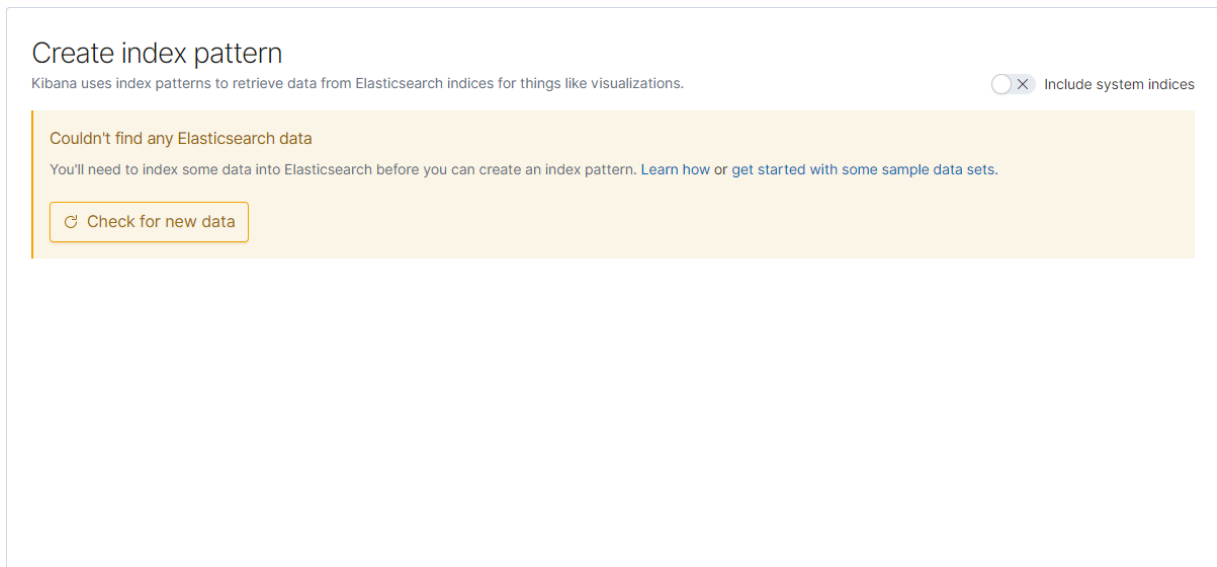
The **Home page** is displayed.



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.

Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations. Include system indices

Step 1 of 2: Define index pattern

Index pattern

You can use a * as a wildcard in your index pattern.
You can't use spaces or the characters \, /, ?, *, <, >, |.

Your index pattern can match any of your **3 indices**, below.

.kibana_1
.kibana_task_manager_1
.security-7

Rows per page: 10 ▾

[> Next step](#)

4. Under Index pattern, enter `.*`.

5. Click **Next step**.

The **Configure settings** page is displayed.

Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations. Include system indices

Step 2 of 2: Configure settings

You've defined `.*` as your index pattern. Now you can specify some settings before we create it.

Time Filter field name [Refresh](#)

The Time Filter will use this field to filter your data by time.
You can choose not to have a time field, but you will not be able to narrow down your data by a time range.

[> Show advanced options](#)

[< Back](#) [Create index pattern](#)

6. From the **Time Filter field name** list, select `@timestamp`.

7. Click **Create Index Pattern**.

The system creates an index pattern.

The screenshot shows the Kibana Mapping API interface for the index pattern `.*`. At the top, there is a star icon, a refresh icon, and a delete icon. Below this, the "Time Filter field name" is set to `@timestamp` with a "Default" dropdown. A descriptive text states: "This page lists every field in the `.*` index and the field's associated core type as recorded by Elasticsearch. To change a field type, use the Elasticsearch Mapping API".

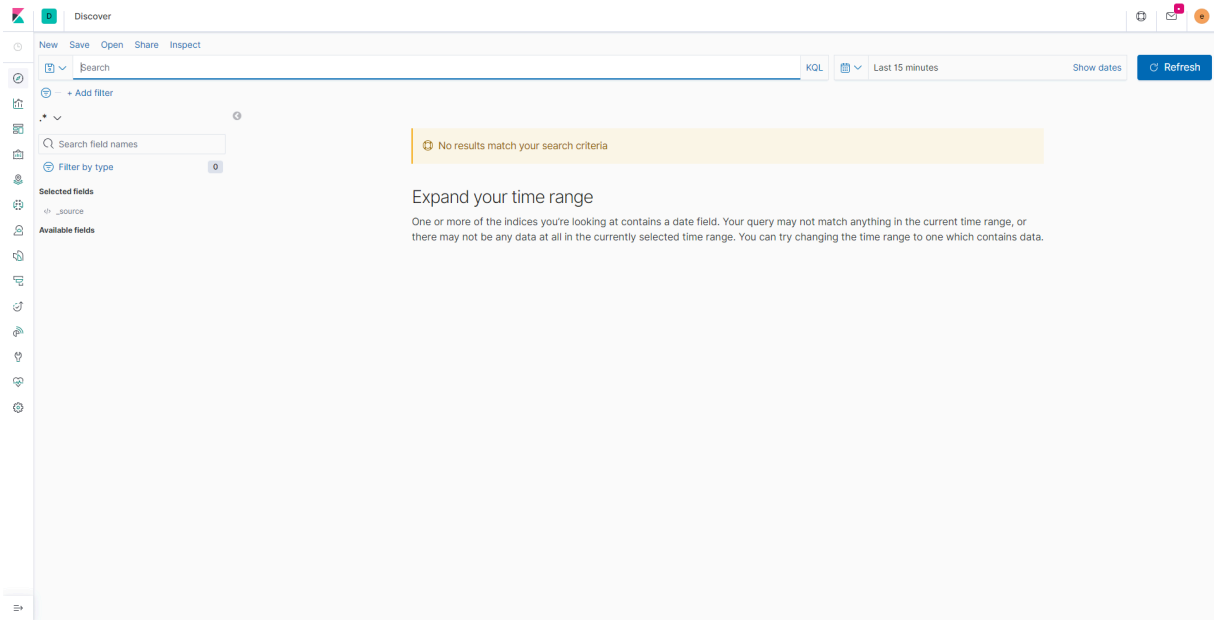
Summary statistics are shown: Fields (391), Scripted fields (0), and Source filters (0). A search filter and a dropdown for "All field types" are present.

Name	Type	Format	Searchable	Aggregatable	Excluded
<code>@timestamp</code>	date		●	●	
<code>_id</code>	string		●	●	
<code>_index</code>	string		●	●	
<code>_score</code>	number				
<code>_source</code>	<code>_source</code>				
<code>_type</code>	string		●	●	
<code>access_token.invalidated</code>	boolean		●	●	
<code>access_token.realm</code>	string		●	●	
<code>access_token.user_token.authentication</code>	unknown				
<code>access_token.user_token.expiration_time</code>	date		●	●	

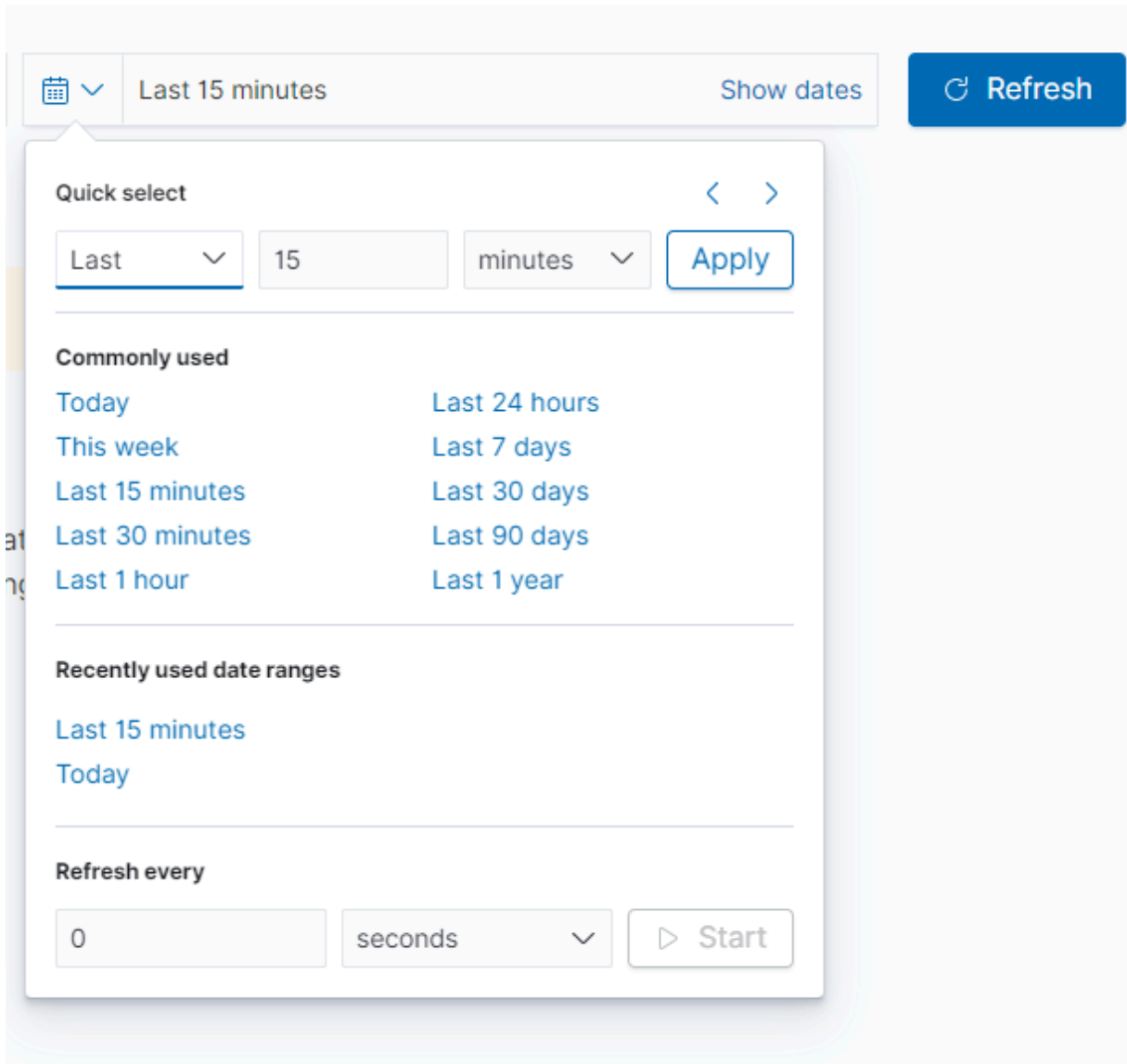
At the bottom, it shows "Rows per page: 10" and a pagination control with page numbers 1, 2, 3, 4, 5, ..., 40.

Viewing Logs

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



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:

- a. Click **Save**.
- b. Enter a valid name to save the search.

Save search ×

Save your Discover search so you can use it in visualizations and dashboards

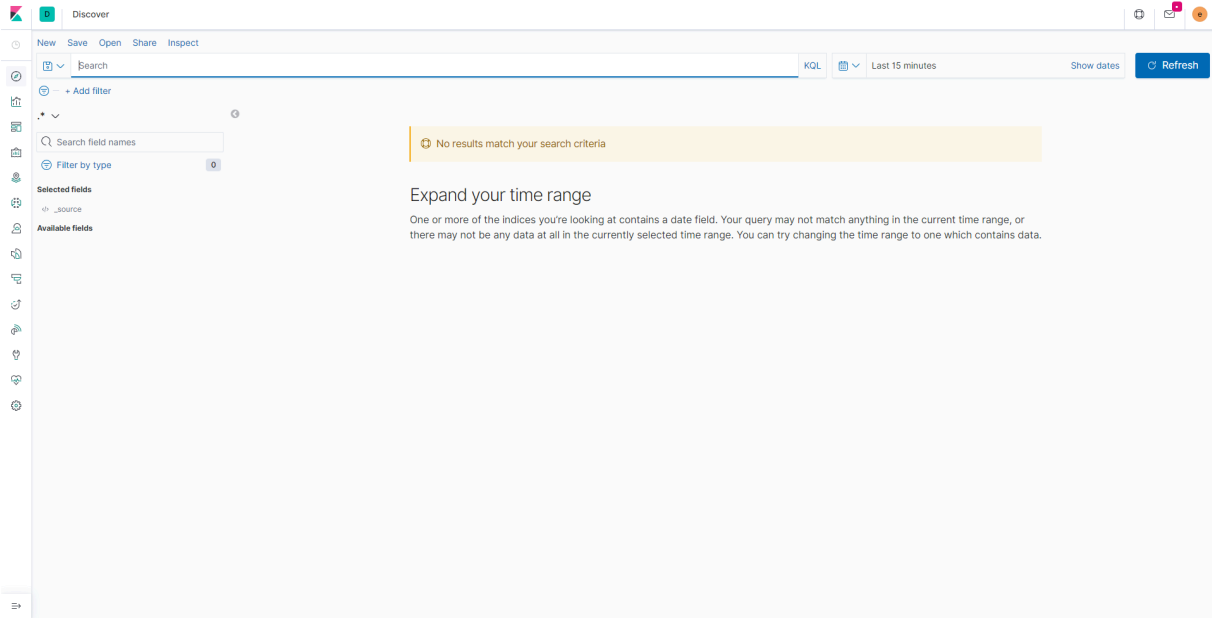
Title

CancelSave

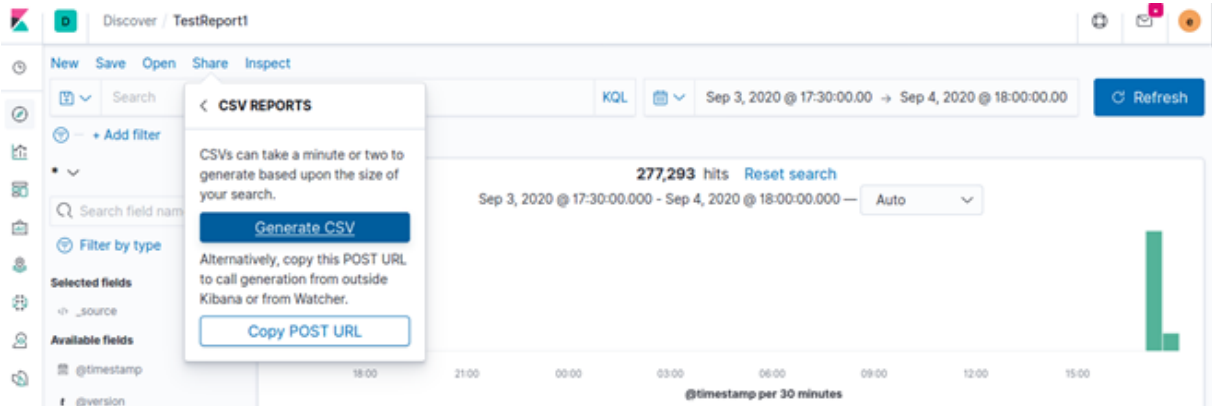
Generating a Report

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>`).

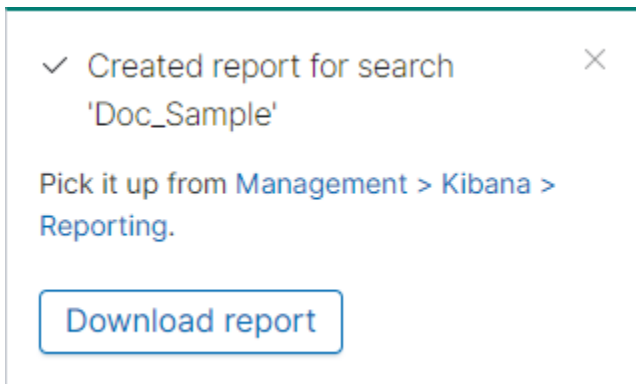
1. Login to Kibana.
2. Under **Visualize and Explore Data**, click **Discover**.
The **Discover** page is displayed.



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



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.

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
avx-commons-696f66b88f-68pnq	2/2	Running	14	4d19h	10.10.10.10	gs-apvx-dev86-10.10.10.10	<none>
<none>							
avx-config-server-765bc549c8-h92wt	2/2	Running	13	4d19h	10.10.10.11	gs-apvx-dev86-10.10.10.11	<none>
<none>							
avx-platform-core-97d99cddd-c9q6c	2/2	Running	14	4d19h	10.10.10.12	gs-apvx-dev86-10.10.10.12	<none>
<none>							
avx-platform-gateway-7c957fdd4f-2br5d	2/2	Running	15	4d19h	10.10.10.13	gs-apvx-dev86-10.10.10.13	<none>
<none>							
avx-platform-queue-9dbcc9ccb-txns5	2/2	Running	14	4d19h	10.10.10.14	gs-apvx-dev86-10.10.10.14	<none>
<none>							
avx-platform-web-6b4df49fb6-2phqs	2/2	Running	0	4d19h	10.10.10.15	gs-apvx-dev86-10.10.10.15	<none>
<none>							
avx-subsystems-75db48b9b4-5gfgk	2/2	Running	13	4d19h	10.10.10.16	gs-apvx-dev86-10.10.10.16	<none>
<none>							
avx-subsystems-75db48b9b4-8xn15	2/2	Running	18	10d	10.10.10.17	gs-apvx-dev86-10.10.10.17	<none>
<none>							
avx-subsystems-75db48b9b4-9hwlv	2/2	Running	13	4d19h	10.10.10.18	gs-apvx-dev86-10.10.10.18	<none>
<none>							
avx-subsystems-75db48b9b4-rn22c	2/2	Running	18	10d	10.10.10.19	gs-apvx-dev86-10.10.10.19	<none>
<none>							
avx-subsystems-sync-7f59dc8b9-nlsc6	2/2	Running	18	10d	10.10.10.20	gs-apvx-dev86-10.10.10.20	<none>
<none>							
avx-vendors-586f9db568-8vncv	2/2	Running	0	4d16h	10.10.10.21	gs-apvx-dev86-10.10.10.21	<none>
<none>							

```
[appviewx@gs-apvx-dev86 ~]$
```



Note: For troubleshooting issues, please refer to the [Troubleshooting.ditamap](#) section.

Offline Patching for CentOS

This section provides the steps, in chronological order, to perform patching in the customer environment offline. The steps are categorized into tasks for the customer and tasks for the AppViewX team.

Steps for Customers

Follow the steps below to patch the customer nodes where internet access is restricted.

1. Run the following command to generate a log file with the list of installed RPMs in customer nodes separately:

```
rpm -qa >> installed_rpms_<node_name>.log
```

This command will create a file *installed_rpms_<node_name>.log* that contains a list of all the installed rpms.

2. SCP the log files of each node to your laptop and upload them as input in the following Jenkins job: http://ci.appviewx.in/job/Offline_patching_of_the_centos_packages/
3. Do not edit the default HOST_IP and HOST_PASSWORD.
4. After successful execution of the Jenkins job, download the rpms_dir.tar.gz and SCP to respective nodes.
5. Unzip rpms_dir.tar.gz and cd to rpms_dir. Run the following command to install the latest RPMs:

```
tar -xvf rpms_dir.tar.gz
cd rpms_dir
rpm -Uvh * --force
```

Steps for the AppViewX Team

1. SCP the log files of each node to your laptop and upload it as input in the following Jenkins job:
http://%3Cjenkins_server%3E/job/Offline_patching_of_the_centos_packages



Note: Replace jenkins_server with the actual appviewx jenkins server IP or hostname.

2. Do not edit the default HOST_IP and HOST_PASSWORD.
3. After the successful execution of Jenkins job, download the rpms_dir.tar.gz.
4. Install the generated rpms with the lab setup.
5. Perform BVT and Dev Sanity test case execution to ensure the system is healthy.
6. After verification, provide the RPMs to the customer in a tarball rpms_dir.tar.gz.

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`.

```
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$
```

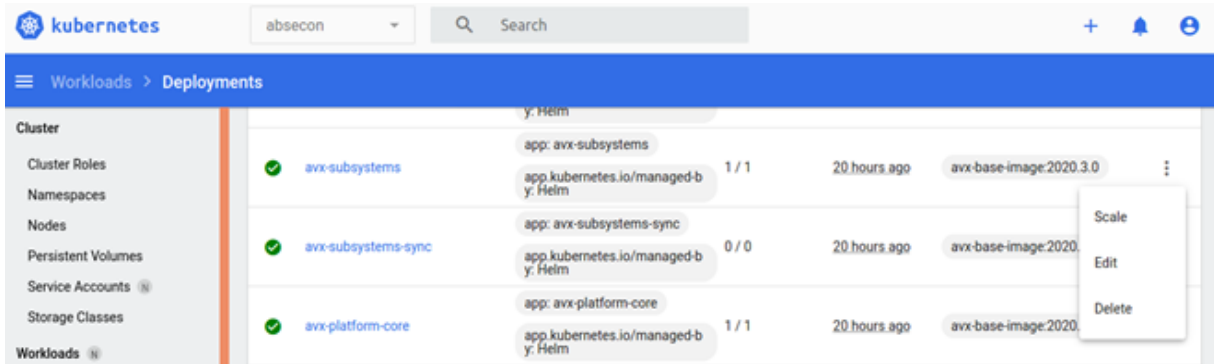
5. Save and exit the `appviewx.conf` file.
6. Navigate to the `scripts` directory.
7. In the `scripts` directory, execute the following command:

```
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.

Scale a resource

deployment avx-subsystems will be updated to reflect the desired replicas count.

Desired replicas * Actual replicas

i This action is equivalent to: `kubectl scale -n absecon deployment avx-subsystems --replicas=1`

[Scale](#) [Cancel](#)

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.

Scale a resource

deployment avx-subsystems will be updated to reflect the desired replicas count.

Desired replicas * Actual replicas

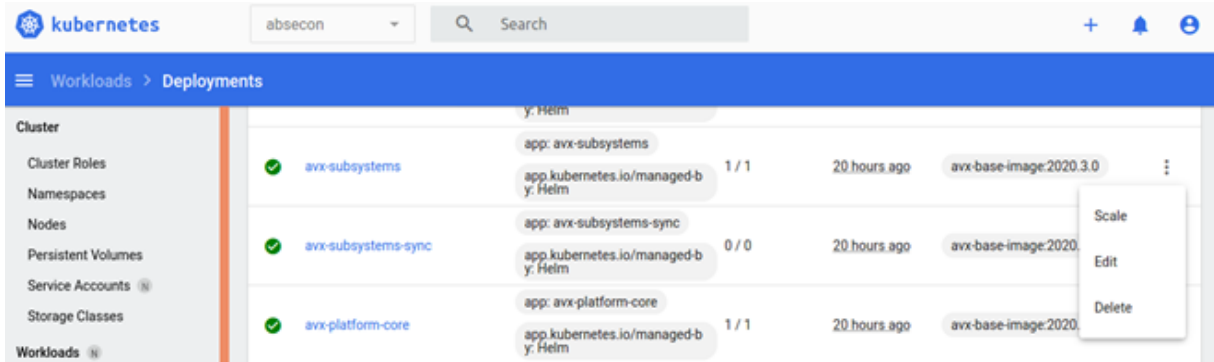
i This action is equivalent to: `kubectl scale -n absecon deployment avx-subsystems --replicas=1`

[Scale](#) [Cancel](#)

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

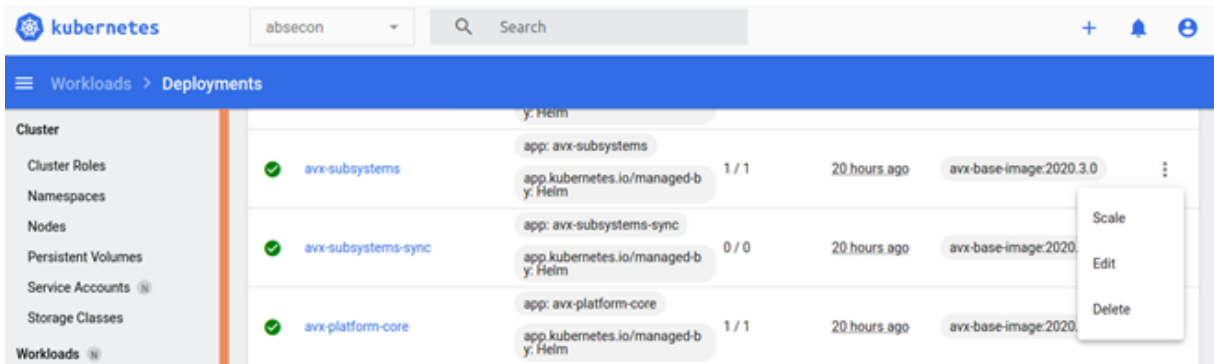


Scaling a Plugin

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.

Scale a resource

deployment avx-subsystems will be updated to reflect the desired replicas count.

Desired replicas * Actual replicas

 This action is equivalent to: `kubectl scale -n absecon deployment avx-subsystems --replicas=1`

[Scale](#) [Cancel](#)

- Update the value of the **Desired replicas** parameter to increase or decrease the number of pods for a plugin.
- 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:

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

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

```

320 image: 'avx-base-image:2020.3.0'
321 command:
322   - /bin/bash
323   - '-c'
324 args:
325   - >-
326     source /appviewx/dependencies/properties/hsm && useradd -u 1000
327     appviewx && chown -R appviewx:appviewx /usr/lib/jvm && chown -R
328     appviewx:appviewx /etc/pki/ca-trust/extracted/java && chown -R
329     appviewx:appviewx /etc/pki/java/ && chmod 777
330     /etc/pki/ca-trust/extracted/java/cacerts && su appviewx -s
331     /bin/bash -c "source /appviewx/dependencies/properties/hsm && java
332     -Xms256m -Xmx2g | -cp
333     /appviewx/avx_vendor_a10/20.3.0.0/avx_vendor_a10.jar:/appviewx
     /avx_vendor_akamai/20.3.0.0/avx_vendor_akamai.jar:/appviewx
     /avx_vendor_amazonlb/20.3.0.0/avx_vendor_amazonlb.jar:/appviewx
     /avx_vendor_automation/20.3.0.0/avx_vendor_automation.jar:/appviewx
     /avx_vendor_avi/20.3.0.0/avx_vendor_avi.jar:/appviewx/avx_vendor_bigiq/20
     .3.0.0/avx_vendor_bigiq.jar:/appviewx/avx_vendor_cert_adc/20.3.0.0
     /avx_vendor_cert_adc.jar:/appviewx/avx_vendor_cert_ca/20.3.0.0
     /avx_vendor_cert_ca.jar:/appviewx/avx_vendor_cert_cloud/20.3.0.0
     /avx_vendor_cert_cloud.jar:/appviewx/avx_vendor_cert_firewall/20.3.0.0
  
```

Update Cancel

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`

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

The screenshot shows the Kubernetes management console interface. The top navigation bar includes the 'kubernetes' logo, a dropdown menu set to 'default', a search bar, and utility icons for adding, notifications, and user profile. The left sidebar lists various cluster components under 'Cluster' and 'Workloads'. The main content area is divided into two sections: 'Jobs' and 'Pods'.

Jobs Section:

Name	Labels	Pods	Created ↑	Images
mongoutil-mongoseed	<ul style="list-style-type: none"> controller-uid: 22152af0-2b64-48f9-8423-63a00835dadd job-name: mongoutil-mongoseed 	0 / 1	a.day.ago	mongo

Pods Section:

Name	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created ↑
mongoutil-mongoseed-p428z	<ul style="list-style-type: none"> controller-uid: 22152af0-2b64-48f9-8423-63a00835dadd job-name: mongoutil-mongoseed 	appviewx-kube-install-94-190	Terminated: Completed	0	-	-	a.day.ago

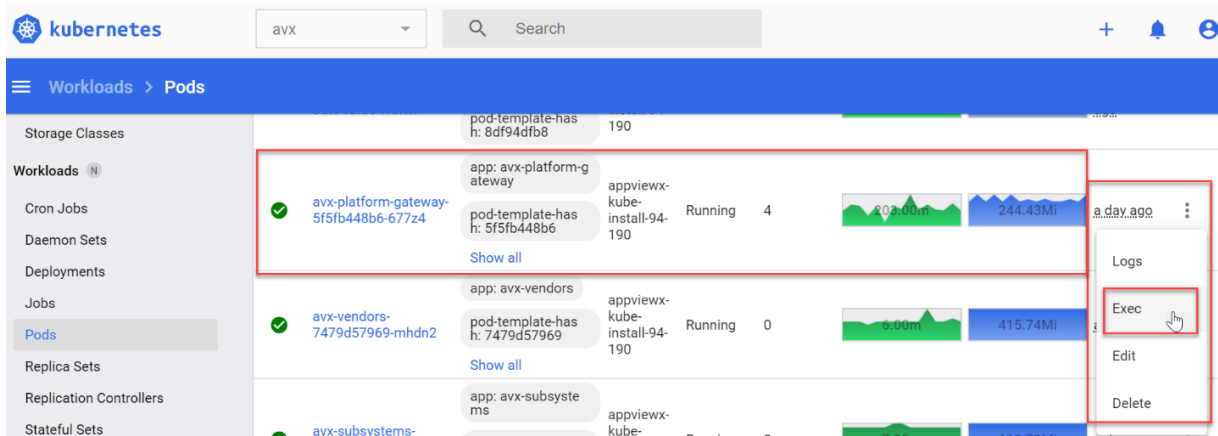
Viewing the POD Status

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

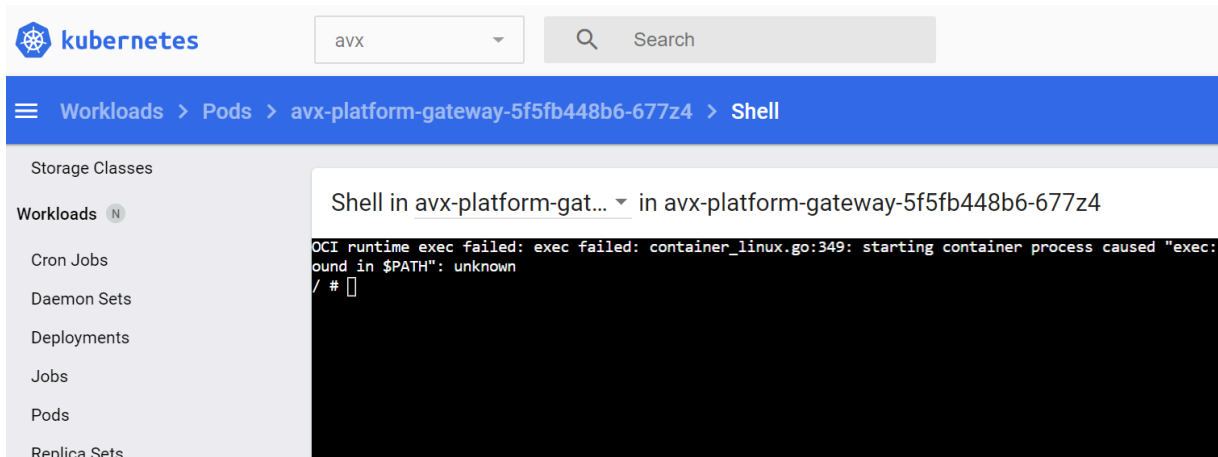
This screenshot shows the Kubernetes management console with the namespace dropdown menu highlighted in red, containing the text 'avx'. The main content area displays the 'Workloads' section, specifically a 'CPU Usage' graph. The graph shows CPU usage in cores over time, with a peak of 0.5 cores. The left sidebar lists various cluster components under 'Cluster' and 'Workloads'.

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**.

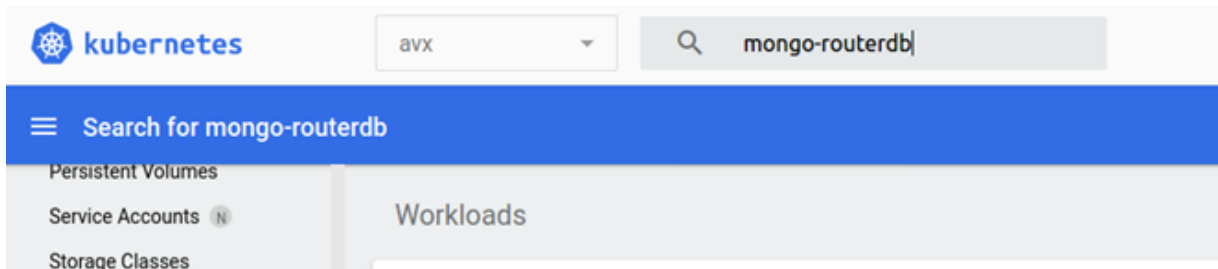


The Pod command line shell is displayed.

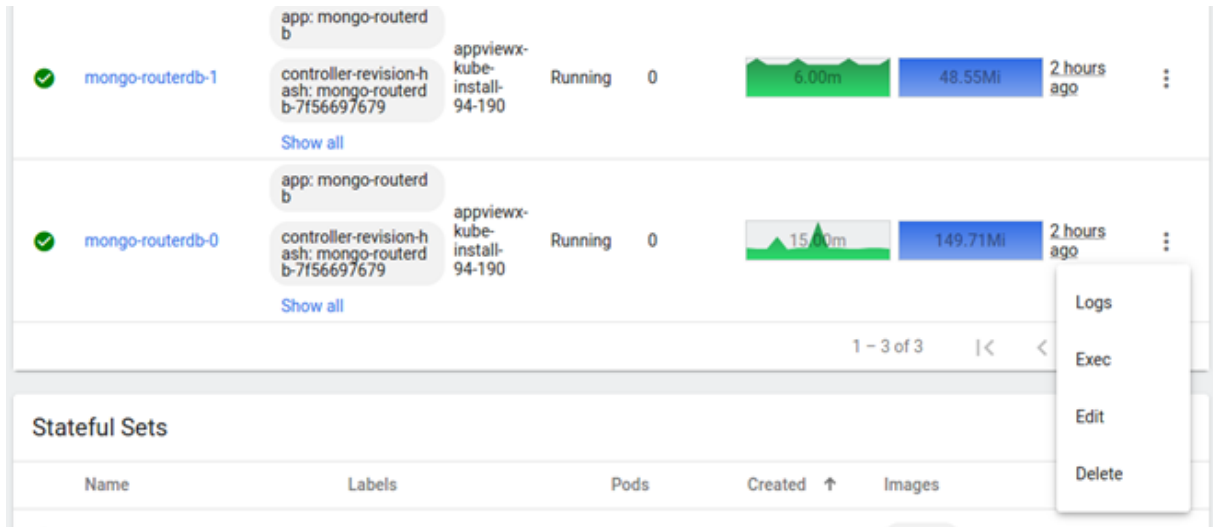


Accessing the Database Command Line

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



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



- To launch the mongo db prompt, execute the following command: `<mongo>`
- Execute the following command:

```
<use admin>
```

- Execute the following command:

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



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

Shell in mongo-routerdb... ▾ in mongo-routerdb-0

```
groups: cannot find name for group ID 1337
root@mongo-routerdb-0:/# mongo
MongoDB shell version v4.2.6
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("ad40632e-2fee-4569-9f4d-8168a180484c") }
MongoDB server version: 4.2.6
mongos> use admin
switched to db admin
mongos> db.auth("admin","PZ@Pg71SawX5abX0")
1
mongos> █
```

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: `mongoexport --username admin --password <password> --db=appviewx --collection=<collectionName> --out=<fileName>.json --authenticationDatabase admin` Change the fields highlighted in bold with the desired values according to your setup. 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`

Deleting a Node from the Cluster

Steps to Delete a Node

Run the following steps from the installer to delete a node from the existing kube cluster:

1. To safely evict all of your pods from a node before you perform maintenance on the node, run the command,

```
kubectl drain <node-name> --delete-local-data --ignore-daemonsets --timeout 30s --grace-period=0 --force --disable-eviction
```

2. To gracefully delete the node, run the command

```
kubectl delete node <node-name>
```

3. Edit the **appviewx.conf** manually and delete all entries of the node that have been deleted.

Post-Delete Steps on the Node

Run the following steps on the deleted node:

1. Copy installer tar file from installer node to the deleted node.
2. To extract the installer tar file, run the command

```
tar -xvf <installer>
```

3. Navigate to the scripts folder using the command

```
cd <installer location>/appviewx_kubernetes/scripts
```

4. To remove the config file from the deleted node, run the command

```
rm -rf ~/.kube/config
```

5. To copy the appviewx.conf from the conf template file, run the command

```
cp appviewx.conf.template appviewx.conf
```

6. Navigate to the uninstall folder using the command

```
cd uninstall
```

7. To delete all the installed packages in the deleted node, trigger the `uninstall.sh` command as below

```
./uninstall.sh
```

Chapter 7: Upgrading AppViewX

- [Infra Readiness](#)
- [Backups and VMSnapshots](#)
- [Upgrading AppViewX](#)

Infra Readiness

Perform these steps before upgrading AppViewX.

1. To check the status of pod (if they are in the running state) execute the commands below.

```
kubectl get pods -A
```

```
kubectl get nodes -A
```



Note: By default, all the pods should be up and running and all the nodes should be in the 'ready' state.

2. To verify the hard disk space usage execute the commands below.

```
./appviewx.sh --run-on-all 'df -h <installation_path>'
```

```
./appviewx.sh --run-on-all 'df -h /var/log'
```



Note: The hard disk space usage should always be less than or equal to 70%.

3. To verify the date values in all the nodes execute the command below.

```
./appviewx.sh --run-on-all 'date'
```



Note: Dates must be the same in all the nodes.



Troubleshooting: If any of the above values are not as specified, contact AppViewX TechSupport.

Backups and VMSnapshots

1. To take backup of mongo and vault, navigate to the installer node trigger `appviewx_kubernetes/scripts`

Refer to [Performing a Backup for MongoDB and Vault](#)

2. Securely copy files tar files to an external server using the command below.

```
scp <backups-tar-location> <remote-username>@<remote-hostname>:<remote-location>
```

3. Navigate to the installer node trigger `appviewx_kubernetes/scripts` and stop all the services using the command below.

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

4. Take VM snapshots
5. Start all the services using the command below.

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

Upgrading AppViewX

This section provides details about the upgrade process. Upgrade is required when the customer wants to migrate to a newer version of AppViewX. Upgrade is also recommended when the customer wants to include additional features that are not available in the version that they are using.



Note:

- The node selected for the upgrade must be the worker node and the database allocated node
- Before upgrading to v2020.3.0 from older versions, ensure that `avx_platform_amc` plugin is enabled and up and running. For more information, refer to the [Enabling the avx_platform_amc plugin](#) section.

To upgrade AppViewX from older versions to v2020.3.0:

1. Open the terminal.
2. To extract the contents of the **upgrade.tar.gz** file, execute the following command:

```
tar -xvzf upgrade.tar.gz
```


- `./upgrade.sh` - perform a serial upgrade where Kubernetes installation is done after the migration script execution.

```
[appviewx_kubernetes]$ cd ../
[appviewx_installer]$ ls
appviewx_kubernetes  appviewx_kubernetes_2020.3.0.tar.gz  AppViewX.tar.gz  upgrade.sh
[appviewx_installer]$ ./upgrade.sh
Preparing the installation workspace. This may take a few minutes.
Preparing libraries
```

- `./upgrade.sh --parallel` - perform a parallel upgrade where Kubernetes installation is done in parallel along with migration script execution.

```
[appviewx_installer]$ ./upgrade.sh --parallel
Preparing the installation workspace. This may take a few minutes.
Preparing libraries
Enter the AppViewX path to upgrade to the latest version: /home/appviewx/appviewx
Preparing upgrade from v12.3.0 to v2020.3.0
```

The upgrade command prompts you to enter the old AppViewX installation path.

10. Enter the installation path.

```
Preparing the installation workspace. This may take a few minutes.
Preparing libraries
Enter the old AppViewX installation Path: /home/appviewx/AppViewX
```

11. After you enter the installation path, the upgrade process is initiated. It prompts you to take an application backup. Enter Yes or No.

- Enter Yes, to take an application backup.
- Enter No, to skip the backup process.



Note: This process is performed using rsnapshot and application backup is triggered for all the nodes selected for the upgrade. After the application backup is complete, you can find the backup in the snapshots folder.

```
Preparing upgrade from v2019.3.0 to v2020.3.0
If you have already taken the application backup, specify "No" to skip and proceed further
Do you want to take application backup? [Yes/No] : No
```

12. After the backup process, AppViewX Configuration appears on the terminal. During the upgrade, the configuration of the existing AppViewX version is synced with the new version (v2020.3.0).

```

AppViewX Configuration
-----
MULTINODE | FALSE
SSH |
SSH_HOST | absecon
INSTALLATION_PATH | /home/appviewx/appviewx_cluster
ENABLED_PLUGINS | avx_platform_gateway_external,avx_platform_core,avx_platform_web,avx_crontab,appviewx_dependencies,avx_config_server,avx_platform_anc,avx_platform_report_generator,avx_vendors,avx_platform_gateway,avx_subsystems,avx_vendor_cert_network_discovery,avx_platform_queue,avx_vendor_cert_scep_agent

SSH_OTHER_USER | appviewx
avx_config_server | absecon
avx_platform_core | absecon
avx_platform_queue | absecon
avx_subsystems | absecon
avx_subsystems_sync | absecon
avx_vendors | absecon
avx_platform_gateway | absecon
avx_platform_web | absecon
avx_crontab | avx
ELK | FALSE
MONITORING | FALSE
MONGODB_HOST |
VAULT_HOST |
MASTER_HOST |
SECONDARY_MASTER_HOST |
WORKER_HOST |
ELASTICSEARCH_HOST |
API_ADDRESS |
INSIGHT | TRUE
SYSLOG | TRUE
INSIGHT_ELASTICSEARCH_HOST |
avx_platform_gateway_external | external-system
appviewx_dependencies | absecon
avx_platform_anc | absecon
avx_platform_report_generator | absecon
avx_vendor_cert_network_discovery | absecon
avx_vendor_cert_scep_agent | absecon

Do you want to change any of the above values? (Yes/No) [No] :Yes

```

The system prompts you to make changes to the AppViewX Configuration.

13. Enter **Yes** or **No**.

- Enter **Yes**, if you want to make changes to the AppViewX Configuration.
- Enter **No**, if you want to skip this step.

14. If you enter **Yes**, the Configuration Mode appears on the terminal. There are two modes:

- **Basic Mode** - To add or remove master nodes.
- **Advanced Mode** - To modify all configurations.

If you select the **Basic Mode**, you are prompted to enter only the below-listed parameters:

- Node Details
- Hostname Details
- Installation Path
- Kubernetes Master Nodes
- Kubernetes Secondary Master Nodes
- Kubernetes Worker Nodes

```

Do you want to change any of the above values? (Yes/No) [No] :Yes
Configuration Mode:
-----
1.Basic mode - To add/remove master nodes
2.Advanced Mode - To modify all configurations
Enter the preferred mode(1 or 2) :1
-----
Modifying Configuration File
-----
Note:
1.Use comma(,) to enter multiple values
2.Use "" to enter an empty value
3.Enter to retain the existing value for a key

Enter the node IPs where the application is to be deployed.
Use comma(,) to enter multiple values

Node Details [ ] :

```

If you select the **Advanced Mode**, you can modify all the available configurations.

```

Do you want to change the above AppViewX configuration ? (Yes/No) [No] :Yes
AppViewX Configuration Mode:
-----
1.Basic mode - To add/remove AppViewX nodes
2.Advanced Mode - To modify all the configurations
Enter the preferred configuration mode(1 or 2) :2
-----
Modifying Configuration File
-----
Note:
1.Use comma (,) to enter multiple values
2.Use "" to enter an empty value
3.Press Enter to retain the current value for a key and proceed to the next one
4.Re-enter all the values for a key as it does not add/append to the current value

Enter TRUE if AppViewX is installed on more than one node
Multi-node Installation [Current Value: TRUE] :

Enter the node IPs where the application is to be installed.
Use comma (,) to enter multiple values.

Node Details [Current Value: ] :

```

15. Make the required changes to the AppViewX Configuration such as nodes, components, and so on.
16. After you make the required changes to the AppViewX configuration, it prompts you to enter the AppViewX password. Enter the password to continue the upgrade process.

```
[Optional] Enter the hostnames of Worker nodes that are used to deploy the appviewx components. Use
Kubernetes worker Nodes [] :
Error in kube conf merge list index out of range
[08-28 12:15:41] p43163 {upgrade.py:1363} ERROR Error in kube conf merge list index out of range
Please enter appviewx password of absecon :

```

It takes 30-90 minutes for the upgrade process to complete depending on the size of the data.

17. After the upgrade is complete, a success message is displayed on the terminal.

```
Restoring the mongo collections from legacy to kube cluster
Copying the mongo collections to kube cluster. This process might take time so kindly wait
Mongo collections are copied. We are initiating the mongo restore
NAME: mongorestore
LAST DEPLOYED: Fri Sep  4 11:07:57 2020
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
Stopping legacy application components
Components are stopped
Upgrade is completed successfully. Kindly refer /Test/Test1/appviewx_cluster/.appviewx_configuration

```



Note: Once AppViewX installation is successful, refer to the [Monitoring and Maintaining AppViewX](#) section for maintenance.

- [Infra Readiness](#)
- [Backups and VMSnapshots](#)
- [Upgrading AppViewX](#)

Chapter 8: Uninstalling AppViewX

- [Troubleshooting Uninstall Issues](#)
- [Uninstalling AppViewX](#)

Troubleshooting Uninstall Issues



Note: For troubleshooting issues, please refer to the [Troubleshooting.ditamap](#) section.

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`
3. To start the uninstallation process, execute the following command: `./uninstall.sh`
4. Enter the node's credentials when prompted.

```
[appviewx@pesrv03-regression02-98-13 uninstall]$ cd
[appviewx@pesrv03-regression02-98-13 ~]$ cd /home/appviewx/ /scripts/uninstall/
[appviewx@pesrv03-regression02-98-13 uninstall]$ ./uninstall.sh
Please enter appviewx password of master:pesrv03-regression02-98-13 :
Please enter appviewx password of dc1:pesrv03-regression03-98-14 :
Please enter appviewx password of dc2:pesrv03-regression04-98-15 :█
```

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

- [Troubleshooting Uninstall Issues](#)
- [Uninstalling AppViewX](#)

Chapter 9: Troubleshooting

- [Troubleshooting](#)
- [AppViewX Installation Failed](#)
- [General Troubleshooting](#)

Troubleshooting

This section lists the issues encountered with AppViewX.

- [Troubleshooting](#)
- [AppViewX Installation Failed](#)
- [General Troubleshooting](#)

AppViewX Installation Failed

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

- [Common Installation Errors](#)
- [Frequently Faced Errors](#)

Common Installation Errors

Frequently Faced Errors

- Pre requisites not met- port not opened, insufficient disk/cpu, time not in sync, packages not found, hostname incorrect in configuration etc. please check for all the above items.
- Error while installing the docker

In some custom OS which the customer brings in, the linux packages that we bundle along with the installer might not be compatible with the OS. In such cases, we might need to install the relevant package to proceed further. The same can be seen from the log messages stating error while installing a package.

- Error while installing the docker

In some cases, we have seen there are intermittent errors from the OS while installing the docker. When you face an error in this stage, please try doing an uninstall of the application and reboot all the nodes and proceed with the installation.

- Error while initializing the kube master/worker

In certain cases, when uninstallation does not clean up the data properly, we can see errors while initializing kube master and worker. Please perform an uninstall, reboot all the nodes and go ahead with the install once when we face this error. Also there are cases where the installation fails because of the port connectivity issues. Please check if 6443, 10250, 2379 and 2380 ports are opened properly if a failure occurs in this stage.

- Error while initializing the mongodb chart

This specific error occurs after a timeout of 5 minutes to initialize the mongodb charts. When we face this error, it happens because the pods are not able to communicate between themselves. Use the following commands to verify that:

`Kubectl describe statefulset -n avx mongo-shareddb` - If we face any connectivity errors, this will give the specific error stating connection timed out.

- Node is enabled with IPv6 but the application is not.

Check the output of `ifconfig | grep -i inet6` when this shows an IPv6 address, we need to enable ipv6 in the `appviewx.conf` file, else the communication does not happen properly.

- IP in IP tunnelling is not enabled

When the IP in IP traffic is not enabled, (IPv4 protocol is not allowed) we will be facing the same issue. The prerequisite check script does not capture this. Therefore, we must have this confirmed.

- Error while installing the AppViewX plugins

When there is an error while installing AppViewX plugins, most probably there is an error in the configuration file. Please double check the configuration file and proceed with the execution of `plugins_install.sh` to install the plugins alone.

General Troubleshooting

- [Pods not started after installation](#)
- [Unable to login](#)
- [Error while downloading certificates](#)

Pods not started after installation

After the plugins are installed, it generally takes a maximum of 5 minutes for the pods/plugins to come up. When the pods does not come up even after 5 minutes, please perform the following troubleshooting steps:

- To check the pod status, execute the following command:

```
kubectl get pods -n dcname
```

- Check the node in which the pod is spinned and login into the node to check pod start logs.
- If no errors are seen, check the pod logs.
- When the subsystem pod is not started properly, there may be a failure in execution of the release scripts. Log in into the DB and execute the following command to see if there are any release scripts in the migration_failed state.


```
db.getCollection('avx_script_execution_info').find({"status":"MIGRATION_FAILED"}
```
- Execute the following query to manually mark this as completed:


```
db.getCollection('avx_script_execution_info').update({"status":"MIGRATION_FAILED"},{ $set: {"status":"VALIDATION_COMPLETED"}})
```

There are dependencies between the pods and this could also be a cause for a specific pod to not get started. You must identify the root pod that is causing the problem.

Unable to login

You may have issues where the users lost connectivity to the application in an environment which was running properly. Please perform the following steps to troubleshoot this issue:

- Check the pod status in all namespaces. When some of the pods are in not running state, restart the corresponding plugins.
- Check the described pods output if one of the pods is in a state which is not running or is in $\frac{1}{2}$. This will give you information on why the pod is not coming up.

- Check the disk space of all nodes for disk pressure.
- To perform a pod delete on all the name spaces, execute the following command:

```
kubectrl delete pods -n avx $(kubectrl get pods -n avx | awk '{print $1}' | tail -n +1) --force
```

- To check the kubelet status, execute the following command:

```
systemctl status kubelet -l
```

- To check the kubernetes node status, execute the following command:

```
kubernetes get nodes
```

- To check the docker status, execute the following command:

```
systemctl status docker -l
```

Error while downloading certificates

When there is an error while downloading a certificate with a pfx or p12 extension and the error states unable to communicate to vault, please check the status of the vault pods and try restarting the vault and consul related pods using the following command:

```
kubectrl delete pods -n avx $(kubectrl get pods -n avx | grep vault | awk '{print $1}' | tail -n +1)
```

```
kubectrl delete pods -n avx $(kubectrl get pods -n avx | grep consul | awk '{print $1}' | tail -n +1)
```

Chapter 10: Glossary

This section describes common terms and their abbreviations used in this guide.

Term/Abbreviation Meaning/Expansion