appviewX

# Admin Guide

**Version: 2019.3.0**

# Copyright AppViewX, Inc.



### External Reference Links

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

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

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



### Contact Information


AppViewX, Inc.

222 Broadway, FL 19

New York, NY 10038

Email: info@appviewx.com

Web: www.appviewx.com

# Contents

# Preface

## Revision History

| Revision | Description | Date |
|---|---|---|
| 1.0 | Initial release of document for Release 2019.3.0 | July 2019 |

## About this Guide

The guide introduces and provides step-by-step instructions for configuring and managing the administrative capabilities of the product.

## Audience

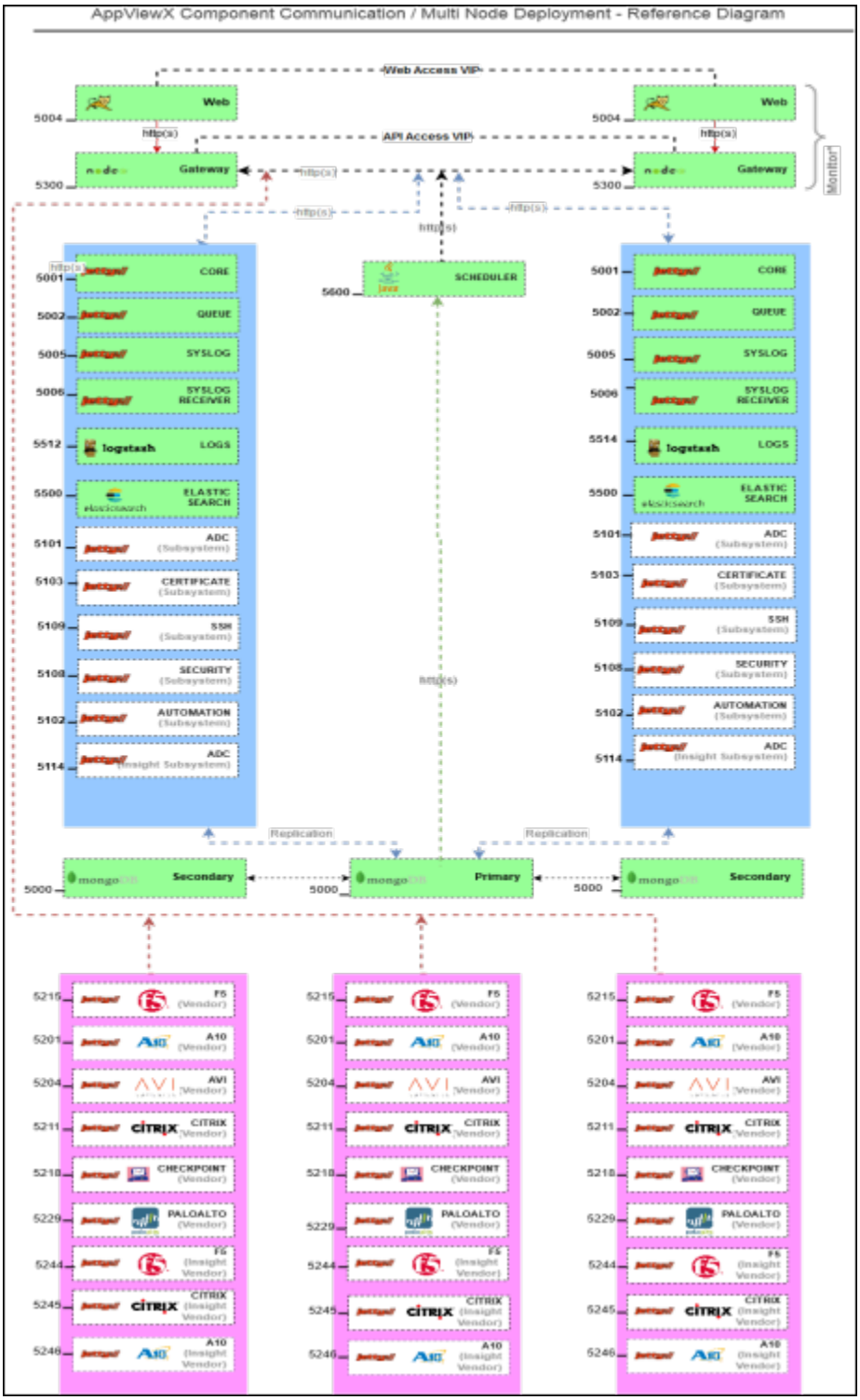
This guide is intended for PKI Security, DevOps, and Application Teams.

## Text Conventions

The following text conventions are used in this document:

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

# Chapter 1: AppViewX Stack Plugins Communication Flow (Multi-Node)



# Chapter 2: AppViewX Single Node Installation (Evaluation OVA)



## Prerequisites

Before installing AppViewX, make sure you have downloaded the release package in **.ova** format from https://release.appviewx.comto either the **Downloads** folder or to the **Desktop** in your local environment.

## Platform Requirements

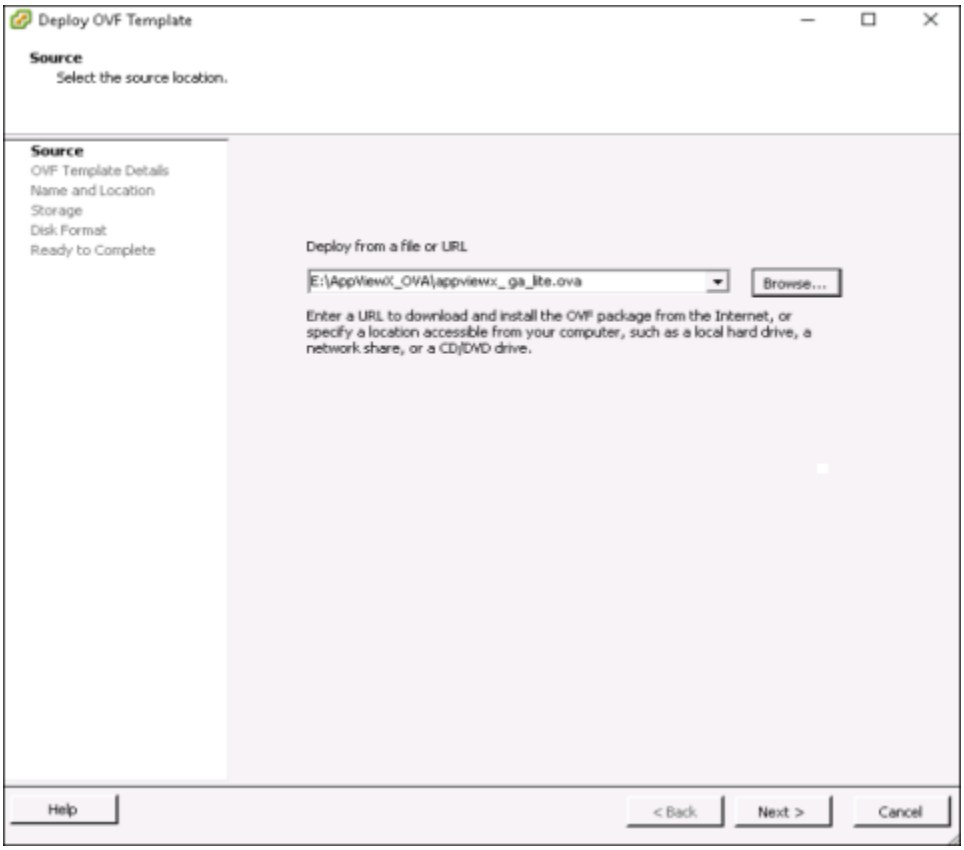
| OS Platforms Supported | Versions | CPU | RAM | HDD |
|---|---|---|---|---|
| VM Server, VMware ESXI | 5.5 and above | 8v | 16 GB | 200 GB |

## Install AppViewX

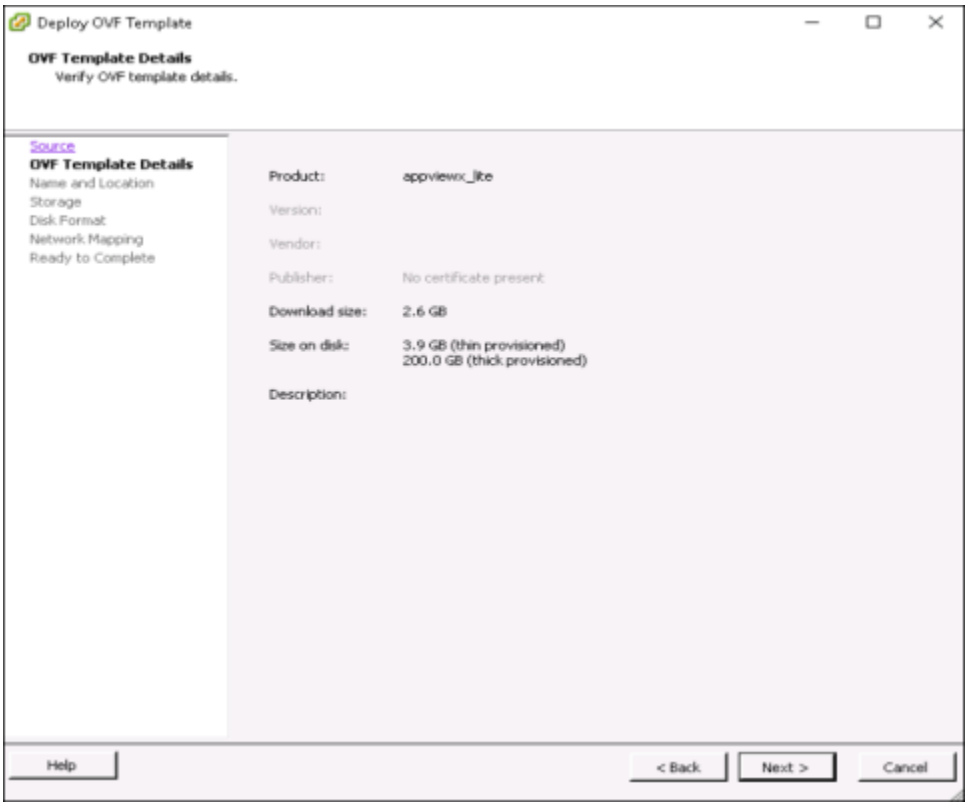
To install the EVAL version of OVA, which is always a single node installation:

**Note:** A maximum of 200 GB will be allocated.

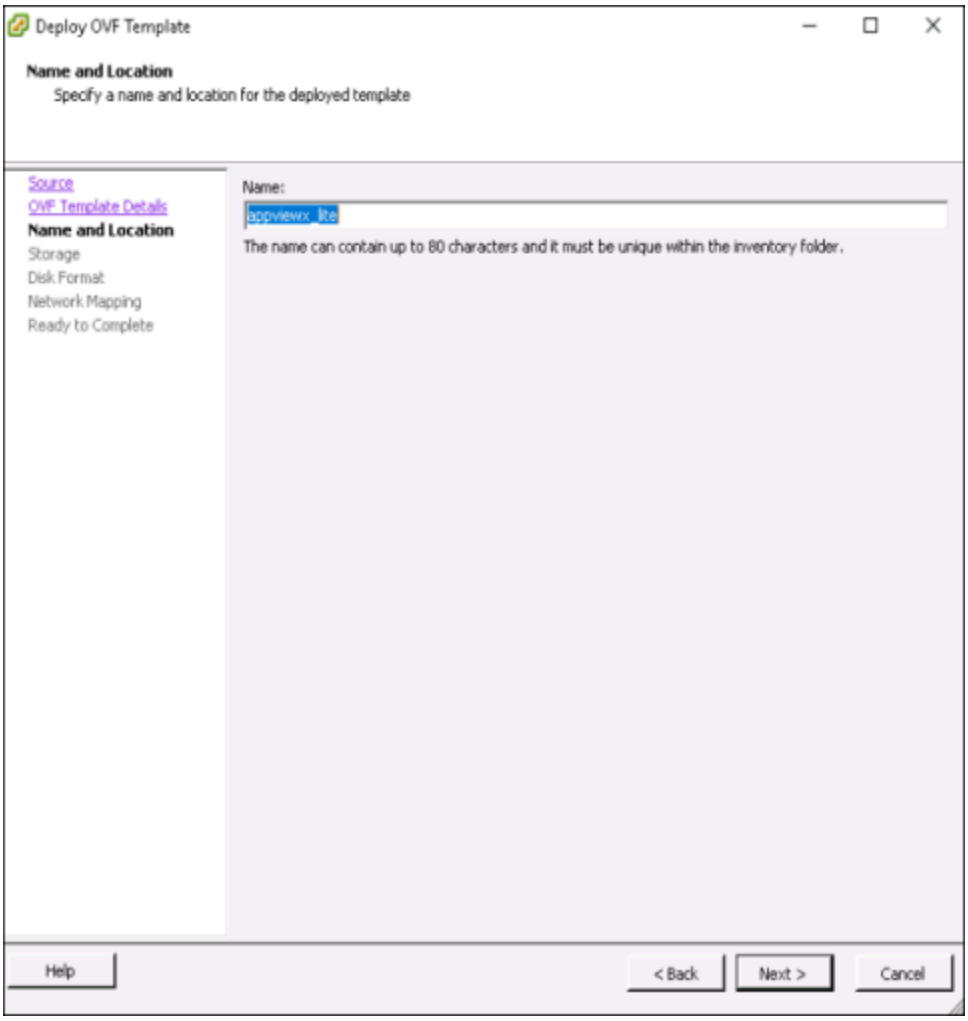
1. Log in to the Vsphere Client.
2. Go to **File>> Select File>> Deploy OVF Template**.
3. On the Source screen, browse to the OVA file location.



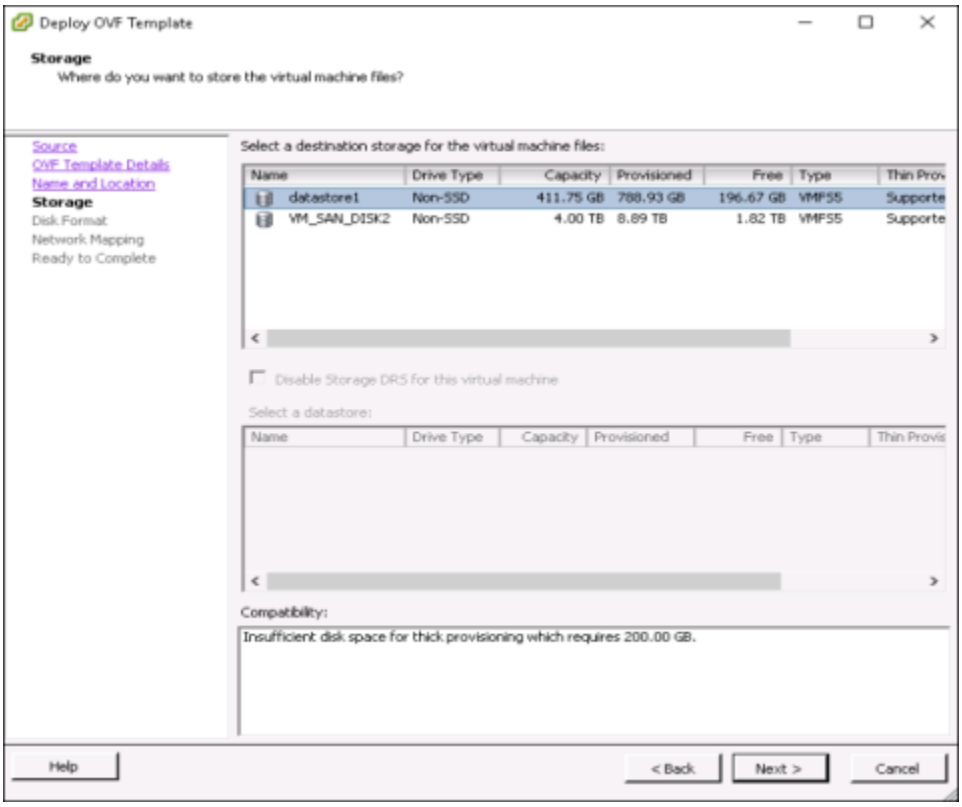

4. Click **Next**.
5. On the OVF Template Details screen, verify the template details to ensure that you have the right OVA.



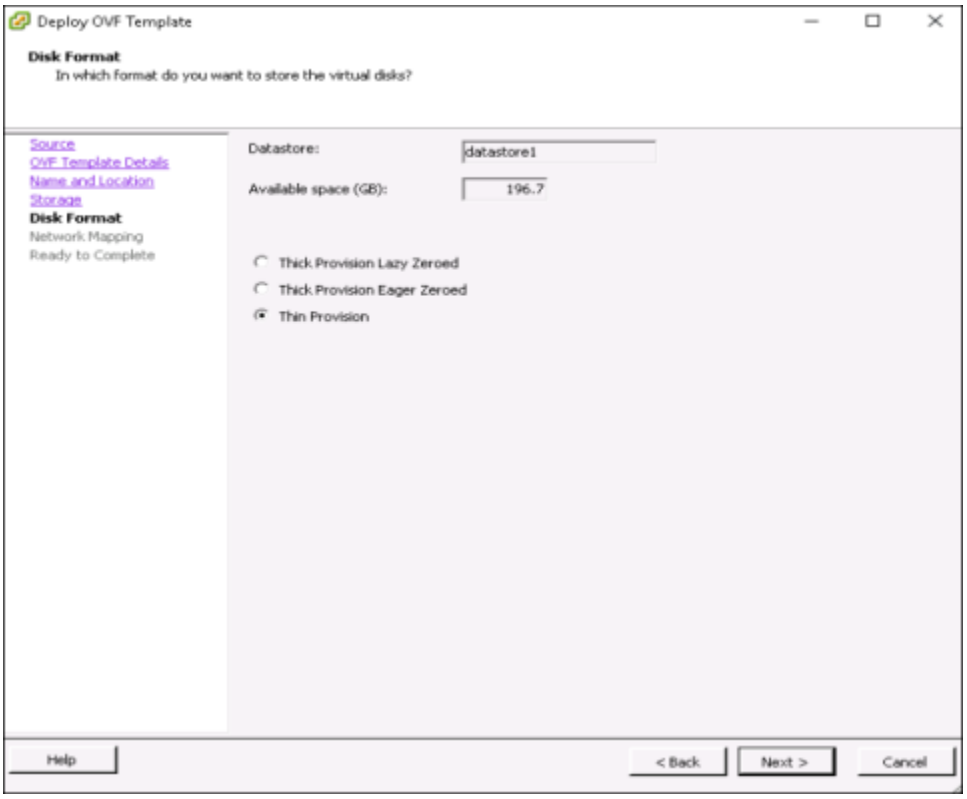

6. Click **Next**.
7. (Optional) On the **Name and Location** screen, modify the server name to be displayed.




8. Click **Next**.
9. On the **Storage** screen, select a storage location.

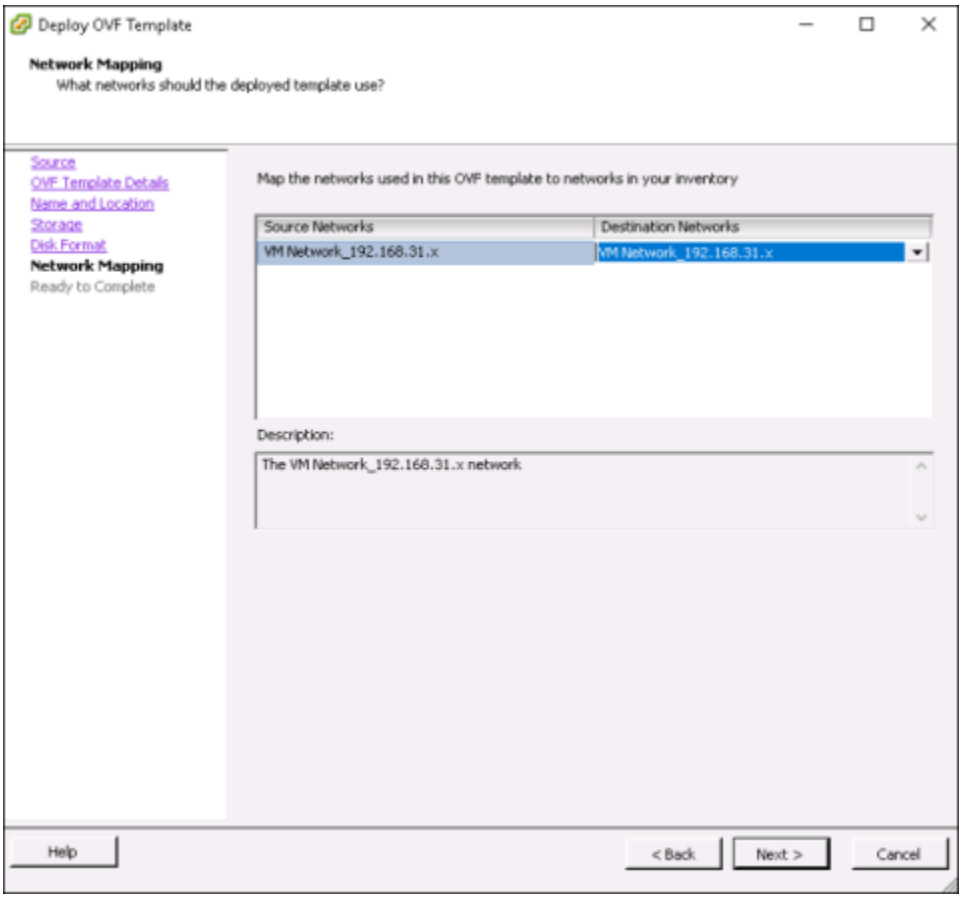



10. Click **Next**.
11. On the **Disk Format** screen, select a disk type.

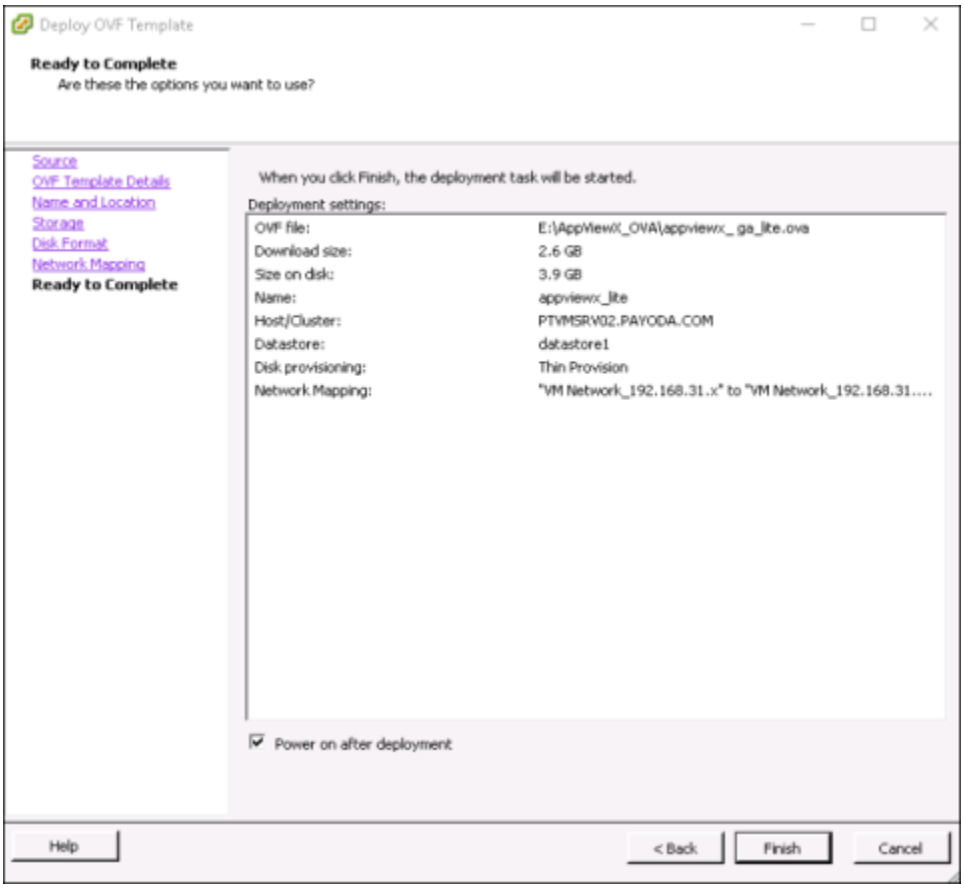



**Note:** A maximum of 200 GB will be allocated.

12. Click **Next**.
13. On the **Network Mapping** screen, choose a network adapter.




14. Click **Next**.
15. On the **Ready to Complete** screen, verify all details, then start the deployment by clicking the **Finish** button.

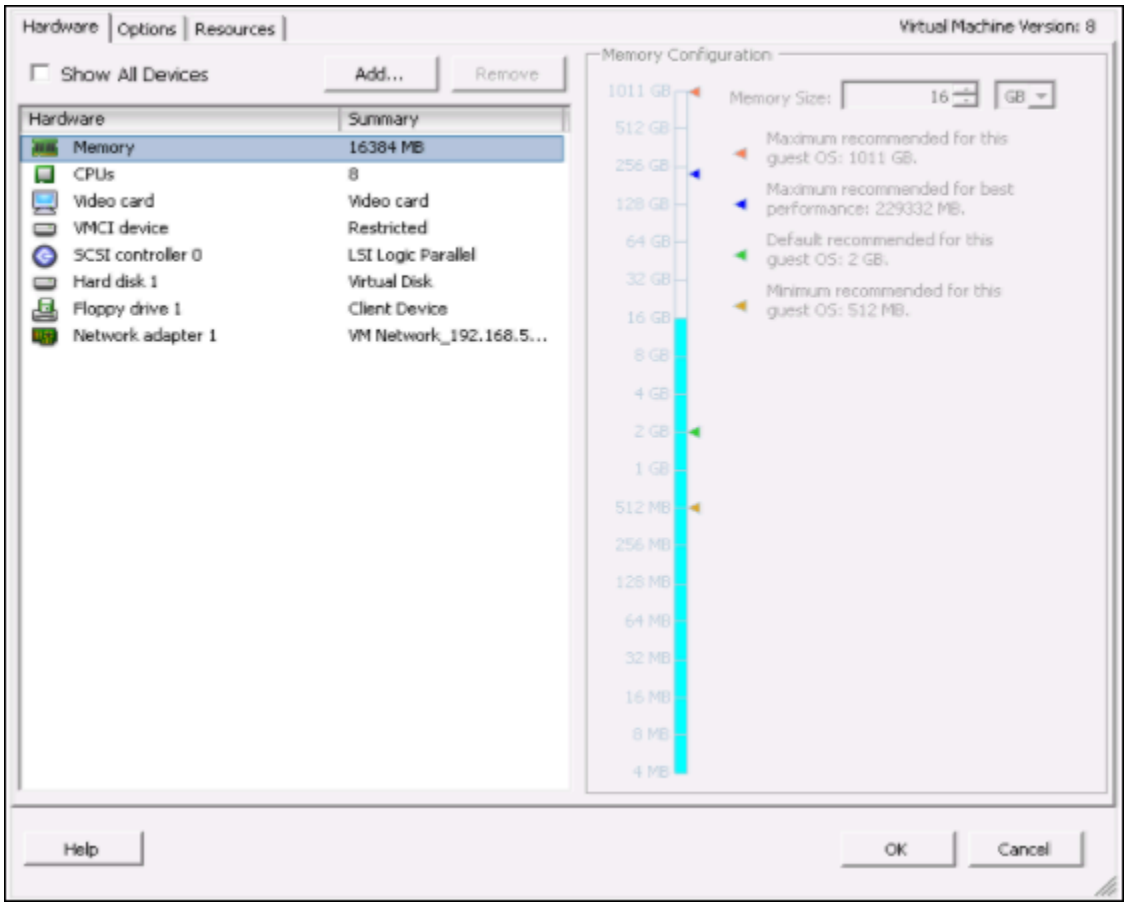



16. (Optional) After the OVA wizard finishes deploying the OVA, you can modify the CPUs and the memory allocation of the server by doing the following:

    a. Right-click the server name.

    b. Select the **Edit Settings** option.

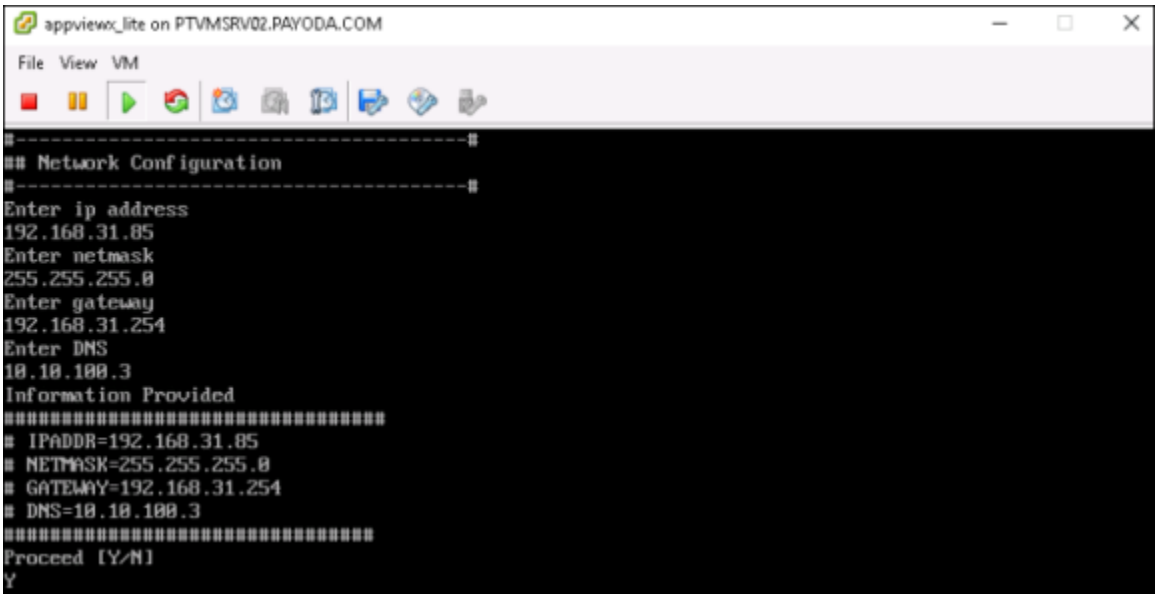
    c. Make changes to the fields within the **Hardware** tab.



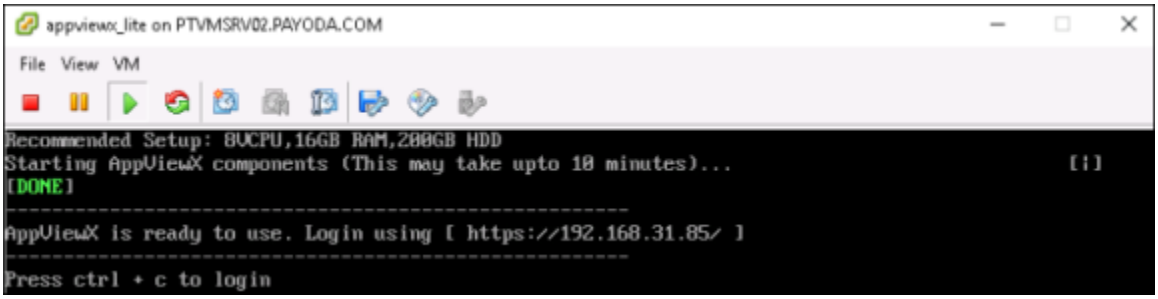

The minimum hardware requirements are as follows:

| OS Platforms Supported | Version | VM or OVA Support | Packages | CPU | RAM |
|---|---|---|---|---|---|
| CentOS | 7.x | Yes | NC, NMAP-NCAT, NMAP, CURL, SYSSTAT, TCPDUMP, RSYNC, NETSTAT, ZIP, UNZIP, OPENSSL, BZIP, OPENLDAP-CLIENTS | 8VCPU | 16 GB |

17. When the deployment wizard finishes, the AppViewX user interactive provisioning console opens within the vSphere Client. You will use this console to set up your basic network configuration.
18. Type `Y` on the console, screen to proceed with the network configuration.




19. After the basic network configuration process finishes, the installation starts automatically.




20. After the installation is complete, you can access the application by opening a browser on the host machine and entering **https://<ip>**.
    - The port 443 is routed to 5004.
    - The lite OVA comes with the normal Linux shell by default. When the custom shell is enabled, it restricts the shell and provides limited functionality to the user.
21. To enable/disable the shell in an AppViewX custom shell, complete the following steps:

    a. Log in as an AppViewX user and run the following command to enable a shell: `enableshell`
    When you log in again, the shell will be enabled with limited functionality.

    b. Run the following command to disable a shell: `disableshell`.

    c. Enter the AppViewX user account password.
    When you log in again, the shell will be disabled.

# Chapter 3: AppViewX Multi-Node Installation (Production OVA)



## Prerequisites

Before installing AppViewX, make sure you have downloaded the release package in **.OVA** format from https://release.appviewx.comto the **Downloads** folder or the **Desktop** in your local environment.

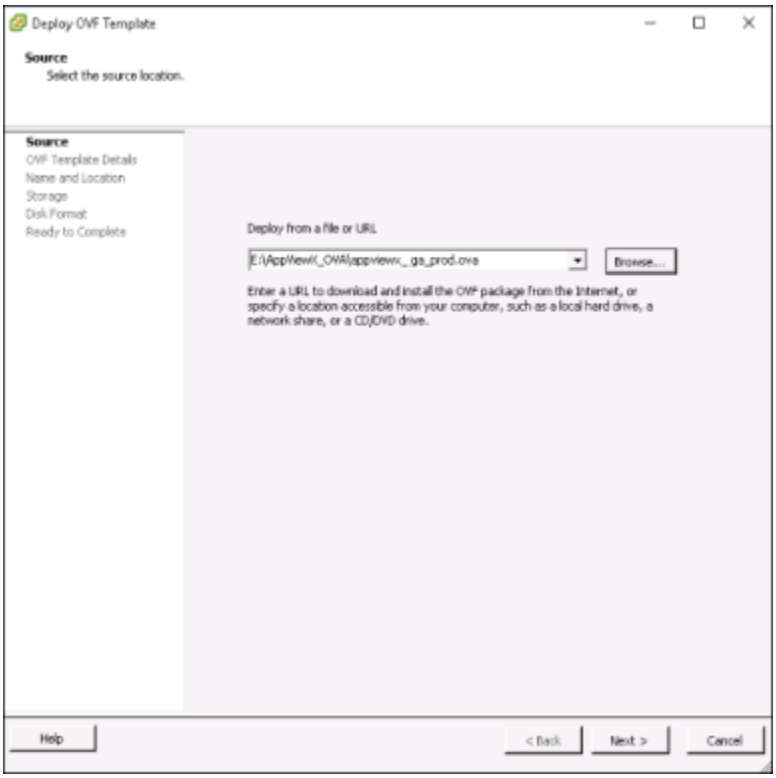
## Platform Requirements

| OS Platforms Supported | Versions | CPU | RAM | HDD |
|---|---|---|---|---|
| VM Server, VMware ESXI | 5.5 and above | 8v | 32 GB | 1 TB |

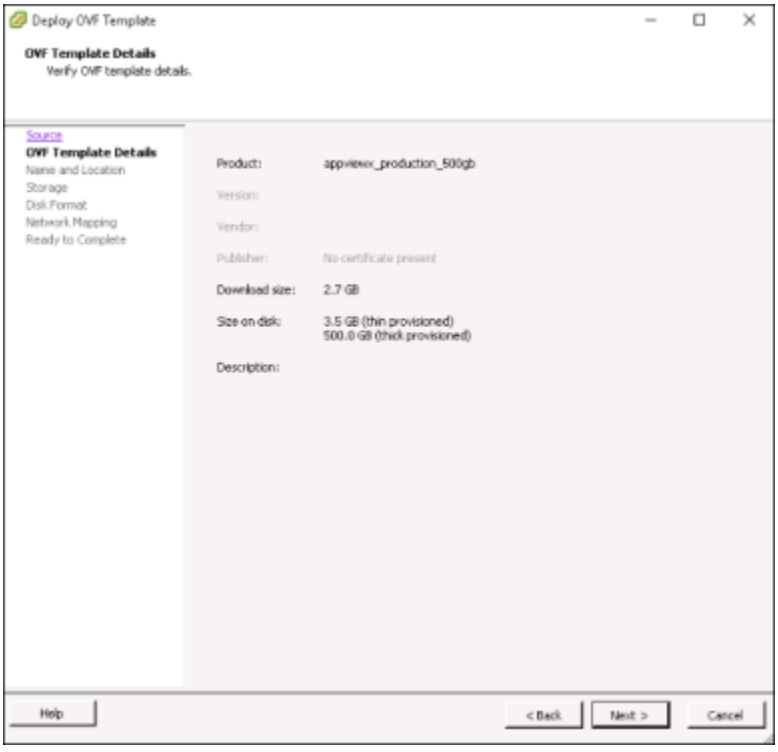
## Installing AppViewX

To install the production version of OVA:

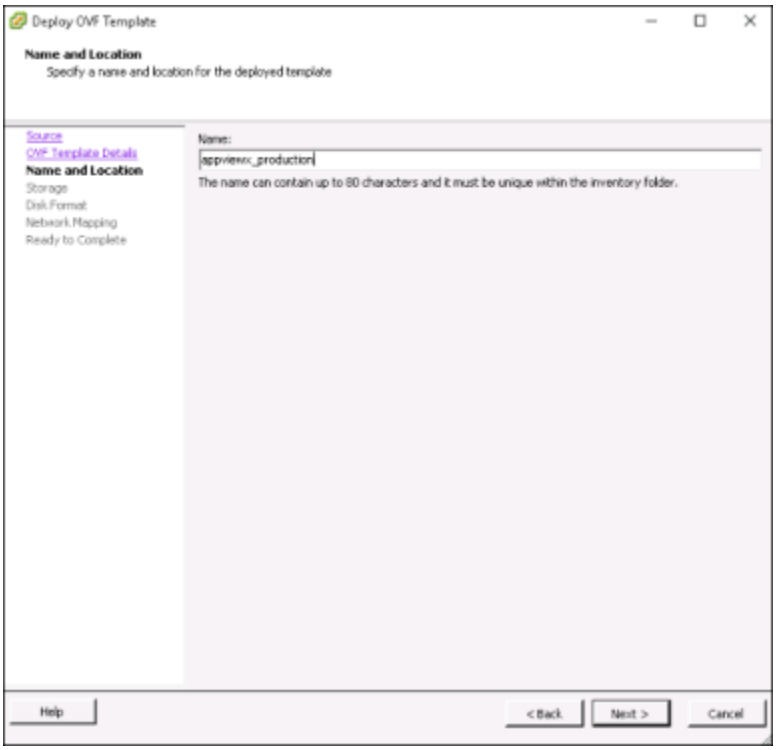
1. Log in to the **vSphere** Client.
2. Within vSphere, go to **File>> Select File>> Deploy OVF Template**
3. On the Source screen, browse to the OVA file location.



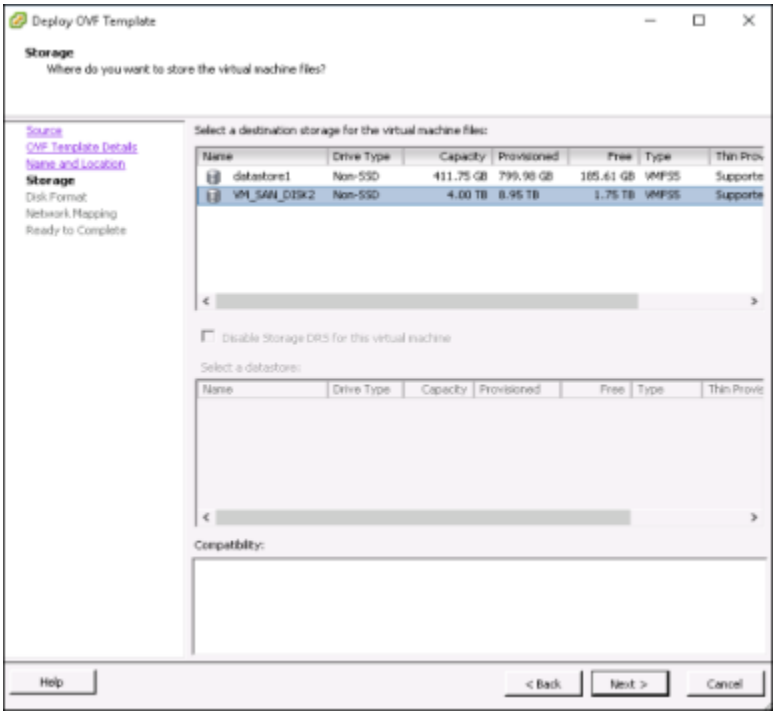

4. Click **Next**.
5. On the **OVF Template Details** screen, verify the template details to ensure that you have the right OVA.



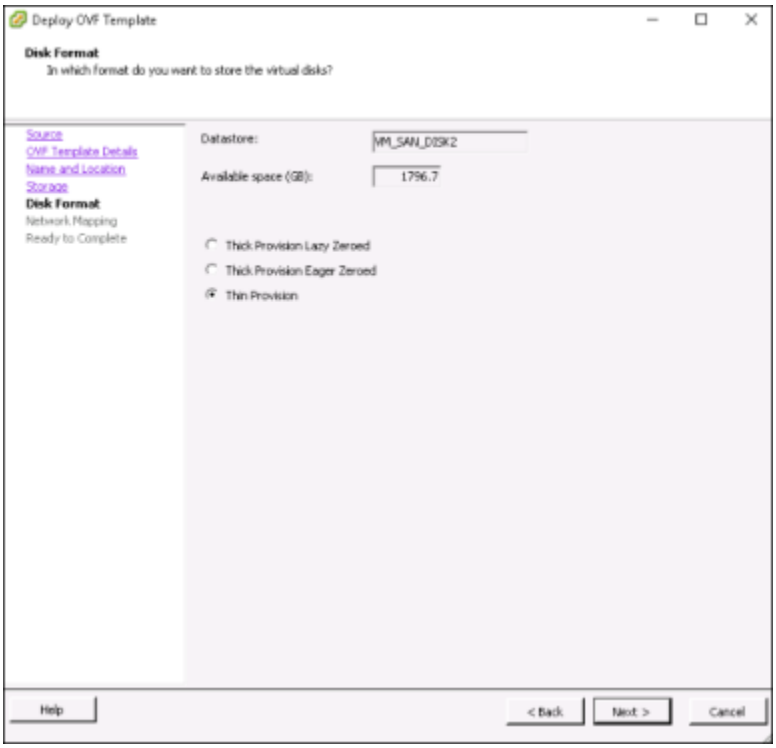
6. Click **Next**.
7. (Optional) On the **Name and Location** screen, modify the server name to be displayed.




8. Click **Next** .
9. On the **Storage** screen, select a storage location.

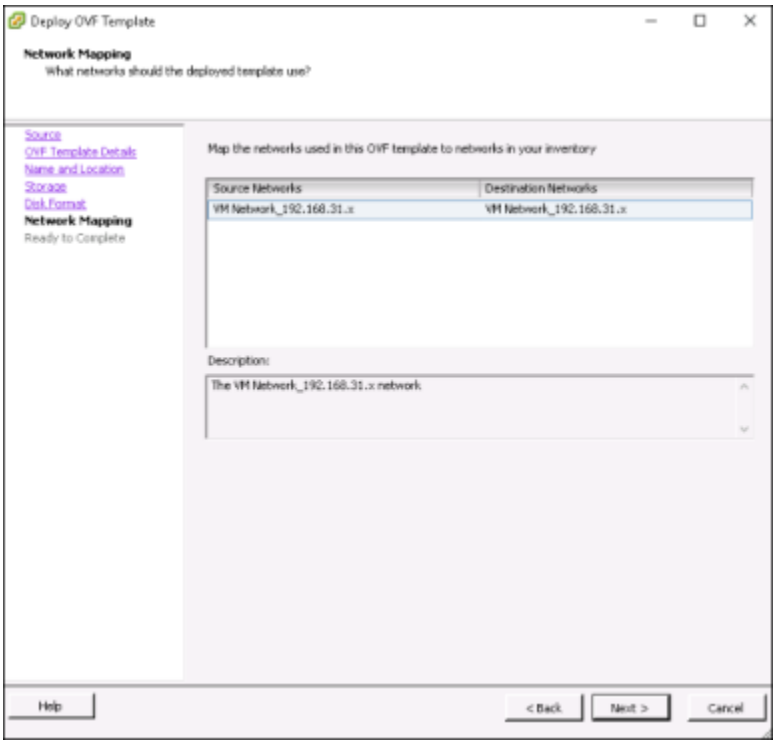


10. Click **Next** .
11. On the **Disk Format** screen, select a disk type.

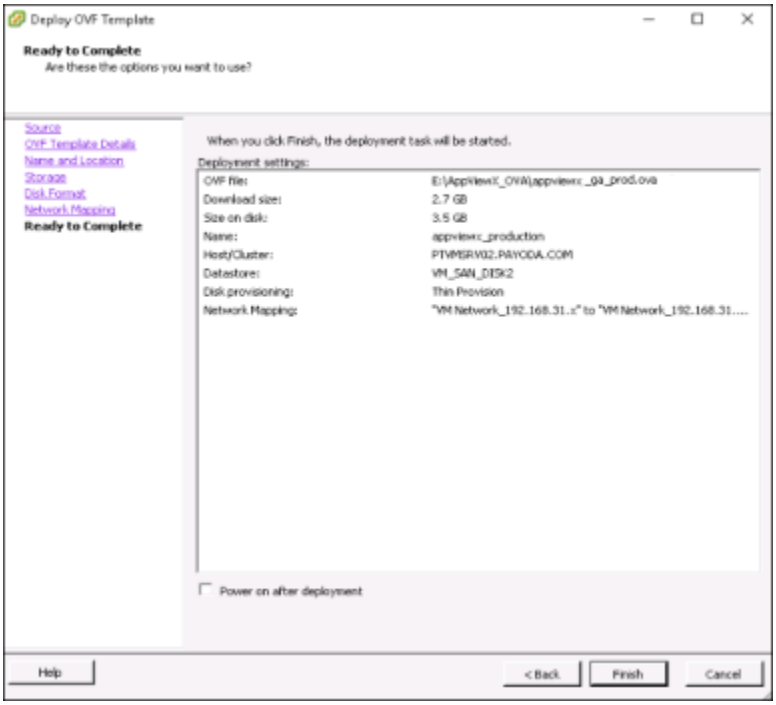



**Note:** A maximum of 1 TB will be allocated.

12. Click **Next** .
13. On the **Network Mapping** screen, choose a network adapter.

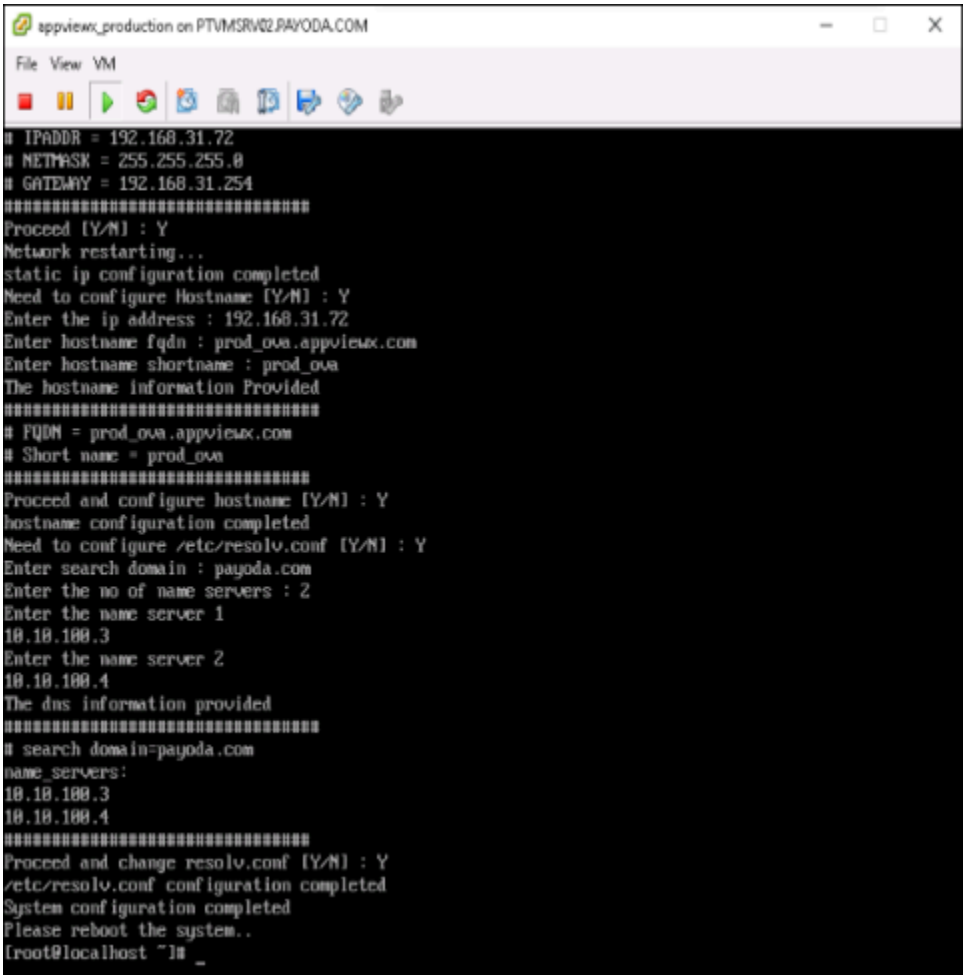



14. Click **Next** .
15. On the **Ready to Complete** screen, verify all details, then start the deployment by clicking the **Finish** button.

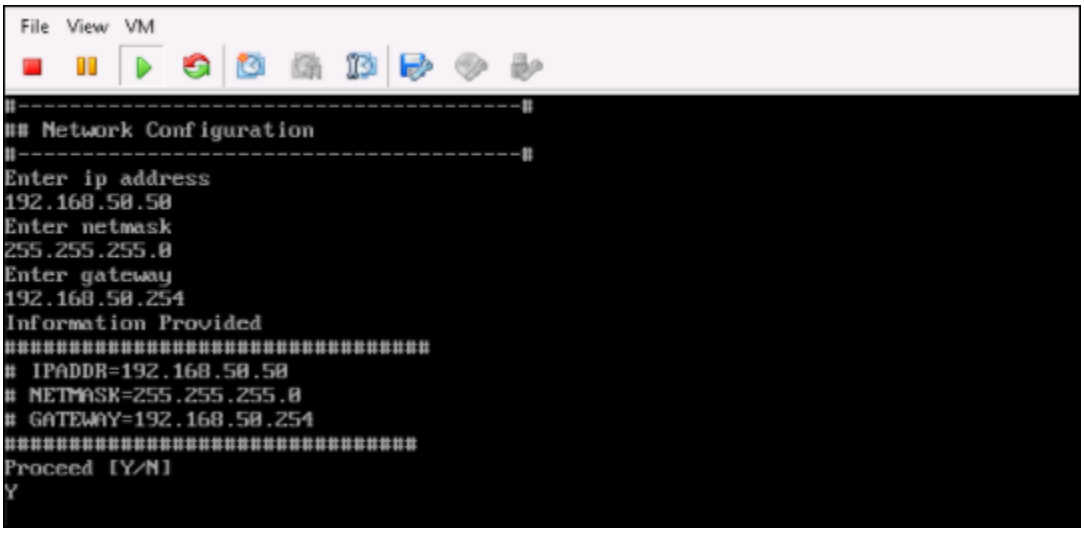


16. After the deployment is complete, access the root folder as a root user by entering the following command: `$ cd /root`.
17. Run the **network_conf_setup.py** script from the root folder by executing the following command: `$ python network_conf_setup.py`.

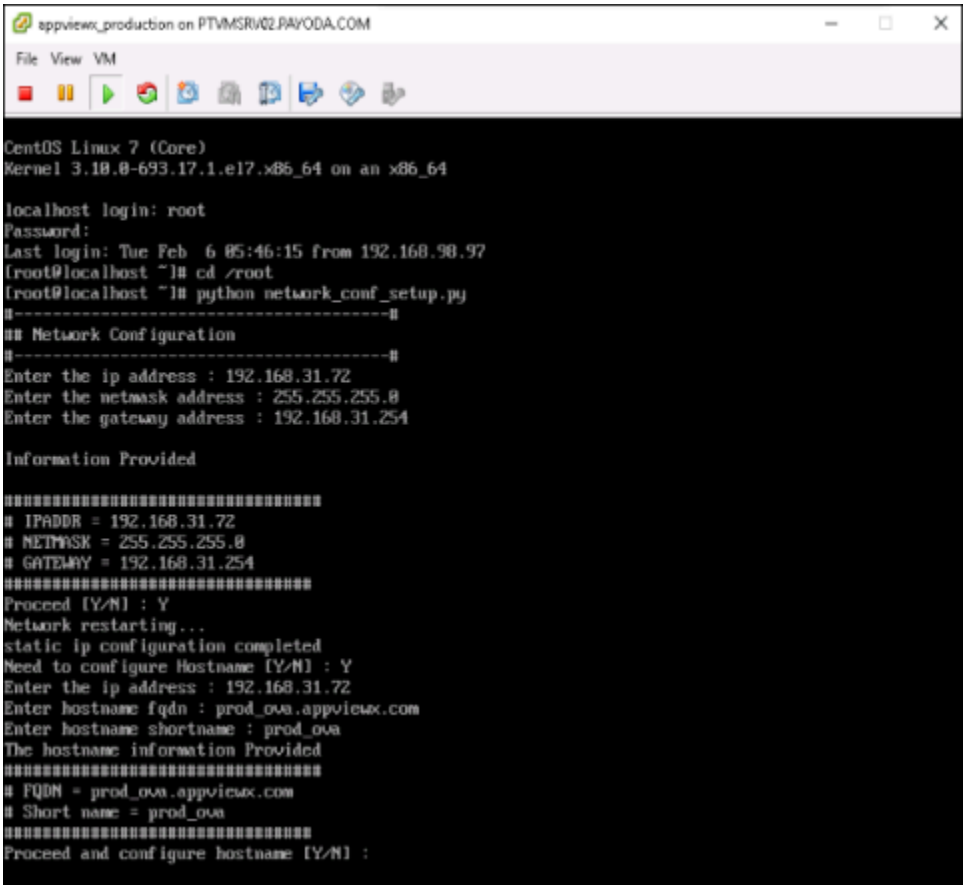
    The console starts and the user is prompted to enter the network configuration for the node.



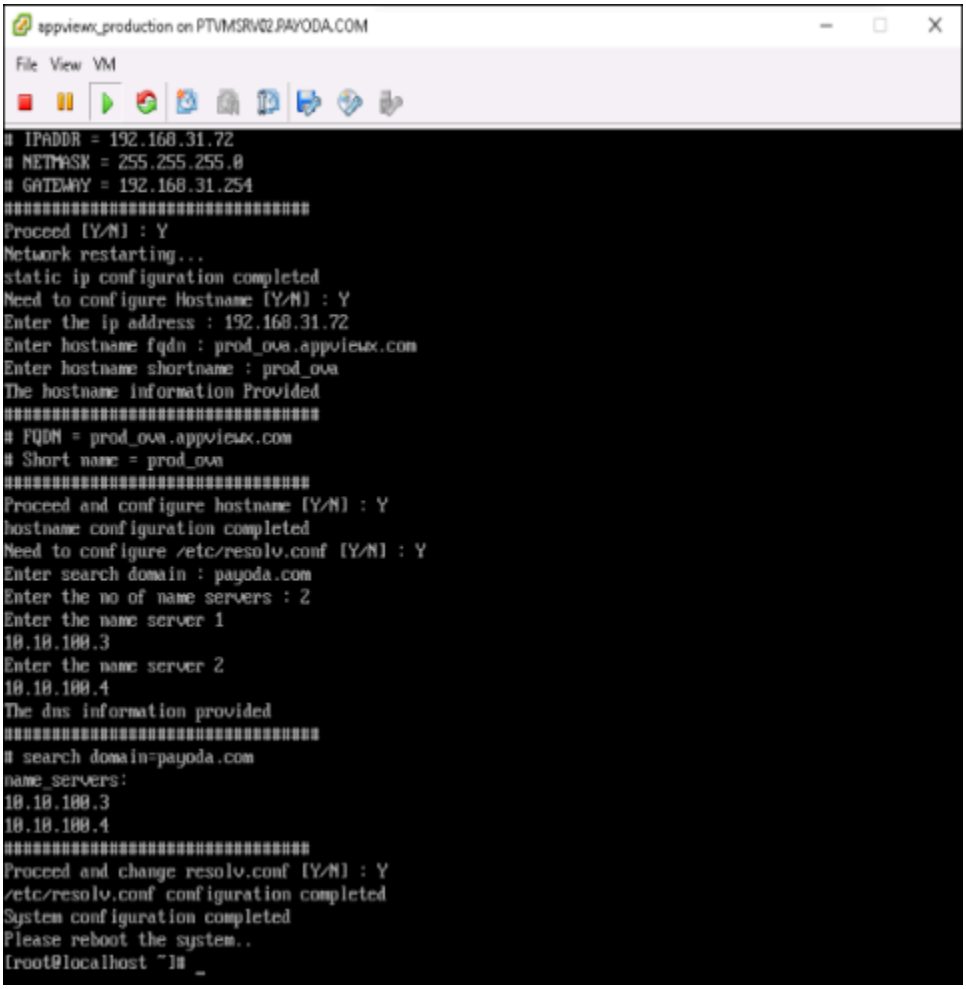

18. After entering the IP address, Netmask, and Gateway information for the node, the following prompt appears: `Proceed [Y/N]`. **Type** `Y` to proceed.




19. When the prompt **Need to configure Hostname** [Y/N] appears, type `Y` to proceed and then provide the IP address, desired hostname, and the short name for the hostname.
20. At the prompt, **Proceed and configure hostname**: [Y/N], type `Y` to proceed with the node configuration.



21. Repeat steps 2-21 for the other nodes where AppViewX is to be deployed.




22. After the OVA and network configuration steps are completed across all nodes, SSH into any one of the nodes as an AppViewX user to start the manual installation process. `ssh appviewx@<node ip address>`
23. Refer to the **AppViewX Native Installation: Multi-Node** section of this guide to complete the installation.

# Chapter 4: KVM Based Installation

**Note:** The port 443 is routed to 5004.

1. Log in to the **OpenStack** server.
2. Go to **Computer** >> **Images**.
3. Click on the **Create Image** option.
4. In the pop-up screen displayed, enter the following mandatory fields:
   - Name
   - Select the Image source from the dropdown list: Image File or Image Location
   - Click the Choose file button, then navigate to the location of the Image file, and click to select it.
   - Select QCOW2 - QEMU Emulator from the format dropdown list.
   - Click the Create Image button.
5. After the image is created, click the dropdown menu under the Actions column and select Create Volume.
6. In the pop-up screen displayed, enter the following mandatory fields:
   - Name
   - Set the Volume Size for the image that you created to 200 GB.
   - Click the Create Volume button.
7. After the volume is created, go to Compute > Volumes and then, click the Launch as Instance under the Actions column.
8. After the instance is created, go to Compute > Instances and then, click on Console under the Actions column.
9. Set up your basic network configuration.
10. Type **Y** on the console screen to proceed with the network configuration.
    After the installation is complete, you can access the application by opening a browser on the host machine and entering **https://<ip>**.

    **Note:** The port 443 is routed to 5004.

# Chapter 5: Deployment of a VHD Image

To install a VHD image:

1. Open the **Hyper-V** manager.
2. Create a new Virtual Machine (VM).
3. On the wizard that opens, click **Next** and enter a name and location of the image.
4. On the **Assign Memory** screen, enter a startup memory space that needs to be allocated for the virtual machine.
5. Click **Next**.
6. On the **Configure Networking** screen, select a network adapter to use a virtual switch from the **Connection** dropdown list.
7. Click **Next**.
8. On the **Connect Virtual Hard Disk** screen, enter a location to attach an existing virtual hard disk, VHD, or VHDX format.
9. Click **Finish**.
10. Right-click the VM that you created and go to Settings.
11. Modify the number of virtual processors that you want to use and click **Apply**.
12. When the deployment wizard finishes, use the console to set up your basic network configuration.

# Chapter 6: AppViewX Native Installation: Single Node



## Prerequisites

Before installing AppViewX, make sure the following are true:

- The release package in .tar.gz format has been downloaded from https://release.appviewx.com to **<user_home_directory>**.
- The AppViewX addons in .tar.gz format have been downloaded from https://release.appviewx.com to **<user_home_directory>**.
- The open files [`ulimit -n`] and the maximum processes [`ulimit -u`] should be **65535** with port (SSH) **22** opened.
- If the elastic search is enabled, make sure the following are true:
  - The open files [`ulimit -n`] should be **65536**.
  - The virtual map count [`sysctl vm.max_map_count`] should be 262144.
- Locale language should be **en_US.utf8**.
- **CLIENT_CERT** should not be enabled when a proxy is available.

> **Note:** Run the following commands as a root user to set the virtual map count to 262144:
>
> - `echo "vm.max_map_count=262144" >> /etc/sysctl.conf`
> - `sysctl --system`

## OS Requirements

| OS Platforms Supported | Versions | VM or OVA Support | Packages | CPU | RAM | HDD |
|---|---|---|---|---|---|---|
| CentOS | 6.x, 7.x | Yes | NC, NMAP-NCAT, NMAP, CURL, SYSSTAT, TCPDUMP, RSYNC, NETSTAT, ZIP, UNZIP, | 8v | 32 GB | 1 TB |

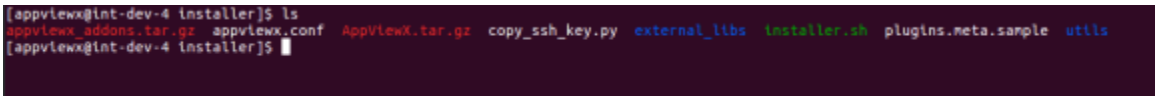
| OS Platforms Supported | Versions | VM or OVA Support | Packages | CPU | RAM | HDD |
|---|---|---|---|---|---|---|
| | | | OPENSSL, BIND-UTILS, FONT CONFIG (version 2.13.0), GIT, GIT-LFS, RSNAPSHOT. | | | |
| RHEL | 6.x, 7.x | No | NC, NMAP-NCAT, NMAP, CURL, SYSSTAT, TCPDUMP, RSYNC, NETSTAT, ZIP, UNZIP, OPENSSL, BIND-UTILS, FONT CONFIG (version 2.13.0), GIT, GIT-LFS, RSNAPSHOT. | 8v | 32 GB | 1 TB |

- OpenLDAP-CLIENTS to support the SSH subsystem.
- BZIP2 to support elastic search.

**Note:** The SSH+ plugin is not supported in CentOS 6.x and Redhat 6.x.

## Installing AppViewX

1. Go to the user home directory: `$ cd <user_home_directory>`
2. Untar the **.tar.gz** installer package by entering the following command, which extracts the installer directory in the current location: `$ tar -xvf appviewx_2020.1.0_BXX_RXXXX.tar.gz`
3. Navigate to the **installer** directory by executing the following command: `$ cd installer`
4. Move appviewx_addons.tar.gz to the installer directory using the following command: `mv ../ appviewx_addons.tar.gz.`
5. Once moved, complete the steps provided in this section.
6. Type the command `ls` to verify the existence of the following files: **utils/migration_validator.js**, **utils/adc_collection_copy.js**, **AppViewX.tar.gz**, **installer.sh**, **copy_ssh_key.py**, **plugins.meta.sample**, and **appviewx.conf**.




7. To update the plugins configuration to be installed, enter the following command: `$ vi appviewx.conf.`
8. After editing the **conf** file press the **Esc** key, then type `:wq` to save and quit.
9. Specify the plugins to install under **ENABLED_PLUGINS** and specify the **IP: PORT** details for each plugin to be installed.

```
[PLUGINS]

##------------------------------------------------
##
## ENABLED_PLUGINS will contain a list of all the plugins that are to be enabled.
## For the plugins mentioned in ENABLED_PLUGINS field, hosts details need to be configured
##
##------------------------------------------------

ENABLED_PLUGINS=avx_platform_core,avx_platform_queue,avx_subsystems,avx_vendors,avx_vendor_cert_network_discovery,avx_vendor_cert_scep_agent

#####################################################################
#
#  AVAILABLE PLATFORM PLUGINS = avx_platform_core, avx_platform_queue,
#                               avx_platform_syslog, avx_platform_syslog_receiver
#
#####################################################################

avx_platform_core=localhost:5001
avx_platform_queue=localhost:5002
avx_platform_syslog=localhost:5005
avx_platform_syslog_receiver=localhost:5006
avx_commons=localhost:5007
```

10. When you are done editing the fields, press the **Esc** key, then type `:wq` to save and quit the file.
11. If the latest patches are available for particular versions, then the latest patch file can be downloaded and applied. Download the patch file from the release portal and extract it in the installer directory.

```
[appviewx@int-dev-8 installer]$ ls
AppViewX_2019.1.1_Latest_Plugins          AppViewX.tar.gz      utils
AppViewX_2019.1.1_Latest_Plugins.tar.gz   installer.sh
appviewx.conf                             plugins.meta.sample
```

12. Enter the following command: `$./installer.sh` .

    Upon executing the installer script, the user will be prompted with the following options:

    • Fresh Installation

    • Migration

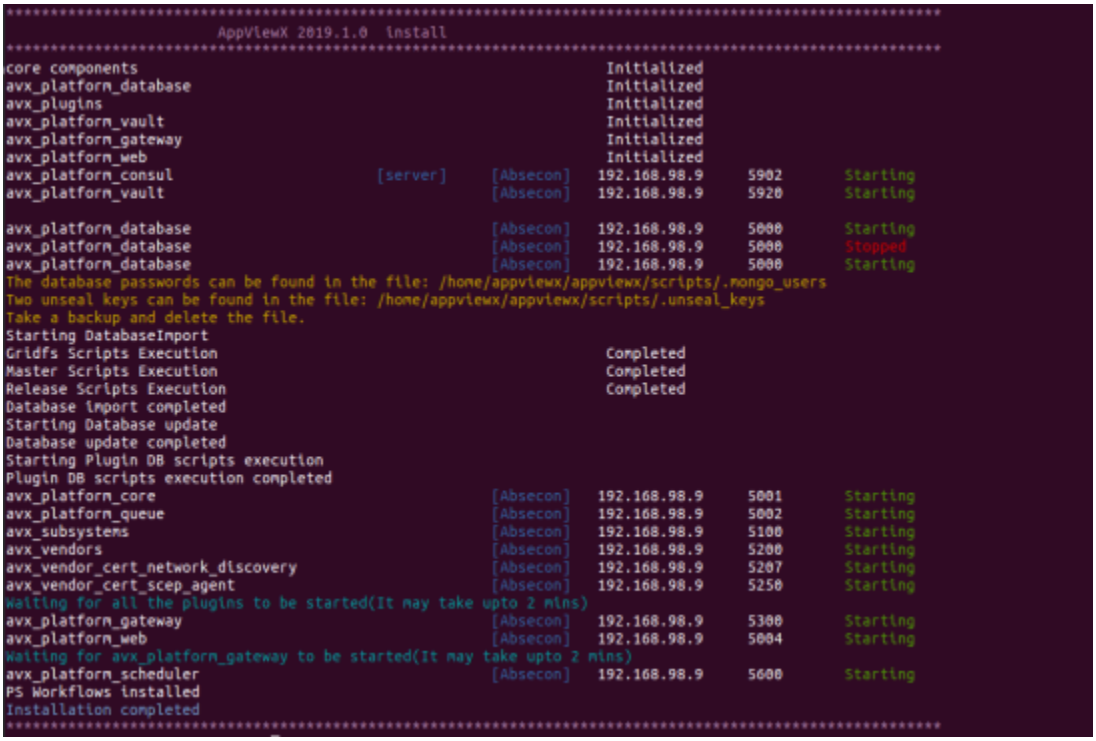
13. Select the option **Fresh Installation** to trigger installation.

    After the install setup script is run, the installation starts extracting AppViewX dependent libraries and packages.

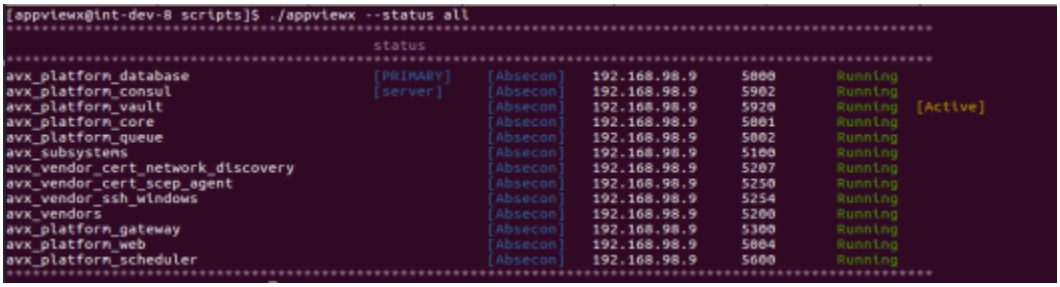
14. By default, AppViewX will be installed at `/home/appviewx/appviewx`. To install it in a custom location, enter the installation path when prompted, and then hit **Enter** to proceed with the installation.

```
[appviewx@int-dev-8 installer]$ ./installer.sh
Preparing installation workspace. This may take a few minutes. Please wait.
Preparing libraries


Choose one of the following installation options
1. Fresh Installation
2. Migration-CLI
3. Migration-GUI

Enter your choice : 1
Extracting Python
************************** Fresh installation started **************************
AppViewX installation path (Default: (/home/appviewx/appviewx):
Starting pre-requisite check
WARNING!! Following system specifications are not upto mark for 192.168.98.9.
        Prerequisite            :    Recommended            :    Availablity
        number of cpus          :    8                      :    4
        total ram memory        :    32GB                   :    24
        free disk space         :    200GB                  :    29
        Locale-lang             :                           :    en_US.UTF-8
        openldap-clients                                    :    Not Installed
Copying the installer
Copying appviewx addons and external libs directory
Extracting the installer
New patches not found
Copying appviewx.conf
```

After the prerequisite check finishes, the installation starts.




15. Enter the following command to ensure that all components are up and running: `$ cd <avx_installed_directory>/scripts && ./appviewx --status all`.




16. Access the application by opening a browser on the host machine and entering **https://<web_ip>:<web_port>**.

    During installation, random passwords generated for the MongoDB users will be available in the temporary file location: **<appviewx_dir>/scripts/.mongo_users** (hidden file). This file location will be displayed during the installation process.

**Note:** Random passwords will be generated only for the users whose passwords were not changed.

**Note:** If the proxy has been configured for the CA setting, please execute the below command to update the changes: `appviewx --restart plugins avx_vendors`

# Chapter 7: AppViewX Native Installation: Multi-Node



## Prerequisites

Before installing AppViewX, make sure the following are true:

- The release package in .tar.gz format has been downloaded from https://release.appviewx.com to **<user_home_directory>**.
- The AppViewX addons in .tar.gz format have been downloaded from https://release.appviewx.com to **<user_home_directory>**.
- The open files [`ulimit -n`] and the maximum processes [`ulimit -u`] should be **65535** with port (SSH) **22** opened.
- If the elastic search is enabled, make sure the following are true:
  - The open files [`ulimit -n`] should be **65536**.
  - The virtual map count [`sysctl vm.max_map_count`] should be **262144**.
- The system time difference between the cluster nodes does not exceed 10 seconds.
- The component ports are accessible across the cluster.
- An SSH key is shared across the cluster nodes.
- Locale language should be **en_US.utf8**.
- **CLIENT_CERT** should not be enabled when a proxy is available.

**Note:** Run the following commands as a root user to set the virtual map count to 262144:

- `echo "vm.max_map_count=262144" >> /etc/sysctl.conf`
- `sysctl --system`

## OS Requirements

| OS Platforms Supported | Versions | VM or OVA Support | Packages | CPU | RAM | HDD |
|---|---|---|---|---|---|---|
| CentOS | 6.x, 7.x | Yes | NC, NMAP-NCAT, NMAP, CURL, SYSSTAT, TCPDUMP, RSYNC, NETSTAT, ZIP, UNZIP, OPENSSL, BIND-UTILS, FONT CONFIG (version 2.13.0), GIT, GIT-LFS, RSNAPSHOT. | 8v | 32 GB | 1 TB |
| RHEL | 6.x, 7.x | No | NC, NMAP-NCAT, NMAP, CURL, SYSSTAT, TCPDUMP, RSYNC, NETSTAT, ZIP, UNZIP, OPENSSL, BIND-UTILS, FONT CONFIG (version 2.13.0), GIT, GIT-LFS, RSNAPSHOT. | 8v | 32 GB | 1 TB |

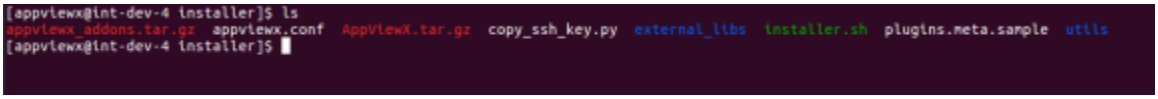
**Note:** The following packages are required only in specific cases:

- OpenLDAP-CLIENTS to support the SSH subsystem.
- BZIP2 to support elastic search.

**Note:** The SSH+ plugin is not supported in CentOS 6.x and Redhat 6.x.

## Installing AppViewX

1. Go to the user home directory: `$ cd <user_home_directory>`
2. Untar the **.tar.gz** installer package by entering the following command, which extracts the **installer** directory to the current location: `$ tar -xvf appviewx_2020.1.0_BXX_RXXXX.tar.gz`
3. Go to the **installer** directory by executing the following command: `$ cd installer`.
4. Move **appviewx_addons.tar.gz** to the **installer** directory using the following command: `mv ../appviewx_addons.tar.gz`.
5. Once moved, complete the steps provided in this section.
6. Type the command `ls` to verify the existence of the following files: **AppViewX.tar.gz**, **installer.sh**, **copy_ssh_key.py**, **plugins.meta.sample**, and **appviewx.conf**.



**Note:** Step 5 to 9 are optional. The procedure to enable password-free communication between the nodes has been incorporated as part of installation process. You can either pass the key during installation process or complete the following steps to pass the key before installation.

7. For password-free communication between the nodes on which AppViewX is going to be installed, enter the following command: `$ vi copy_ssh_key.py` .
8. Press the **Insert** key on your keyboard to activate **Edit** mode.
9. Edit the **NODE_DETAILS** field with the node IPs where AppViewX is going to be installed.

```
# Following Values are to be modified by the user
################################################
MULTINODE = 'TRUE'
NODE_DETAILS = ['192.168.31.32', '192.168.31.33', '192.168.31.34']
USER_DETAILS = ['appviewx']
PORT_DETAILS = [22]
################################################
```

10. When you are done editing the fields, press the **Esc** key, then type `:wq` to save and quit the file.
11. Execute the following command, which allows the keys to be passed between the servers: `python copy_ssh_key.py`.

```
[appviewx@avx-31-32 installer]$ python copy_ssh_key.py
password for appviewx@192.168.31.32 :
password for appviewx@192.168.31.33 :
password for appviewx@192.168.31.34 :

 Success. RSA keys are copied to all the servers
```

12. To update the multinode setup to be installed, enter the following command: `$ vi appviewx.conf`.
13. After editing the conf file press the **Esc** key, then type `:wq` to save and quit.
14. Update the multi-node flag as **TRUE** and update the node details: `SSH HOST = username@<ipaddress : installation path>`
15. By default, the SSH PORT is set to **22**. Update this field if the SSH PORT has been configured with a different value.

```
MULTINODE=True
SSH_HOSTS=appviewx@192.168.31.32:/home/appviewx/appviewx,appviewx@192.168.31.33:/home/appviewx/appviewx,appviewx@192.168.31.34:/home/appviewx/appviewx
SSH_PORT=22
```

**Note:** Ensure that the installation path are same across all the nodes.

16. Update the data center details of the node.

```
DEFAULT_DATACENTER = Absecon
DATACENTER = Absecon:192.168.31.32 && Virginia:192.168.31.33,192.168.31.34
```

17. Update the node details with the port where the database should be running.

```
[MONGODB]
HOSTS=192.168.31.32:5000,192.168.31.33:5000,192.168.31.34:5000
ARBITER_HOSTS=
```

18. Set the **ENABLE_VAULT** field in the **VAULT** section to **TRUE** and update the following fields to install the consul and vault components:

> **Note:** The Consul must have been configured in an odd number of nodes and the Vault must have been configured in the node where a consul is configured. Also, it is not mandatory to configure a vault in the database nodes. It is recommended to configure the vault in two nodes and the consul in three nodes for a multinode.

- CONSUL_CLUSTER
- CONSUL_CLIENT_PORT
- HOSTS
- VAULT_CLUSTER_PORT
- LOG_LEVEL

```
[VAULT]
ENABLE_VAULT = True

CONSUL_CLUSTER = localhost:5902

##------------------------------------------------
## The consul client port only needs to be configured in case of a multinode setup
##------------------------------------------------
CONSUL_CLIENT_PORT = 5912

##------------------------------------------------
## VAULT_CLIENTS should be preferrably configured on gateway nodes
##------------------------------------------------
HOSTS = localhost:5920
VAULT_CLUSTER_PORT = 5921

##------------------------------------------------
## Possible values: info / debug / trace
##------------------------------------------------
LOG_LEVEL = Info
```

19. Update the node details with the port where the gateway should be running. To enable the external VIP configured for Gateway, change `GATEWAY_VIP_ENABLED = TRUE`.
20. Configure the external **VIP IP: PORT** detail at **APPVIEX_GATEWAY_VIP** and set the `APPVIEWX_GATEWAY_VIP_HTTPS = TRUE`, if SSL is configured in VIP.

```
[GATEWAY]
HOSTS=192.168.31.32:5300,192.168.31.33:5300
APPVIEWX_GATEWAY_KEY=f000ca01

##-----------------------------------------------
## To enable secure connection in gateway, set APPVIEWX_GATEWAY_HTTPS as TRUE
##
##-----------------------------------------------

APPVIEWX_GATEWAY_HTTPS=True

##-----------------------------------------------
## To enable VIP for gateway, set GATEWAY_VIP_ENABLED as TRUE
##
##-----------------------------------------------

GATEWAY_VIP_ENABLED=False
APPVIEWX_GATEWAY_VIP=localhost:5300
APPVIEWX_GATEWAY_VIP_HTTPS=False
```

21. Update the node details with the port where the web should be running. To enable the external VIP configured for the Web, change **WEB_VIP_ENABLED = TRUE.**
22. Configure the external **VIP IP: PORT** detail at **APPVIEX_WEB_VIP** and **APPVIEWX_WEB_VIP_HTTPS=TRUE**, if SSL is configured in VIP.

```
[WEB]
HOSTS=192.168.31.32:5004,192.168.31.33:5004

##-----------------------------------------------
## To enable secure connection in web, set APPVIEWX_WEB_HTTPS as TRUE
##
##-----------------------------------------------

APPVIEWX_WEB_HTTPS=True

##-----------------------------------------------
## To enable VIP for web, set WEB_VIP_ENABLED as TRUE
##
##-----------------------------------------------
WEB_VIP_ENABLED=False
APPVIEWX_WEB_VIP=localhost:5004
APPVIEWX_WEB_VIP_HTTPS=False
```

23. Specify the plugins under **ENABLED_PLUGINS** and specify the **<IP: PORT>** details for each plugin to be installed.

```
[PLUGINS]

##------------------------------------------
##
## ENABLED_PLUGINS will contain a list of all the plugins that are to be enabled.
## For the plugins mentioned in ENABLED_PLUGINS field, hosts details need to be configured
##
##------------------------------------------

ENABLED_PLUGINS=avx_platform_queue,avx_platform_core,avx_subsystems,avx_vendors

#################################################################
#
#  AVAILABLE PLATFORM PLUGINS = avx_platform_core, avx_platform_insight, avx_platform_queue,
#                               avx_platform_syslog, avx_platform_syslog_receiver
#
#################################################################

avx_platform_core = 192.168.31.32:5001,192.168.31.33:5001
avx_platform_queue = 192.168.31.32:5002,192.168.31.33:5002
avx_platform_insight=192.168.31.32:5003,192.168.31.34:5003
avx_platform_syslog=192.168.31.32:5005,192.168.31.33:5005
avx_platform_syslog_receiver=192.168.31.33:5006,192.168.31.34:5006
```

24. When you are done editing the fields, press the **Esc** key, then type `:wq` to save and quit the file.
25. If the latest patches are available for particular versions, then the latest patch file can be downloaded and applied. Download the patch file from the release portal and extract it in the **installer** directory.

```
[appviewx@int-dev-8 installer]$ ls
AppViewX_2019.1.1_Latest_Plugins          AppViewX.tar.gz      utils
AppViewX_2019.1.1_Latest_Plugins.tar.gz   installer.sh
appviewx.conf                             plugins.meta.sample
```

26. Enter the following command: `$./installer.sh`.

    Upon executing the installer script, the user will be prompted with the following options:

    - Fresh Installation
    - Migration

27. Select the option **Fresh Installation** to trigger installation.

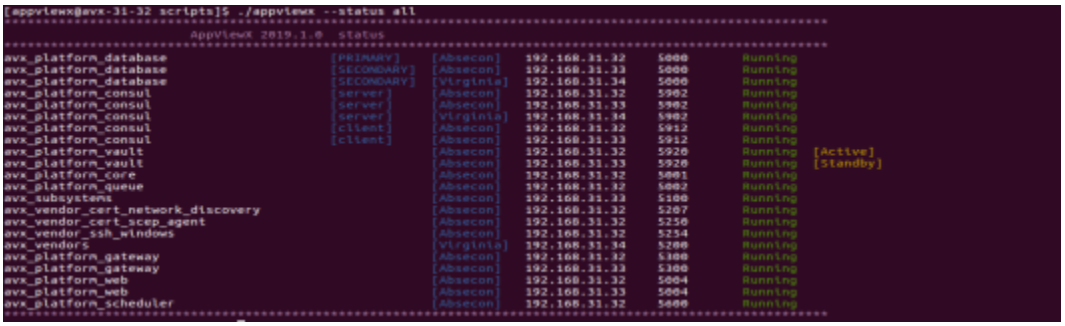
    After the installation setup script is run, the installation starts extracting AppViewX dependent libraries and packages.

```
[appviewx@int-dev-8 installer]$ ./installer.sh
Preparing installation workspace. This may take a few minutes. Please wait.
Preparing libraries


Choose one of the following installation options
1. Fresh Installation
2. Migration-CLI
3. Migration-GUI

Enter your choice : 1
Extracting Python
************************** Fresh installation started **************************
AppViewX installation path (Default: (/home/appviewx/appviewx):
Starting pre-requisite check
WARNING!! Following system specifications are not upto mark for 192.168.98.9.
        Prerequisite            :    Recommended            :    Availablity
        number of cpus          :    8                      :    4
        total ram memory        :    32GB                   :    24
        free disk space         :    200GB                  :    29
        Locale-lang             :                           :    en_US.UTF-8
        openldap-clients                                    :    Not Installed
Copying the installer
Copying appviewx addons and external libs directory
Extracting the installer
New patches not found
Copying appviewx.conf
```

28. After the prerequisite check finishes, the installation starts.

```
*****************************************************************************************************
                    AppViewX 2019.1.0  install
*****************************************************************************************************
core components                                         Initialized
avx_platform_database                                   Initialized
avx_plugins                                             Initialized
avx_platform_vault                                      Initialized
avx_platform_gateway                                    Initialized
avx_platform_web                                        Initialized
avx_platform_consul                 [server]    [Absecon]   192.168.31.32   6000      Starting
avx_platform_consul                 [server]    [Absecon]   192.168.31.33   6000      Starting
avx_platform_consul                 [server]    [Virginia]  192.168.31.34   6000      Starting
avx_platform_consul                 [client]    [Absecon]   192.168.31.32   5912      Starting
avx_platform_consul                 [client]    [Absecon]   192.168.31.33   5912      Starting
avx_platform_vault                              [Absecon]   192.168.31.32   7000      Starting
avx_platform_vault                              [Absecon]   192.168.31.33   7000      Starting

avx_platform_database                           [Absecon]   192.168.31.32   5000      Starting
avx_platform_database                           [Absecon]   192.168.31.33   5000      Starting
avx_platform_database                           [Virginia]  192.168.31.34   5000      Starting

avx_platform_database                           [Absecon]   192.168.31.32   5000      Stopped
avx_platform_database                           [Absecon]   192.168.31.33   5000      Stopped
avx_platform_database                           [Virginia]  192.168.31.34   5000      Stopped
avx_platform_database                           [Absecon]   192.168.31.32   5000      Starting
avx_platform_database                           [Absecon]   192.168.31.33   5000      Starting
avx_platform_database                           [Virginia]  192.168.31.34   5000      Starting
The database passwords can be found in the file: /home/appviewx/appviewx/scripts/.mongo_users
Two unseal keys can be found in the file: /home/appviewx/appviewx/scripts/.unseal_keys
Take a backup and delete the file.
Starting DatabaseImport
Gridfs Scripts Execution                                Completed
Master Scripts Execution                                Completed
Release Scripts Execution                               Completed
Database import completed
Starting Database update
Database update completed
```

29. Use the following command to ensure all the components are up and running: `$ cd <avx_installed_directory>/scripts && ./appviewx --status all`.




30. Access the application by opening a browser on the host machine and entering **https://<web_ip>:<web_port>**.
    During installation, random passwords generated for the MongoDB users will be available in the temporary file location: **<appviewx_dir>/scripts/.mongo_users** (hidden file). This file location will be displayed during the installation process.

    **Note:** Random passwords will be generated only for the users whose passwords were not changed.

    **Note:** If the proxy has been configured for the CA settings, please execute the below command to update the changes `appviewx --restart plugins avx_vendors`.

# Chapter 8: AppViewX Native Upgrade: Single Node



## Prerequisites

Before installing AppViewX, make sure the following are true:

- The release package in .tar.gz format has been downloaded from https://release.appviewx.com to **<user_home_directory>**.
- The AppViewX addons in .tar.gz format have been downloaded from https://release.appviewx.com to **<user_home_directory>**.
- The open files [`ulimit -n`] and the maximum processes [`ulimit -u`] should be **65535** with port (SSH) **22** opened.
- If the elastic search is enabled, make sure the following are true:
  - The open files [`ulimit -n`] should be **65536**.
  - The virtual map count [`sysctl vm.max_map_count`] should be **262144**.
- Locale language should be **en_US.utf8**.
- **CLIENT_CERT** should not be enabled when a proxy is available.

**Note:** Run the following commands as a root user to set the virtual map count to 262144:

- `echo "vm.max_map_count=262144" >> /etc/sysctl.conf`
- `sysctl --system`

## OS Requirements

| OS Platforms Supported | Versions | VM or OVA Support | Packages | CPU | RAM | HDD |
|---|---|---|---|---|---|---|
| CentOS | 6.x, 7.x | Yes | NC, NMAP-NCAT, NMAP, CURL, SYSSTAT, TCPDUMP, RSYNC, NETSTAT, ZIP, UNZIP, OPENSSL, BIND-UTILS, FONT CONFIG (version 2.13.0), GIT, GIT-LFS, RSNAPSHOT. | 8v | 32 GB | 1 TB |
| RHEL | 6.x, 7.x | No | NC, NMAP-NCAT, NMAP, CURL, SYSSTAT, TCPDUMP, RSYNC, NETSTAT, ZIP, UNZIP, OPENSSL, BIND-UTILS, FONT CONFIG (version 2.13.0), GIT, GIT-LFS, RSNAPSHOT. | 8v | 32 GB | 1 TB |

**Note:** To configure RSNAPSHOT download the .config file from this link. In that config file, you can configure the value of snapshot_root (specify a location with free space) and backup (location where AppViewX is installed).

**Note:** The following packages are required only in specific cases:

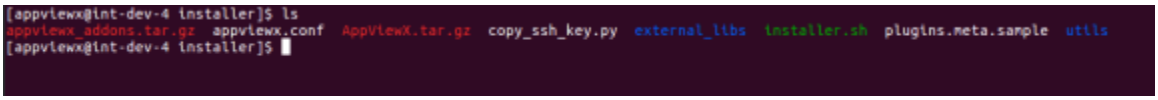
- OpenLDAP-CLIENTS to support the SSH subsystem.
- BZIP2 to support elastic search.

**Note:** The SSH+ plugin is not supported in CentOS 6.x and Redhat 6.x.

## Upgrading AppViewX

1. Go to the user home directory and untar the .tar.gz upgrade package by entering the following command: `$ cd <user_home_directory> && tar -xvf appviewx_2019.4.0_BXX_RXXXX.tar.gz`
2. Go to the **installer** directory by executing the following command: `$ cd installer`
3. Move **appviewx_addons.tar.gz** to the **installer** directory using the following command: `mv ../appviewx_addons.tar.gz`.
4. Once moved, complete the steps provided in this section.

5. Type the command `ls` to verify the existence of the following files: **AppViewX.tar.gz**, **installer.sh**, **copy_ssh_key.py**, **plugins.meta.sample**, and **appviewx.conf**.




> **Note:** To proceed with the new configuration, you need to follow steps 4 to 13. If not, it will proceed with the old configuration setup. If you are upgrading from a version below 12.0., you have to follow the new configuration steps. Old configuration steps will not work for versions below 12.0.

6. Go to the plugins.meta.sample file using the following command: `$ vi plugins.meta.sample`
7. Update the ports where platform plugins are going to be installed.

```
avx_platform_core=localhost:5001
avx_platform_queue=localhost:5002
avx_platform_insight=localhost:5003
avx_platform_syslog=localhost:5005
avx_platform_syslog_receiver=localhost:5006
```

8. Update the ports where subsystem plugins are going to be installed.

```
avx_subsystems = localhost:5100
```

9. Update the ports where vendor plugins are going to be installed.

```
avx_vendors = localhost:5200
```

10. Update the external_certificate upgrade configuration to **True** if an external certificate is already in use. Update the external_certificate upgrade configuration to **False** if a self-signed certificate is already in use.

```
#########################################################################
##
## External_Certificate should be either True or False
##
#########################################################################

[SSL]

External_Certificate=True
```

11. Set the **ENABLE_VAULT** field in the **VAULT** section to **TRUE** and update the following fields to install the consul and vault components:

> **Note:** If the existing AppViewX version is 12.4.0 or later, make sure you update the VAULT section same as the existing version in the `appviewx.conf` file.

- CONSUL_CLUSTER
- CONSUL_CLIENT_PORT
- HOSTS
- VAULT_CLUSTER_PORT
- LOG_LEVEL

```
[VAULT]
ENABLE_VAULT = True

CONSUL_CLUSTER = localhost:5902

##------------------------------------------------
## The consul client port only needs to be configured in case of a multinode setup
##------------------------------------------------
CONSUL_CLIENT_PORT = 5912

##------------------------------------------------
## VAULT_CLIENTS should be preferrably configured on gateway nodes
##------------------------------------------------
HOSTS = localhost:5920
VAULT_CLUSTER_PORT = 5921

##------------------------------------------------
## Possible values: info / debug / trace
##------------------------------------------------
LOG_LEVEL = Info
```

12. Update the custom SSH port configuration if required where default communication is through port 22.

```
#####################################################################
##
## SSH_PORT should have the value of the AppViewX environment SSH Port
## In case of multinode setup, it should be comma separated values.
##
## Eg:
##       SSH_PORT=22         (Singlenode)
##       SSH_PORT=22,22,22   (Multinode)
##
#####################################################################

[ENVIRONMENT]
SSH_PORT=22
```

    For more details on how to configure syslog, refer to the Enable SYSLOGS Reception from Devices section of this guide.

13. When you are done editing the fields, press the **Esc** key, then type `:wq` to save and quit the file.
14. Rename the file to **plugin.meta**.
15. If the latest patches are available for particular versions, then the latest patch file can be downloaded and applied. Download the patch file from the release portal and extract it in the **installer** directory.

```
[appviewx@int-dev-8 installer]$ ls
AppViewX_2019.1.1_Latest_Plugins          AppViewX.tar.gz      utils
AppViewX_2019.1.1_Latest_Plugins.tar.gz   installer.sh
appviewx.conf                             plugins.meta.sample
```

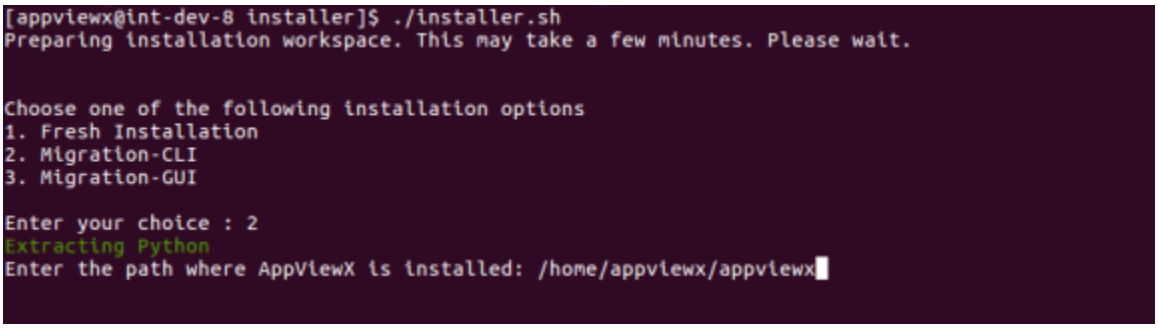
16. Trigger the migration process with the following command: `$ ./installer.sh`

    Upon executing the install script, the user will be prompted with the following options:

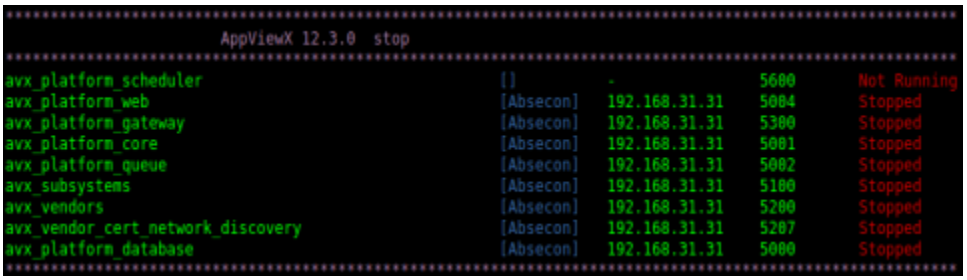
- Fresh Installation
- Migration CLI
- Migration GUI

## Migrating via CLI

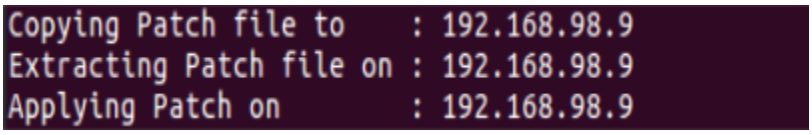
1. Select the option **Migration CLI** to trigger the migration. If you want to upgrade the application in the GUI, please refer to point No. 24.
2. Specify the path where the earlier version is currently running and press **Enter** on your keyboard.



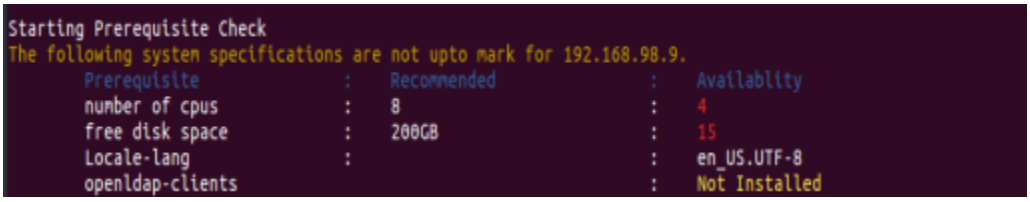

The upgrade process is triggered, which stops the previous running AppViewX components.




After the components are stopped, the prerequisite check for the node is performed. The patch file is copied, extracted, and applied to the node.

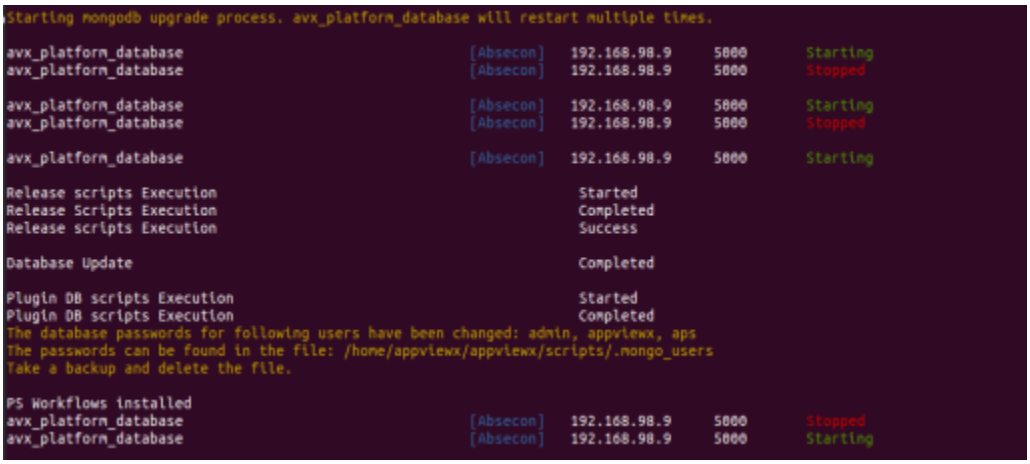



The prerequisite check for the 2019.4.0 upgrade starts.

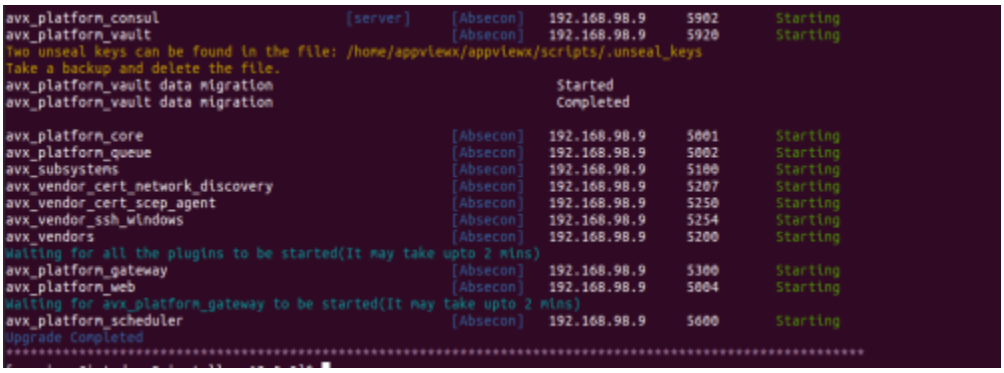


The initialization process for the 2019.4.0 upgrade starts.

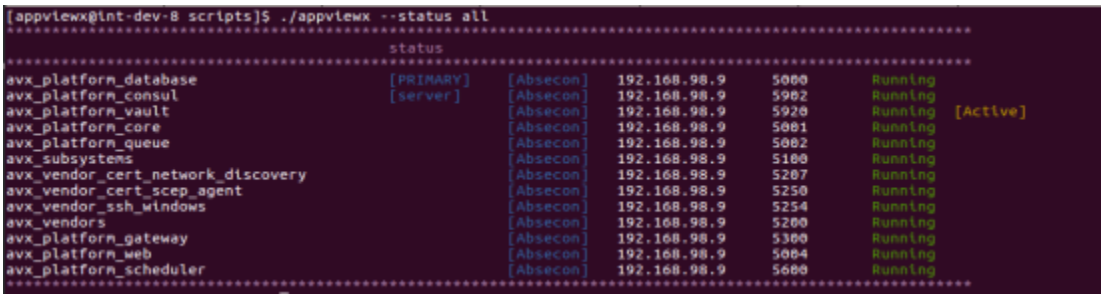

The **avx_platform_database** starts and data migration begins.




The plugins defined in the plugins **.meta** configuration starts.



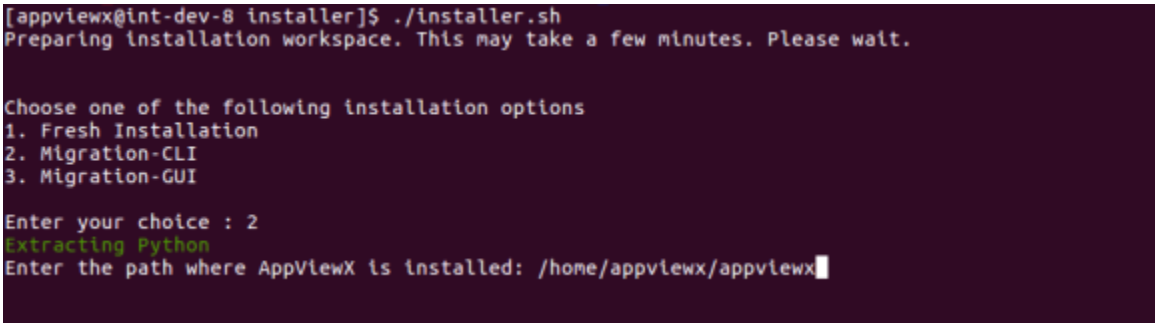

3. Validate the status of the upgrade using the following command and ensure that all components are up and running: `$ cd <avx_installed_directory>/scripts && ./appviewx --status all`



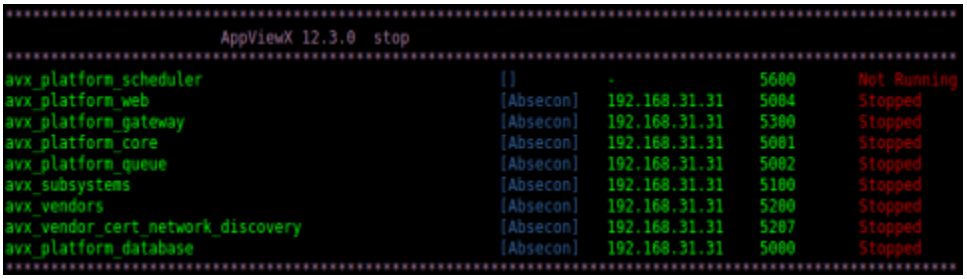

4. Access the application by opening a browser on the host machine and entering **https://<web_ip>:<web_port>**.

## Migrating via GUI

1. Select the option **Migration GUI** to trigger the migration.
2. Specify the path where the earlier version is currently running and press **Enter** on your keyboard.



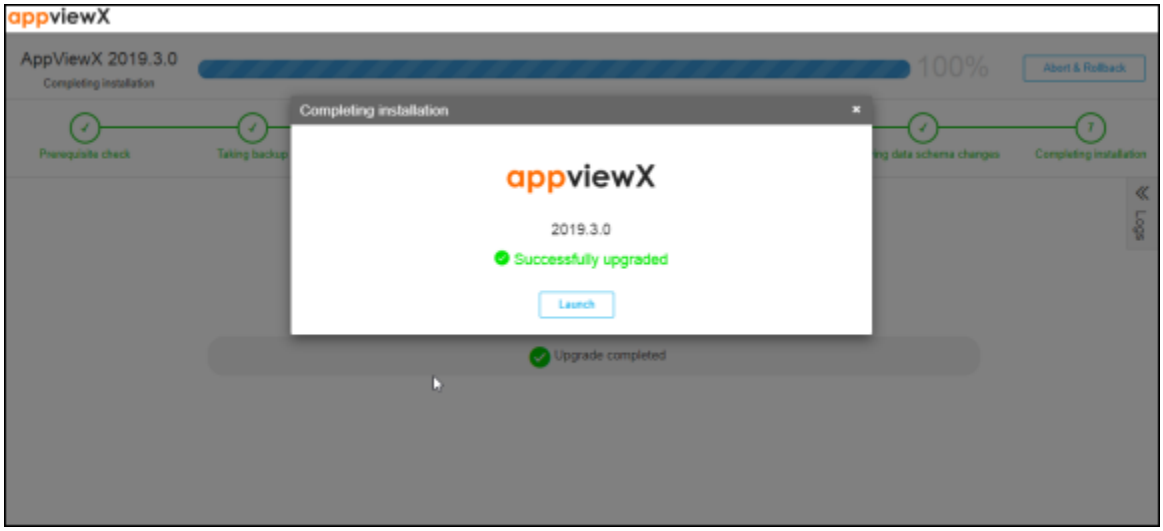

The upgrade process is triggered, which stops the previous running AppViewX components.




- After the components are stopped, the prerequisite check for the node is performed.
- AppViewX now takes a backup of the whole application.
- A unique URL will be generated.

3. Copy that URL and paste it in your browser.
   - Now the upgrade process will begin. You can track the installation progress on the window.
   - You can view the step-wise details by clicking the **Logs** icon.

- You will see the following screen once the process is complete:




4. Click **Launch**.

   You will be redirected to the AppViewX login page.

**Note:** During migration, random passwords generated for the MongoDB users will be available in the temporary file location: **<appviewx_dir>/scripts/.mongo_users** (hidden file). This file location will be displayed during the installation process. Random passwords will be generated only for the users whose passwords were not changed.

- Case 1 - If the admin password has been changed, then no random passwords will be generated.
- Case 2 - If the admin password has not been changed but the appviewx user password has been changed, then the random passwords will be generated for an admin and aps user.
- Case 3- If the admin password has not been changed but the aps user password has been changed, then the random passwords will be generated for an admin and appviewx user.
- Case 4 - If the admin password has not been changed but the appviewx and aps user passwords have been changed, then the random passwords will be generated for an admin.
- Case 5 - If the password has been changed for all the users (admin, appviewx, and aps), then no random passwords will be generated.
- The passwords that were changed will only get updated in the .mongo_users file and the user will be intimated. If no password has been changed, the user will not be intimated and .mongo_users file will not be created.

If the proxy has been configured for the CA settings, please execute the below command to update the changes `appviewx --restart plugins avx_vendors`.

# Chapter 9: AppViewX Native Upgrade: Multi-Node



## Prerequisites

Before installing AppViewX, make sure the following are true:

- The upgrade release package in **.tar.gz** format has been downloaded from https://release.appviewx.com to the **<user_home_directory>**.
- The AppViewX addons in **.tar.gz** format have been downloaded from https://release.appviewx.com to **<user_home_directory>**
- Move **appviewx_addons.tar.gz** to the **installer** directory using the following command: `mv ../appviewx_addons.tar.gz`.
- The open files [`ulimit -n`] and the maximum processes [`ulimit -u`] should be **65535** with port (SSH) **22** opened.
- If an elastic search is enabled, make sure the following are true:
  - The open files [`ulimit -n`] should be **65536**.
  - The virtual map count [`sysctl vm.max_map_count`] should be **262144**.
- The system time difference between the cluster nodes does not exceed 10 seconds.
- The component ports are accessible across the cluster.
- An SSH key is shared across the cluster nodes.
- Locale language should be **en_US.utf8**.
  - Execute below command to check default locale language: `localectl`
  - To set locale language: As a root user, execute `localectl set-locale LANG=<locale name>`
- **CLIENT_CERT** should not be enabled when a proxy is available. Run the following command as a root user to set the virtual map count to **262144**:

> **Note:** Run the following command as a root user to set the virtual map count to 262144:
>
> - `echo "vm.max_map_count=262144" >> /etc/sysctl.conf`
> - `sysctl --system`

## OS Requirements

| OS Platforms Supported | Versions | VM or OVA Support | Packages | CPU | RAM | HDD |
|---|---|---|---|---|---|---|
| CentOS | 6.x, 7.x | Yes | NC, NMAP-NCAT, NMAP, CURL, SYSSTAT, TCPDUMP, RSYNC, NETSTAT, ZIP, UNZIP, OPENSSL, BIND-UTILS, FONT CONFIG (version 2.13.0), GIT, GIT-LFS, RSNAPSHOT. | 8v | 32 GB | 1 TB |
| RHEL | 6.x, 7.x | No | NC, NMAP-NCAT, NMAP, CURL, SYSSTAT, TCPDUMP, RSYNC, NETSTAT, ZIP, UNZIP, OPENSSL, BIND-UTILS, FONT CONFIG (version 2.13.0), GIT, GIT-LFS, RSNAPSHOT. | 8v | 32 GB | 1 TB |

**Note:** To configure **RSNAPSHOT**, download the **.config** file from this link. In that config file, you can configure the value of **snapshot_root** (specify a location with free space) and backup (the location where AppViewX is installed).

**Note:** The following packages are required only in specific cases:

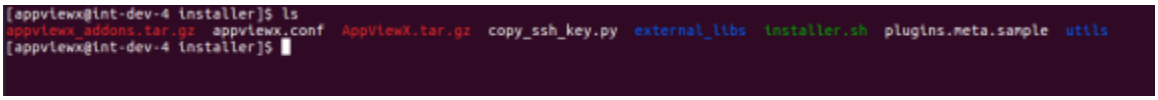
- OpenLDAP-CLIENTS to support the SSH subsystem.
- BZIP2 to support elastic search.

**Note:** The SSH+ plugin will not be supported in CentOS 6.x and Redhat 6.x.

## Upgrading AppViewX

1. Go to the user home directory and untar the .tar.gz upgrade package by entering the following command: `$ cd <user_home_directory> && tar -xvf appviewx_2019.4.0_BXX_RXXXX.tar.gz`
2. Go to the **installer** directory by executing the following command: `$ cd installer`
3. Move **appviewx_addons.tar.gz** to the **installer** directory using the following command: `mv ../appviewx_addons.tar.gz`.
4. Once moved, complete the steps provided in this section.

5. Type the command `ls` to verify the existence of the following files: **AppViewX.tar.gz**, **installer.sh**, **copy_ssh_key.py**, **plugins.meta.sample**, and **appviewx.conf**.




6. To proceed with the new configuration, you need to follow steps 4 to 13. If not, it will proceed with the old configuration setup. If you are upgrading from a version below 12.0., you have to follow the new configuration steps. Old configuration steps will not work for versions below 12.0.

> **Note:** The Consul must have been configured in an odd number of nodes and the Vault must have been configured in the node where a consul is configured. Also, it is not mandatory to configure a vault in the database nodes. It is recommended to configure the vault in two nodes and the consul in three nodes for a multinode. If the existing AppViewX version is 12.4.0 or later, make sure you update the VAULT section the same as the existing version in the appviewx.conf file.

7. Go to the plugins.meta.sample file using the following command: `$ vi plugins.meta.sample`
8. Update the IP and PORT details of the nodes where platform-specific plugins are going to be installed.

```
avx_platform_core=192.168.31.32:5001,192.168.31.34:5001
avx_platform_queue=192.168.31.32:5001,192.168.31.31:5001
#avx_platform_insight=localhost:5003
#avx_platform_syslog=localhost:5005
#avx_platform_syslog_receiver=localhost:5006
```

9. Update the IP and PORT details of the nodes where subsystem plugins are going to be installed.

```
avx_subsystems = 192.168.31.32:5100,192.168.31.33:5100
```

10. Update the IP and PORT details of the node where vendor plugins are going to be installed.

```
avx_vendors = 192.168.31.32:5200,192.168.31.33:5200
```

11. Update the external_certificate upgrade configuration to **True** if an external certificate is already in use. Update the external_certificate upgrade configuration to **False** if a self-signed certificate is already in use.

```
#####################################################################
##
## External_Certificate should be either True or False
##
#####################################################################

[SSL]

External_Certificate=True
```

12. Set the **ENABLE_VAULT** field in the **VAULT** section to **TRUE** and update the following fields to install the consul and vault components:

> **Note:** The Consul must have been configured in an odd number of nodes and the Vault must have been configured in the node where a consul is configured. Also, it is not mandatory to configure a vault in the database nodes. It is recommended to configure the vault in two nodes and the consul in three nodes for a multinode. If the existing AppViewX version is 12.4.0 or later, make sure you update the VAULT section same as the existing version in the `appviewx.conf` file.

- CONSUL_CLUSTER
- CONSUL_CLIENT_PORT
- HOSTS
- VAULT_CLUSTER_PORT
- LOG_LEVEL

```
[VAULT]
ENABLE_VAULT = True

CONSUL_CLUSTER = localhost:5902

##------------------------------------------------
## The consul client port only needs to be configured in case of a multinode setup
##------------------------------------------------
CONSUL_CLIENT_PORT = 5912

##------------------------------------------------
## VAULT_CLIENTS should be preferrably configured on gateway nodes
##------------------------------------------------
HOSTS = localhost:5920
VAULT_CLUSTER_PORT = 5921

##------------------------------------------------
## Possible values: info / debug / trace
##------------------------------------------------
LOG_LEVEL = Info
```

13. Update the custom SSH port configuration if required where default communication is through port 22.

```
#####################################################################
##
## SSH_PORT should have the value of the AppViewX environment SSH Port
## In case of multinode setup, it should be comma separated values.
##
## Eg:
##       SSH_PORT=22         (Singlenode)
##       SSH_PORT=22,22,22   (Multinode)
##
#####################################################################

[ENVIRONMENT]
SSH_PORT=22
```

For more details on how to configure syslog, refer to the Enable SYSLOGS Reception from Devices section of this guide.

14. When you are done editing the fields, press the **Esc** key, then type `:wq` to save and quit the file.
15. Rename the file to **plugin.meta**.
16. If the latest patches are available for particular versions, then the latest patch file can be downloaded and applied. Download the patch file from the release portal and extract it in the **installer** directory.

```
[appviewx@int-dev-8 installer]$ ls
AppViewX_2019.1.1_Latest_Plugins          AppViewX.tar.gz      utils
AppViewX_2019.1.1_Latest_Plugins.tar.gz   installer.sh
appviewx.conf                             plugins.meta.sample
```

17. Trigger the migration process with the following command: `$ ./installer.sh`

    Upon executing the install script, the user will be prompted with the following options:

    - Fresh Installation
    - Migration CLI
    - Migration GUI

## Migrating via CLI

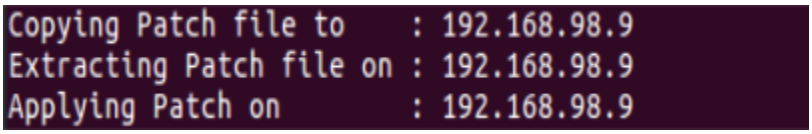
1. Select the option **Migration CLI** to trigger the migration. If you want to upgrade the application in the GUI, please refer to point No. 24.
2. Specify the path where the earlier version is currently running and press **Enter** on your keyboard.

```
[appviewx@int-dev-8 installer]$ ./installer.sh
Preparing installation workspace. This may take a few minutes. Please wait.


Choose one of the following installation options
1. Fresh Installation
2. Migration-CLI
3. Migration-GUI

Enter your choice : 2
Extracting Python
Enter the path where AppViewX is installed: /home/appviewx/appviewx
```

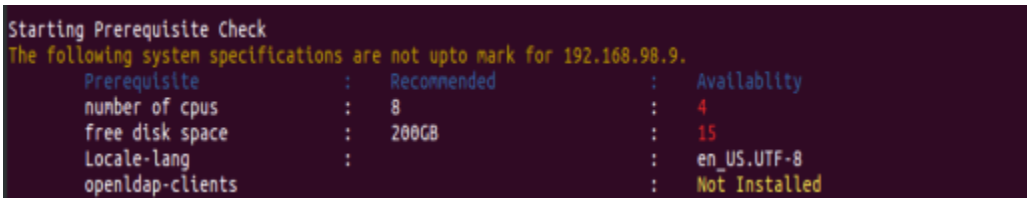
The upgrade process is triggered, which stops the previous running AppViewX components.

```
**********************************************************************************************
                    AppViewX 12.3.0  stop
**********************************************************************************************
avx_platform_scheduler                    []          .               5600      Not Running
avx_platform_web                          [Absecon]   192.168.31.31   5004      Stopped
avx_platform_gateway                      [Absecon]   192.168.31.31   5300      Stopped
avx_platform_core                         [Absecon]   192.168.31.31   5001      Stopped
avx_platform_queue                        [Absecon]   192.168.31.31   5002      Stopped
avx_subsystems                            [Absecon]   192.168.31.31   5100      Stopped
avx_vendors                               [Absecon]   192.168.31.31   5200      Stopped
avx_vendor_cert_network_discovery         [Absecon]   192.168.31.31   5207      Stopped
avx_platform_database                     [Absecon]   192.168.31.31   5000      Stopped
**********************************************************************************************
```

After the components are stopped, the prerequisite check for the node is performed. The patch file is copied, extracted, and applied to the node.

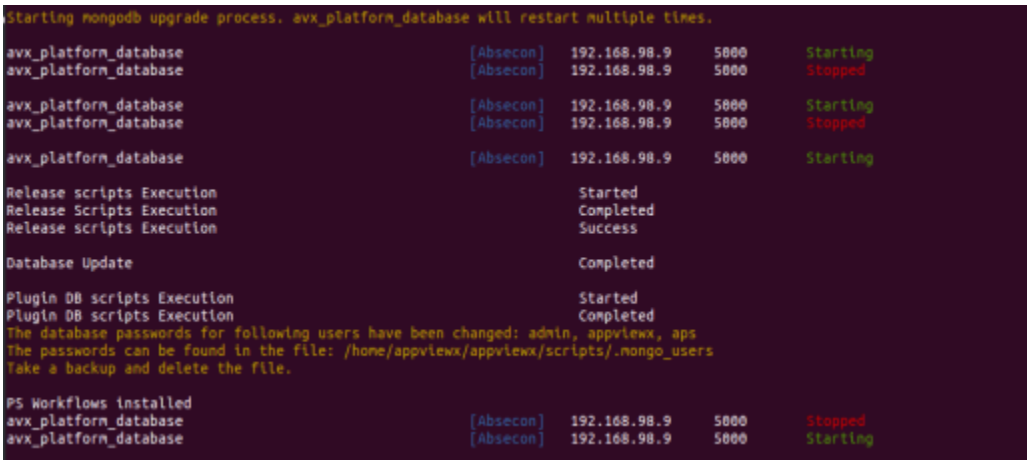



The prerequisite check for the 2019.4.0 upgrade starts.



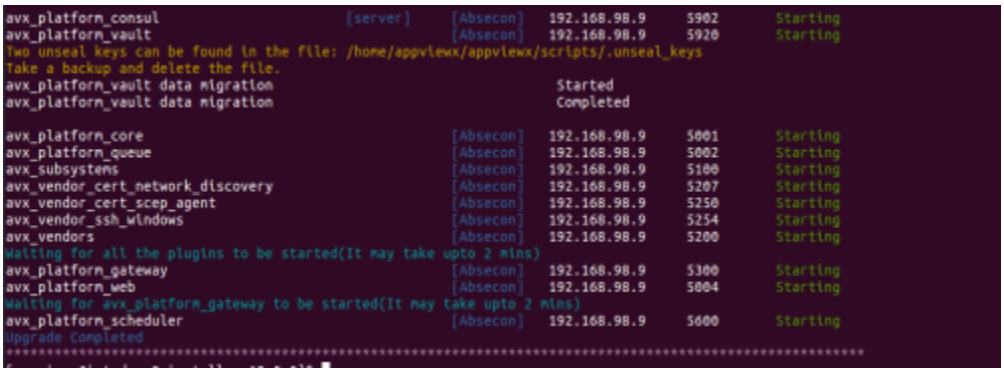

The initialization process for the 2019.4.0 upgrade starts.


The **avx_platform_database** starts and data migration begins.

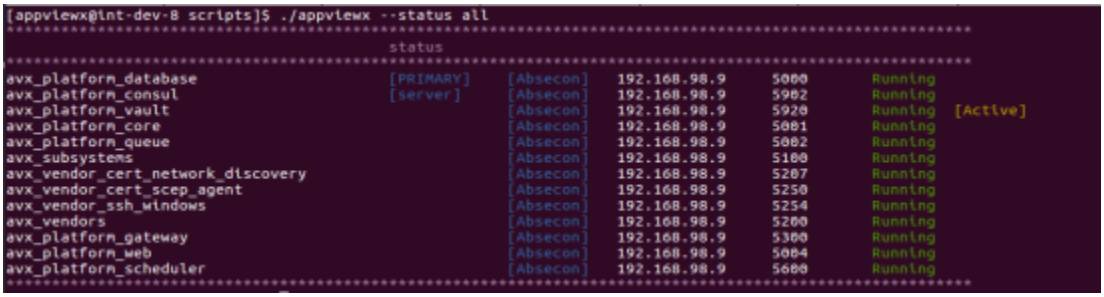



**Note:** The database migration process might take a while based on the data to be migrated from the previous version of AppViewX

The plugins defined in the plugins **.meta** configuration starts.



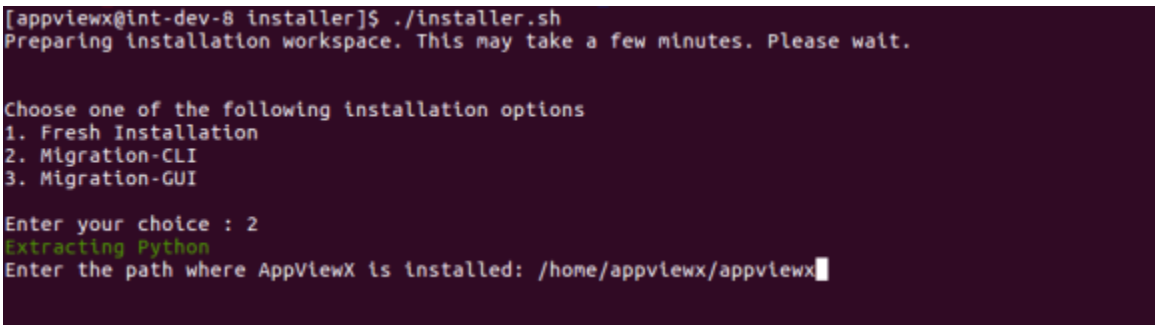

3. Validate the status of the upgrade using the following command and ensure that all components are up and running: `$ cd <avx_installed_directory>/scripts && ./appviewx --status all`



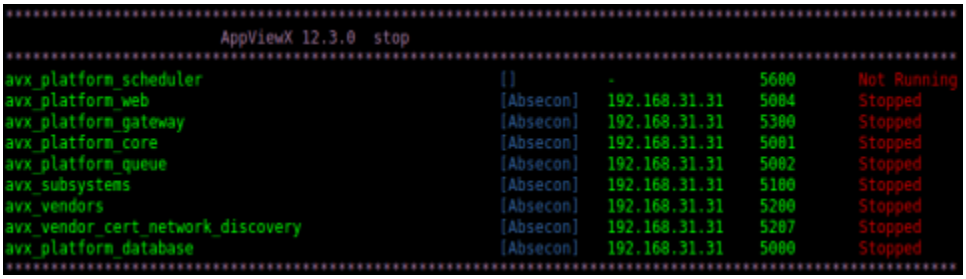

4. For any customers who are having HSM devices managed in appviewx any version prior to 2020.1.0, kindly ensure to execute `appviewx --kek-migration` once after all the components are up and running post-migration.
5. Access the application by opening a browser on the host machine and entering **https://<web_ip>:<web_port>**.

## Migrating via GUI

1. (Alternative Option) Select the option **Migration GUI** to trigger the migration.
2. Specify the path where the earlier version is currently running and press **Enter** on your keyboard.

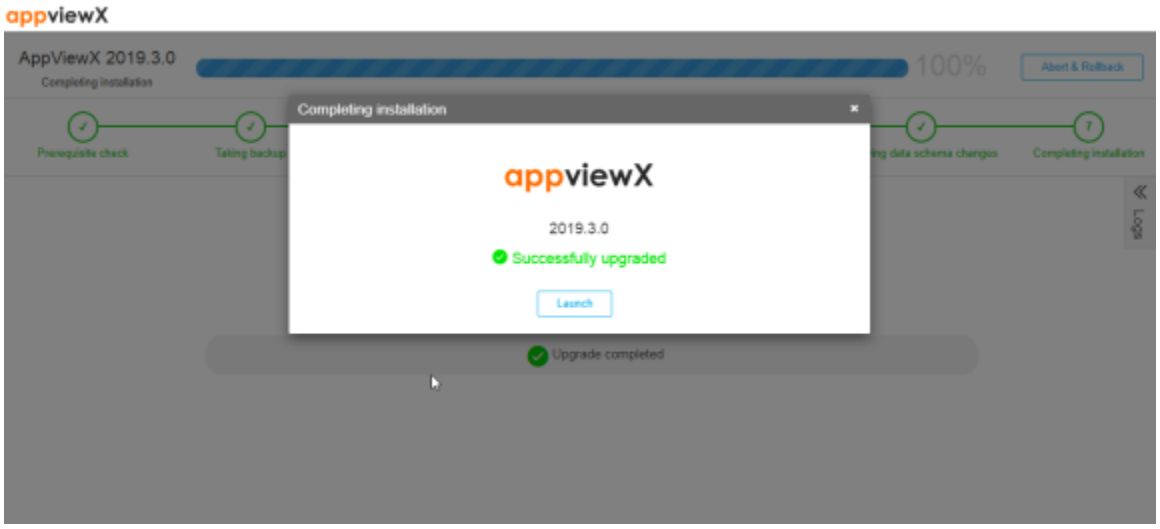


The upgrade process is triggered, which stops the previous running AppViewX components.




- After the components are stopped, the prerequisite check for the node is performed.
- AppViewX now takes a backup of the whole application.
- A unique URL will be generated.

3. Copy that URL and paste it in your browser.
   - Now the upgrade process will begin. You can track the installation progress on the window.
   - You can view the step-wise details by clicking the **Logs** icon.
   - You will see the following screen once the process is complete:




4. Click **Launch**.

   You will be redirected to the AppViewX login page.

> **Note:** During migration, random passwords generated for the MongoDB users will be available in the temporary file location: **<appviewx_dir>/scripts/.mongo_users** (hidden file). This file location will be displayed during the installation process. Random passwords will be generated only for the users whose passwords were not changed.

- Case 1 - If the admin password has been changed, then no random passwords will be generated.
- Case 2 - If the admin password has not been changed but the appviewx user password has been changed, then the random passwords will be generated for an admin and aps user.
- Case 3- If the admin password has not been changed but the aps user password has been changed, then the random passwords will be generated for an admin and appviewx user.
- Case 4 - If the admin password has not been changed but the appviewx and aps user passwords have been changed, then the random passwords will be generated for an admin.
- Case 5 - If the password has been changed for all the users (admin, appviewx, and aps), then no random passwords will be generated.
- The passwords that were changed will only get updated in the .mongo_users file and the user will be intimated. If no password has been changed, the user will not be intimated and .mongo_users file will not be created.

If the proxy has been configured for the CA settings, please execute the below command to update the changes `appviewx --restart plugins avx_vendors`.

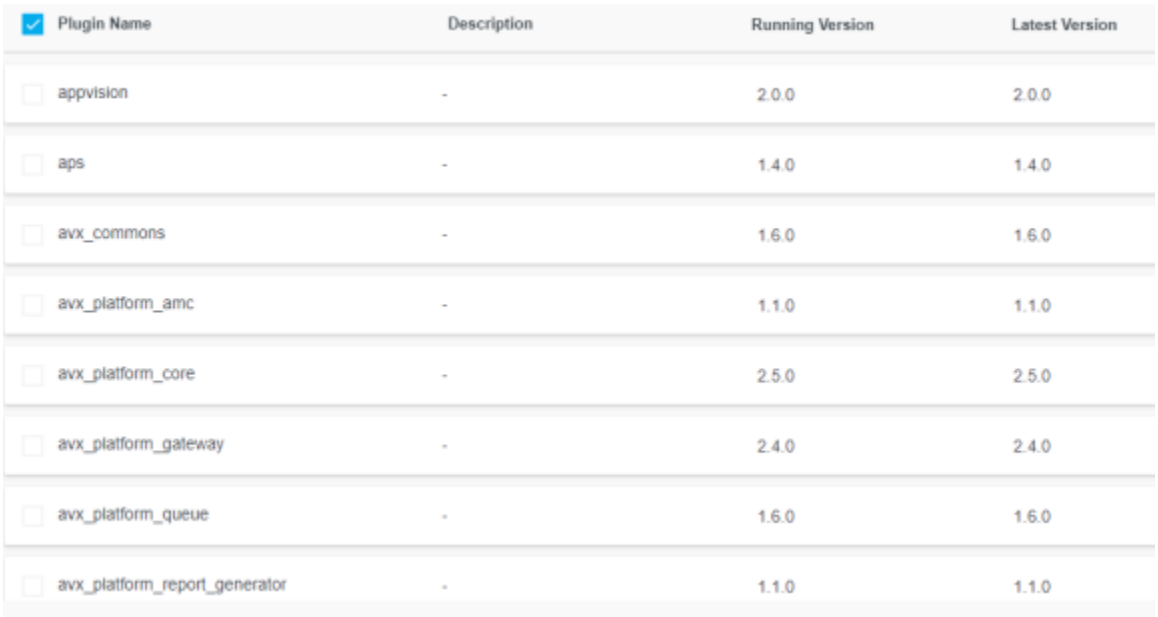
# Chapter 10: AppViewX Plugin Upgrade



## AppViewX Plugin Upgrade

Users can manage plugins in the **System** module.

1. Click **Menu** >> **System** >> **Plugins Manager**.
2. On the **Manage Plugins** page, you will find two sections: **Installed** and **Available**.
   You will be directed to the **Installed** section by default. You can click **Available** to view the list of plugins that are not installed currently, but offered by AppViewX.
3. In the section you will find a list of plugins with details such as **Plugin Name**, **Description**, **Running Version**, and **Latest Version**. There is also a search bar to search for a particular plugin.

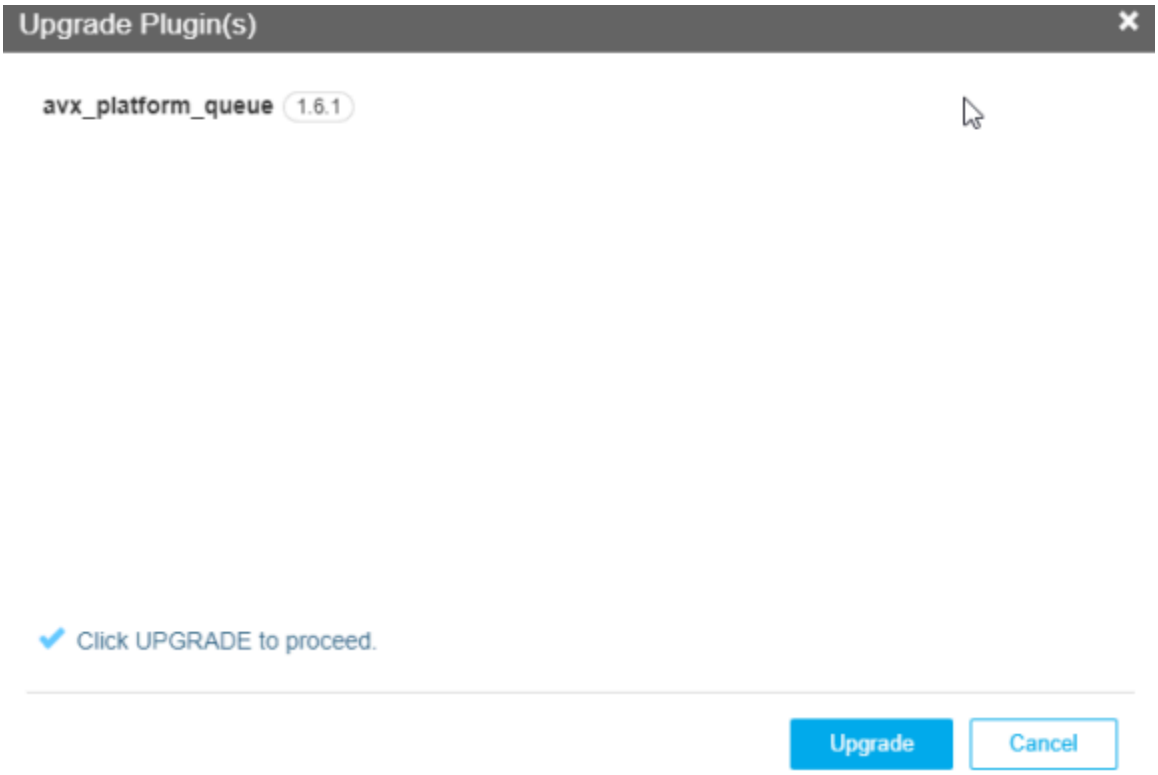


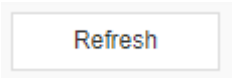
| Plugin Name | Description | Running Version | Latest Version |
|---|---|---|---|
| appvision | - | 2.0.0 | 2.0.0 |
| aps | - | 1.4.0 | 1.4.0 |
| avx_commons | - | 1.6.0 | 1.6.0 |
| avx_platform_amc | - | 1.1.0 | 1.1.0 |
| avx_platform_core | - | 2.5.0 | 2.5.0 |
| avx_platform_gateway | - | 2.4.0 | 2.4.0 |
| avx_platform_queue | - | 1.6.0 | 1.6.0 |
| avx_platform_report_generator | - | 1.1.0 | 1.1.0 |

4. To view the changelog click this icon  near the respective plugin.

5. To upgrade the plugin to the latest version, click the 

icon near the respective plugin.
6. Click **Upgrade** again when the following screen appears:




7. Click the 

icon to refresh the current list of plugins.

## Performing Actions

To perform operations on one or multiple plugins:

1. Select the checkbox before respective plugins.
2. Click the **Actions** icon.

   Now you can upgrade or download the selected plugins.

## Upload Plugin

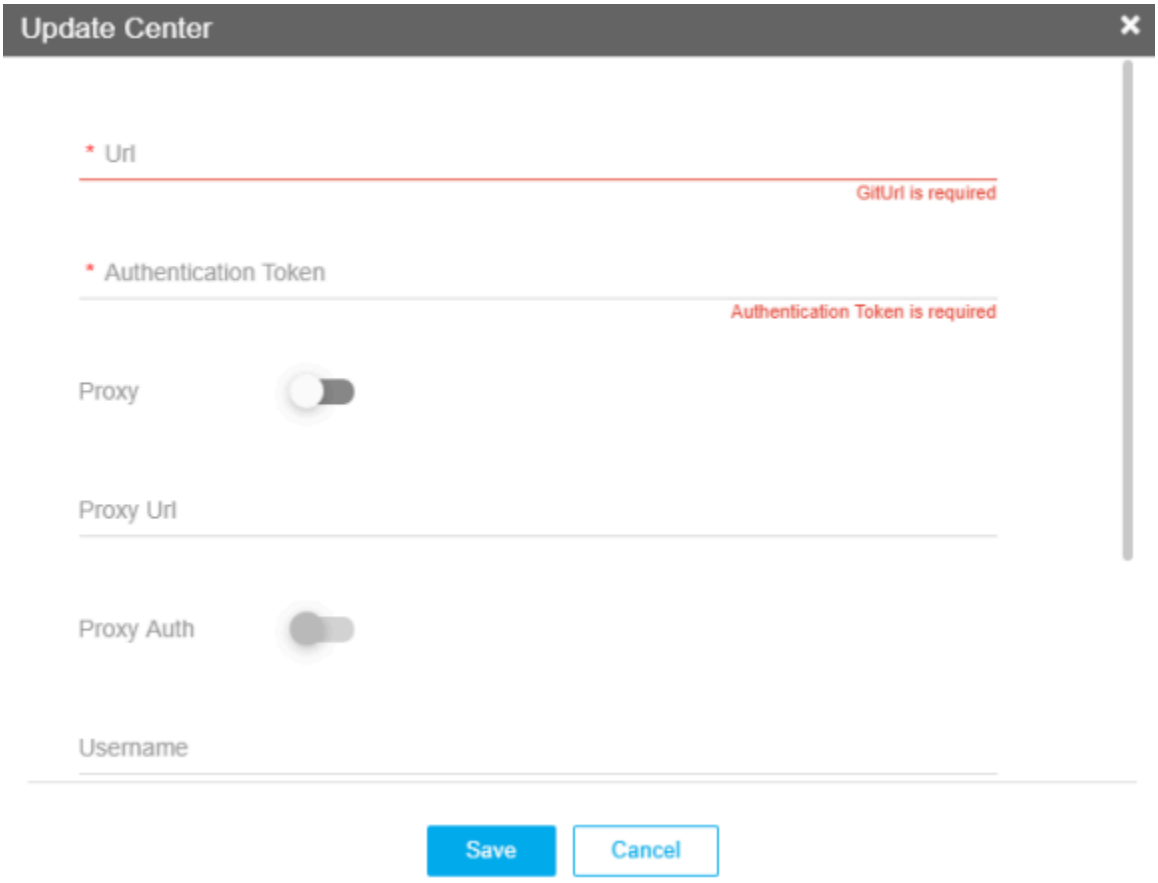
To upload a plugin from your local machine:

1. Select the Upload Plugin (**Upload Plugin**) icon from the top right corner of the screen.
2. Select a plugin in the **TAR.GZ*** format and then click **Upload**.

## Settings

To view the **Update Center**:

1. Click the **Settings** icon from the top right corner of the screen.
   The following screen appears:



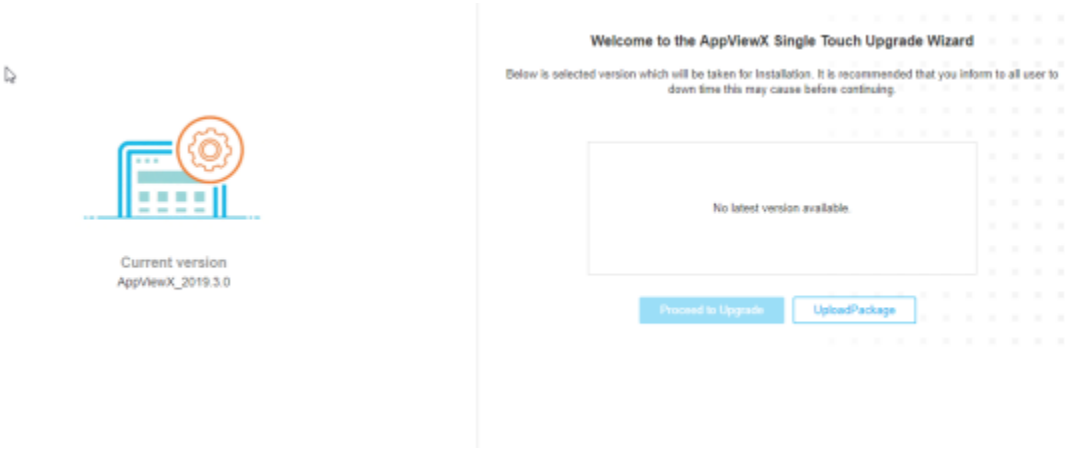

2. Enter the URL and the Authentication Token after receiving the details from the AppViewX support. These details are mandatory inputs.

3. You can enable Proxy if you do not have a direct internet connection. If that proxy server requires authentication, enable the Proxy Authentication. You can enter the credentials in the respective fields and click **Save**.
4. You also have the option to select the **Sync Time** which is the interval where AppViewX checks the existing list of plugins for the latest version available. You can choose between **daily, weekly, monthly**, and **yearly**.

## Platform Upgrade

You can now update your current product version using this feature. To do so complete the following steps:

1. Go to **Menu** > **System** > **System Update**.
   You will be directed to the following screen:

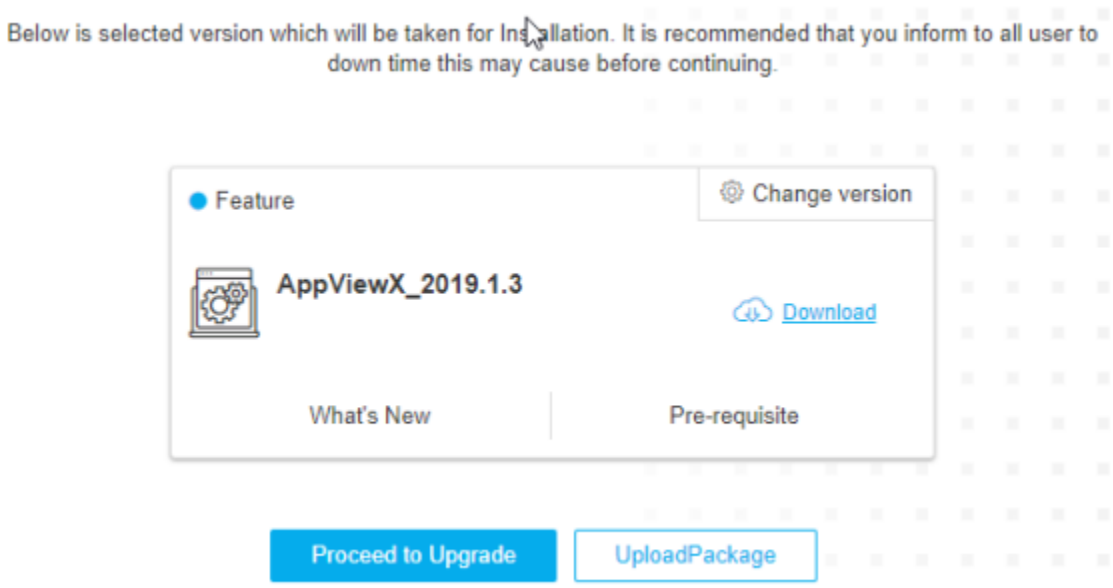



   In the left pane, you can see the current version of the product that you are running. In the right pane, you can see the available list of incremental upgrades for the AppViewX product.

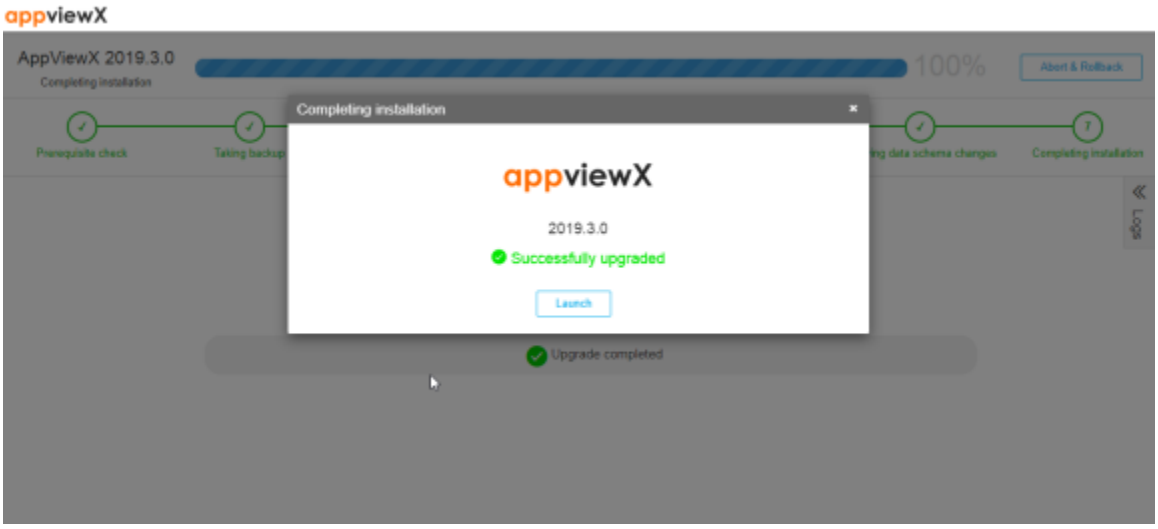
   **Note:** You can only upgrade to a higher version of the product. You will not be able to downgrade to a lower version.

   If you are already running the latest version of AppViewX, you will see this message on the right pane: 'No latest version available'

   If you are running an older version of AppViewX, then you will see the following screen on the right pane:




2. Here, you will find the latest AppViewX version. If you want to upgrade to an intermediate version (Versions that have been released during the timeline between your current release and the latest release) you can select **Change Version** and select the respective version that you want to upgrade to.
   - To know what are new features available in the particular release click **What's New**.
   - To know about the prerequisites needed to perform the upgrade click **Pre-requisite**.
3. If you have any Internet limitations for you to proceed to the upgrade directly, you can also download the update file locally to your machine and then upgrade to that release. To do so, click **Download**.
4. Once you have downloaded the file locally, click **Upload Package**.
5. Select the file on your local machine and then click **Install**.
   Now the upgrade process will begin. You can track the progress on the window.
6. You can view the step-wise details by clicking the **Logs** icon.
   You will see the following screen once the process is complete:




7. Click **Launch**.

   You will be redirected to the AppViewX login page.

# Chapter 11: AppViewX License Generation



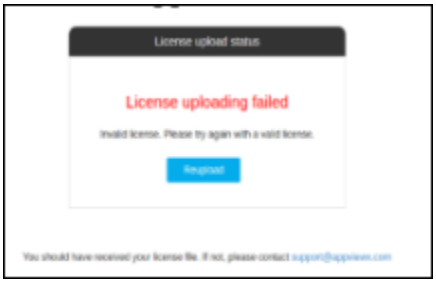
## Troubleshooting

The following issues can cause an error while uploading a license:



### License Upload Failure Due To Invalid Hostname

**Issue:** License upload fails with the error as shown below:

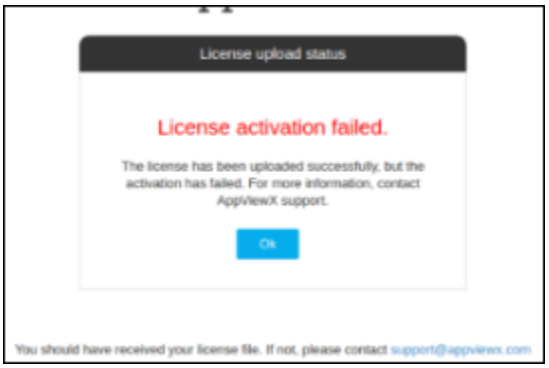



**Solution**:

1. Fetch the hostname on which the license is going to be generated by entering the following command: `$ appviewx --license host-fetch`.
2. Generate and upload a new license with the hostname retrieved.

### License Upload Failure With A License Activation Error

**Issue:** License upload fails with a license activation error




This issue can occur when the gateway refresh fails due to a session timeout.

**Solution:**

To resolve it, perform a gateway refresh by entering the following commands:

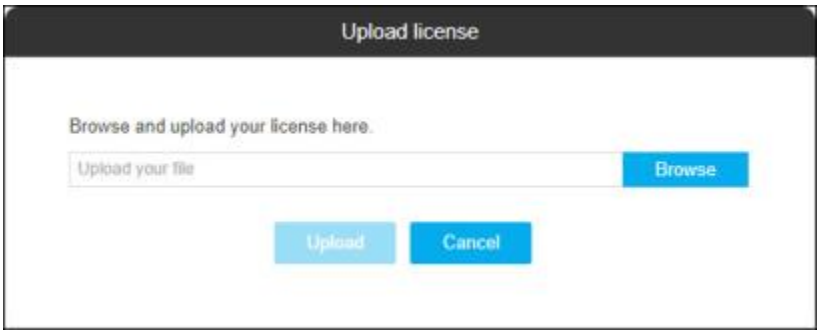
```
$ cd <avx_installed_directory>/scripts && ./appviewx --gwrefresh

$ ./appviewx --restart avx_platform_scheduler
```

# Renew an AppViewX License

To renew an AppViewX license:

1. Log in to AppViewX GUI using the credentials:
   **<http(s)>:<appviewx_web_ip>:<appviewx_web_port>**
2. Click the **Menu** button.
3. Go to **Settings** >> **General** >>**License**.
   The **License** screen appears, showing details about the main license and all subscribed licenses.
4. Click the **Upgrade License** button to renew the license.

5. On the **Upload License** screen that appears, click **Browse** and then locate and select the license upgrade file.




6. As soon as you select a file, the **Upload** button becomes active. Click it to begin the upload process. The screen then displays a license preview screen listing the modules that you have purchased and the license limits and counts for each.


7. Click **Continue**.
   As soon as the license is ploaded and activated, the screen displays the message, "The license has been uploaded and activated successfully."
8. Click **Ok** to return to the login screen.
   When you log in again, your newly upgraded license will be active.

**Troubleshooting:**

- If the license is uploaded and not activated, the screen displays the message: **The license has been uploaded successfully, but the activation has failed. For more information contact AppViewX support.**
- If the license upload has failed, the screen displays the message: **The license upload has failed, try uploading it again.**

# Chapter 12: Troubleshooting



## Deployment Issues

**avx_platform_scheduler** shows the status as **Not Started**.

**Issue**: The status of the **avx_platform_scheduler** is **Not Started**.

**Solution**: Scheduler has a dependency on the database and the gateway, so the gateway might take some time to come up. After the gateway is in a Running state, the avx_platform_scheduler should start automatically. If the scheduler still does not start, start it manually by entering the following command: `$ appviewx --start avx_platform_scheduler`

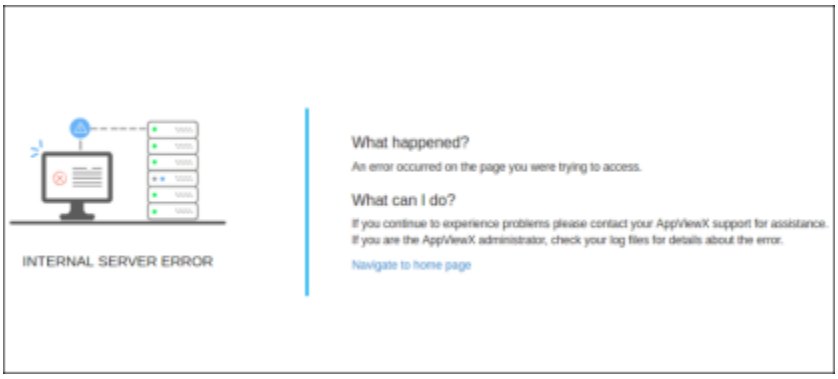
## Post-Deployment Issues

This section covers solutions to issues that might occur after you deploy AppViewX.



### Web UI Throws a 500 Internal Server Error

**Issue:**After you log in to the application from the web, the screen displays a 500 internal server error.



**Solution**: Restart the following two plugins of AppViewX:

- **avx_platform_core**
- **avx_platform_gateway**

1. To restart the plugins, execute the following commands:

```
$ appviewx --restart plugins avx_platform_core

$ appviewx --restart avx_platform_gateway
```

2. After you restart them, check the status of the plugins using the following commands:

```
$ appviewx --status plugins avx_platform_core

$ appviewx --status avx_platform_gateway
```

3. If the 500 error persists, enter the following command to refresh the gateway after all the plugins are in a **Running** state: `$ appviewx --gwrefresh`

## 404 Error When Hitting the Web URL

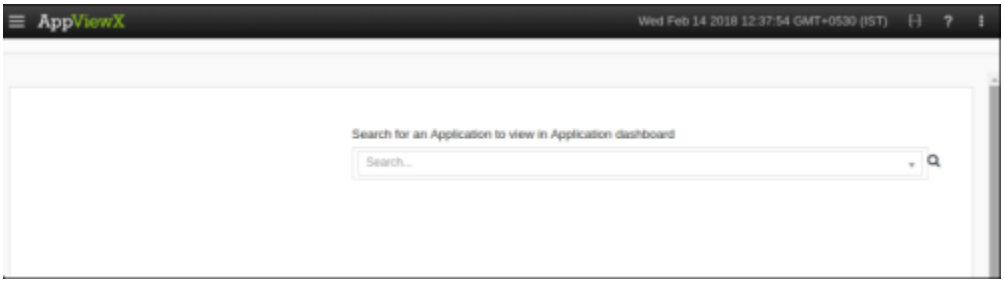
**Issue**: A 404 Error appears when you try to access AppViewX.

**Solution**: Initialize the gateway and then restart the web and plugins by entering the following commands:

```
$ appviewx --initialize avx_platform_gateway

$ appviewx --restart avx_platform_web
```

## Menu not displayed When Opening a Module from the UI

**Issue**: The menu is not displayed when you try to open a module within the UI.




**Solution**: This error can occur when opening individual module pages from the web. To resolve it:

1. Restart the **avx_platform_queue** by entering the following command: `$ appviewx --restart plugins avx_platform_queue`
2. After the plugin displays a status of **Running**, refresh the gateway by entering the following command: `$ appviewx --gwrefresh`

## Windows Gateway Errors and Solutions

| Error | Solution |
|---|---|
| 767cf2b6-bfc3-45a0-9490-a95cf841e693: Connecting to remote server <machine name> failed with the following error message : WinRM cannot process the request. The following error occurred while using Kerberos authentication: The computer <name> is unknown to Kerberos. Verify that the computer exists on the network, that the name provided is spelled correctly, and that the Kerberos configuration for accessing the computer is correct. The most common Kerberos configuration issue is that an SPN with the format HTTP/<machine name> is not configured for the target. If Kerberos is not required, specify the Negotiate authentication mechanism and resubmit the operation. For more information, see the about_Remote_Troubleshooting Help topic. | • This issue occurs with Powershell remoting as it uses Kerberos authe<br>• In the agent machine, start the command prompt as an administrator<br>`setspn -s http/machinename domainusername`.<br>• This will work in the environments where Kerberos authentication and<br>• If no kerberos authentication is set up, then the communication must |
| Retrieving the COM class factory for remote component with CLSID | • The component used for accessing CA (**certadm.dll**) is not installed<br>• Check if the DLL is available in **C:WindowsSystem32** folder or else, **Server Administration Tools (RSAT)** for the respective OS.<br><br>For example, for Windows 10 https://www.microsoft.com/en-in/downl |
| PowerShell ScriptExecution Error: Access is denied. 0x80070005 (WIN32: 5) OR Error Code 0x80070005 - Access is denied | • The username must be configured as **Username@Domain**.<br>• The user must have admin access to the remote/target machine or m administrator group.<br>• Go to the Local Users and Groups and access **Administrators**. Chec username is a part of the administrator group. |

| Error | Solution |
|---|---|
| Connecting to remote server <machine name> failed with the following error message: WinRM cannot process the request. The following error with error code 0x80090322 occurred while using Negotiate authentication: An unknown security error occurred. | • This issue occurs with Powershell remoting as it uses Kerberos authe<br>• In the agent machine, start the command prompt as an administrator<br>`setspn -s http/machinename domainusername`.<br>• This will work in the environments where Kerberos authentication and<br>• If no kerberos authentication is set up, then the communication must |
| The WinRM client received an HTTP status code of 502 from the remote WS-Management service. For more information, see the about_Remote_Troubleshooting Help topic | • Check if the **WinRM** service is running.<br>• Go to the **Powershell** on the target machine and run the command `w`<br>• Execute the command `Enable-PSRemoting -force`.<br>• Execute the command `netsh winhttp show proxy` and if a proxy is configure<br>command `netsh winhttp reset proxy`. |
| 41783361-015b-453f-b321-e31709b1850c: Connecting to remote server <machine name> failed with the following error message : Access is denied. For more information, see the about_Remote_Troubleshooting Help topic. | • The username must be configured as **Username@Domain**.<br>• The user must have admin access to the remote/target machine or m<br>administrator group.<br>• Go to the **Local Users and Groups** and access **Administrators** and<br>username is part of the administrator group.<br>• Check if the **WinRM** service is running.<br>• Go to **Powershell** on the target machine and execute the command `v`<br>• Execute the command `Enable-PSRemoting -force`. |
| The client cannot connect to the destination specified in the request. Verify that the service on the destination is running and is accepting requests. Consult the logs and documentation for the WS-Management service running on the destination, most commonly IIS or WinRM. If the destination is the WinRM service, run the following command on the destination to analyze and configure the WinRM service: `winrm quickconfig` | • Check if the **WinRM** service is running.<br>• Go to **Powershell** on the target machine and execute the command `v`<br>• Execute the command `Enable-PSRemoting -force`. |
| d4f98a6a-41ef-4864-9848-03a07e113d75: CCertRequest::Submit: The RPC server is unavailable. 0x800706ba (WIN32: 1722 RPC_S_SERVER_UNAVAILABLE) | Go to the target machine and start the RPC service if it is stopped. |

| Error | Solution |
|---|---|
| 727838ed-151e-46bf-883c-07ccb3a3989f: Connecting to remote server <machine name> failed with the following error message : The user name or password is incorrect. For more information, see the about_Remote_Troubleshooting Help topic. | • The username must be configured as **Username@Domain**.<br>• The user must have admin access to the remote/target machine or m administrator group.<br>• Go to the **Local Users and Groups** and access **Administrators** and username is part of the administrator group.<br>• Check if the **WinRM** service is running.<br>• Go to **Powershell** on the target machine and execute the command `v`<br>• Execute the command `Enable-PSRemoting -force`. |
| fd3812f9-030a-421c-81e7-0e0510ce49e0: Access to the path '\\<machine name>\C$\ \Windows\\Temp\\qgwwkqi3.fff' is denied. | • The username must be configured as **Username@Domain**.<br>• The user must have admin access to the remote/target machine or m administrator group.<br>• Go to the **Local Users and Groups** and access **Administrators** and username is part of the administrator group. |
| More than 5 connections are not allowed | • Run **Powershell** as an administrator.<br>• Check existing config: `winrm get winrm/config`.<br>• Change the settings to increase the **maxshellsperUser** to 100 on the issue is concurring.<br>`winrm set winrm/config/winrs '@{MaxConcurrentUsers="20"}'`<br>`winrm set winrm/config/winrs '@{MaxShellsPerUser="100"}'`<br>`winrm set winrm/config/winrs '@{MaxMemoryPerShellMB="512"}'` |
| Connecting to remote server failed with the following error message: The WS-Management service cannot process the request. This user is allowed a maximum number of 4 concurrent shells, which has been exceeded. Close existing shells or raise the quota for this user. | • Run **Powershell** as an administrator.<br>• Check existing config: `winrm get winrm/config`.<br>• Change the settings to increase the **maxshellsperUser** to 100 on the issue is concurring.<br>`winrm set winrm/config/winrs '@{MaxConcurrentUsers="20"}'`<br>`winrm set winrm/config/winrs '@{MaxShellsPerUser="100"}'`<br>`winrm set winrm/config/winrs '@{MaxMemoryPerShellMB="512"}'` |
| Client Certificate gives **Permission Denied 403** errors. This can happen in a certain environment and it's intermittent | • Check if the client certificate is installed correctly by validating the cha<br>• The root of the client certificate must be available in the **Trusted Roo** server.<br>• The intermediate of the client certificate must be available in the **Inter** authorities of the server.<br>• If all of the above are fine, go to the agent server and complete the fo |

| Error | Solution |
|---|---|
| | 1. MMC<br>2. Add/Remove **SnapIn**.<br>3. Select certificate.<br>4. Select **LocalMachine**.<br>5. Go to **Personal Store** and click on client certificate<br>6. Go to chain.<br>7. Export the root certificate and save as **Root.cer** in a location<br>8. Import the **Root.cer** into trusted root back again<br>9. If this does not solve the issue, then check if the trusted root conta<br>10. Click on **Trusted Root** store and check if there any certificate whic **IssuedBy** different<br>11. Take a backup of such certificates and move it to respective stores<br>12. If it does not solve the issue, then add the root certificate to the **Cli** |
| The permission on the certificate template do not allow the current user to enroll for this type of certificate | 1. Go to the CA server.<br>2. Open **Certificate Authority** and select the CA Server.<br>3. Right-click on **properties** and select the **Security** tab.<br>4. Check if the user used in **Agent** has the necessary permissions to re request certificate(s).<br>5. If the user is a part of a group, then ensure that the group has the re<br>6. Click on the **Certificate Templates** and right-click to manage the ter<br>7. Right-click on the template which has the issue and navigates to sec<br>8. Add permission to the user or group. |
| An attempt was made to open a **Certification Authority** database session, but there are already too many active sessions on a request using **CERTADMINLib.IenumCERTVIEWROW.Next()**. | In the CA server, navigate to the registry through the `regedit` command a<br><br>1. **HKLMSYSTEMCurrentControlSetServicesCertSvcConfiguration hex** (100 Dec)<br>2. **HKLMSYSTEMCurrentControlSetServicesCertSvcConfiguration** is also set to **64 hex** (100 Dec) |
| 803f4314-3a11-486a-87e5-367b8c5c6f9f: The user name or password is incorrect.rn | • The username must be configured as **Username@Domain**.<br>• The user must have admin access to the remote/target machine or m administrator group.<br>• Go to the **Local Users and Groups** and access **Administrators** and username is part of the administrator group. |
| 42abe1ef-2bff-40e8-82e2-c97c5707a0c1: Connecting to remote server <machine | The user name or password is incorrect. |

| Error | Solution |
|---|---|
| name> failed with the following error message : **The user name or password is incorrect. For more information, see the about_Remote_Troubleshooting Help topic.** | |
| Connecting to remote server <machine name> failed with the following error message: **WinRM cannot complete the operation. Verify that the specified computer name is valid, that the computer is accessible over the network, and that a firewall exception for the WinRM service is enabled and allows access from this computer. By default, the WinRM firewall exception for public profiles limits accesses to remote computers within the same local subnet. For more information, see the about_Remote_Troubleshooting Help topic.** | • **WinRM** service is already running on the following location of the mac **C:Windowssystem32>WinRM quickconfig**<br>• If **WinRM** is not set up to allow remote access to this machine for ma changes must be made:<br>1. Create a WinRM listener on **HTTP://*** to accept WS-Man requests<br>2. Make these changes [y/n]? y |
| There is not enough space on the disk | Ensure that your hard disk has enough free space. |
| Management Connect to remote machine <machine name> as user failed with the following error User credentials cannot be used for local connections | • The username must be configured as **Username@Domain**.<br>• The user must have admin access to the remote/target machine or m administrator group.<br>• Go to the **Local Users and Groups** and access **Administrators** and username is part of the administrator group.<br>• Configure the credentials in **AppViewX.CertPlus.Service Logon** opt |
| Denied by Policy Module 0x80094800, The request was for a certificate template that is not supported by the Active Directory Certificate Services policy: WebServer1. | Use template name instead of the template display name. |
| Device Communication failed while using Native option to connect to CA remotely | 1. Go to the agent machine.<br>2. Open **services.msc** using **Start** > **Run** command on the Windows m<br>3. Find the service **AppViewXCertPlus**.<br>4. Right-click and view properties.<br>5. Click on the **log on** tab.<br>6. Change the option to this account and enter the user account and pa |

| Error | Solution |
| --- | --- |
| | 7. Click on **Apply** and a message will popup to add the account as **Log** and save changes.<br>8. Click on **restart the service**.<br>9. Remove the username and password from AppViewX. |
| Certificate Request (CSR) is using a different account to request a certificate from CA as compared to account configured in AppViewX | 1. Go to the agent machine.<br>2. Open **services.msc** using **Start** > **Run** command on the Windows m<br>3. Find the service **AppViewXCertPlus**.<br>4. Right-click and view properties.<br>5. Click on the **log on** tab.<br>6. Change the option to this account and enter the user account and pa<br>7. Click on **Apply** and a message will popup to add the account as **Log** and save changes.<br>8. Click on **restart the service**.<br>9. Remove the username and password from AppViewX. |

# Chapter 13: Administrative Tasks

- Add New Plugins
- Upgrade a Plugin to a Newer Version
- Change an SSH PORT for Device Communication
- Update an SSL Configuration for Gateway and Web
- Collect Logs from Nodes
- Copy an SSH Key Across an Installation Node
- Change the Ulimit and Nlimit Configuration in the Node as a Root User
- Change the SSL Configuration
- Enable VIP for Web Access
- Enable a VIP for Gateway Access
- Reset GUI Admin Password
- Change the Port for a Plugin After Installation
- Enable SYSLOGS Reception from Devices
- Execute Command on All Nodes
- View Heap Size of the Plugins
- Update the Heap Size of the Plugins
- Update Log Level of the Plugins
- Configure an Elasticsearch
- Modify an Elasticsearch
- Backup and Restore of an Elasticsearch
- View the Version of AppViewX Components
- Set the Location for Database Backup
- Configure a TFTP Server
- Configure the Test Data
- Configure an SSL for the Database

- Configure a Fat JAR Deployment
- Change the Database Password
- Monitor the VIP Status
- Configure an AppViewX Git
- Configure a CyberArk Agent
- Configure a Proxy
- Update the Node Password
- Reverse DNS lookup
- Syslogs
- Troubleshooting Utility
- Prerequisites for SSH Deployment on CentOS 7
- Enable the Consul and Vault
- Restore a Database
- Get the Certificate Information
- Generate a New Certificate for the SSL Components
- Windows Gateway Installer
- Upgrade a Web Component
- Apply Release Patch
- Steps to Add Integration Libraries

## Add New Plugins

AppViewX uses plugins for serving specific functionalities. New plugins can add new features and they can be installed in the application on-demand using the following steps:

1. Go to the directory **<avx_installed_directory>/conf**
2. Go to the appviewx.conf file using the following command to add the new plugin to be installed: `$ vi appviewx.conf`
3. Update the plugin details under **ENABLED_PLUGINS** and also the individual plugin detail with the installation IP: Port

```
[PLUGINS]
ENABLED_PLUGINS = avx_platform_core,avx_platform_queue,avx_subsystem_adc,avx_vendor_f5,avx_subsystem_certificate
avx_platform_core = 192.168.96.111:5005,192.168.96.127:5005
avx_subsystem_adc = 192.168.96.111:5021,192.168.96.127:5021
avx_vendor_f5 = 192.168.96.111:5051,192.168.96.126:5051
avx_platform_queue = 192.168.96.111:5002,192.168.96.127:5002
avx_subsystem_certificate = 192.168.96.111:5006,192.168.96.127:5006
```

4. When you are done editing the fields, press the **Esc** key, then type:wq to save and quit the file. $ appviewx --conf-sync.

> **Note:** In case of multi node, run the following command to update the changes across all the cluster nodes: $ appviewx --conf-sync

5. To install the new plugins, execute the following commands:
   $ appviewx --initialize all $ appviewx --restart plugins <plugin_name> $ appviewx --restart gateway
6. After the components are up and running, perform a gateway refresh by entering the following command: $ appviewx --gwrefresh
7. Renew an AppViewX licenseif the plugin is for a new subsystem.

## Upgrade a Plugin to a Newer Version

AppViewX uses plugins for specific functionalities. The advantage of using plugins is that we can upgrade each one independently without affecting other running plugins.

The following is an example of how to upgrade an **avx_subsystem_queue** plugin from **v1.0.0** to **v1.0.3**

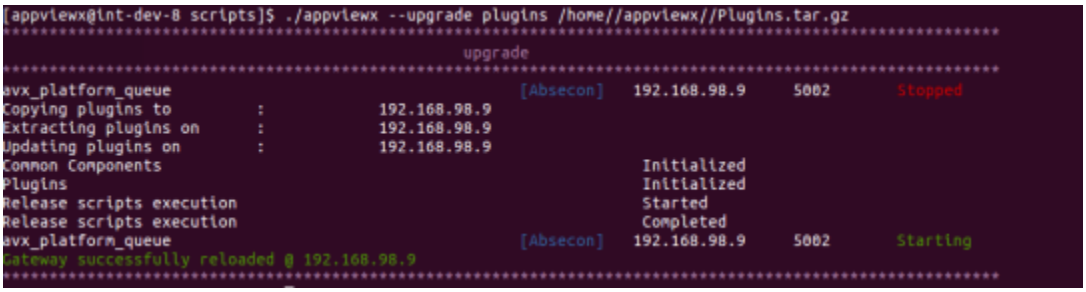
1. Check the status of the plugins that are currently running by entering the following command: appviewx --status plugins avx_platform_queue

```
[appviewx@int-dev-8 scripts]$ ./appviewx --status plugins avx_platform_queue
********************************************************************************************************************
                                        Status
********************************************************************************************************************
avx_platform_queue                          [Absecon]   192.168.98.9     5002       Running
********************************************************************************************************************
```

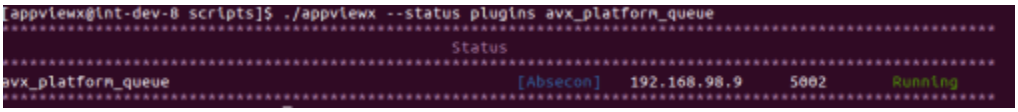
2. Download the plugins you want to install in the following location with the file name Plugins.tar.gz.<user_home_directory>

```
[appviewx@int-31 ~]$ ls
AppViewX  certs  Plugins.tar.gz  upgrade
```

3. Initiate the plugin upgrade process by entering the following command: $ appviewx --upgrade plugins <user_home_directory>/Plugins.tar.gz
   The plugin upgrade starts.




4. After the upgrade process is complete, check the status of the plugins by entering the following command: `appviewx --status plugins avx_platform_queue`



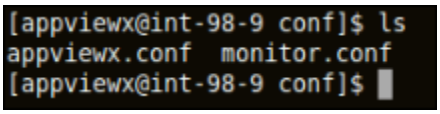

5. Perform a gateway refresh by entering the following command: `$ appviewx --gwrefresh`

## Change an SSH PORT for Device Communication

**Note:** Customizing the SSH port will not change the port on the node, however, will just allow the application to use this specified port.

To customize the SSH port for your environment:

1. Navigate to the following location: **$ cd <avx_installed_path>/conf**
2. Use the `ls` command to verify the existence of the file **appviewx.conf**.
3. Execute the following command to modify the file: `$ vi appviewx.conf`




4. Go to the Device SSH port configuration and change the DEVICE_SSH_PORT to the desired configuration.


5. When you are done editing the fields, press the **Esc** key, then type `:wq` to save and quit the file. `$ appviewx --conf-sync`.

**Note:** In case of multi node, run the following command to update the changes across all the cluster nodes: `$ appviewx --conf-sync`

6. Initialize the SSH configuration change using the following commands:
   `$ appviewx --initialize all $ appviewx --restart plugins`
7. Perform a gateway refresh by entering the following command: `$ appviewx --gwrefresh`

## Update an SSL Configuration for Gateway and Web

If you need to update the SSL certificate for the AppViewX application, complete the following steps:

1. Navigate to the following location: **$ cd <avx_installed_path>/conf**

```
[appviewx@int-98-38 scripts]$ cd /home//appviewx//AppViewX//conf/
```

2. Use the `ls` command to verify the existence of the file **appviewx.conf**.
3. Execute the following command to modify the file: `$ vi appviewx.conf`

```
[appviewx@int-98-38 conf]$ ls
appviewx.conf  monitor.conf
[appviewx@int-98-38 conf]$
```

4. Go to the SSL configuration settings and modify the required parameters. Place the new or updated certificate in the path mentioned in the following configuration file: **<avx_installed_path>/<cert file.p12>**

```
[SSL]
##------------------------------------------------
## Set External_Certificate as True for CA
## Set External_Certificate as FALSE for openssl
##------------------------------------------------
External_Certificate = False
##------------------------------------------------
##
## In case external certificates are provided, the below field
## shuld be given value FQDN or IP depending upon whether the certificate
## was generated based upon hostnames or IPs.
## If the certificate CN is based upon hostname, the value should be FQDN
##              cert_cn = FQDN
## If the certificate CN is based upon IP, the value should be IP
##              cert_cn = IP
## In case an external DNS entry has been provided for a node, give that DNS entry as the value
##      cert_cn = localhost.localdomain.com
## The value need not be changed in case of self-signed cetificates.
##
##------------------------------------------------
CERT_CN = FQDN
##------------------------------------------------
## ssl_web_key should be the path of p12 file
## web_key_password should be the password of the p12 file
##------------------------------------------------
ssl_web_key = /home/appviewx/AppViewX/myserver.p12
web_key_password = appviewx@123
##------------------------------------------------
## ssl_gateway_key should be the path of p12 file
## gateway_key_password should be the password of the p12 file
##------------------------------------------------
ssl_gateway_key = /home/appviewx/AppViewX/myserver.p12
gateway_key_password = appviewx@123
```

**Note:** Self signed certificates are recommended for internal communication with the components (such as mongodb, plugins, and elastic)

5. Update the **CERT_CN** field under the **external_certificates** section in the configuration file as follows:
   - Set **CERT_CN = FQDN**, if the external certificate is based on the hostname.
   - Set **CERT_CN = IP**, if the external certificate is based on the IP.
   - If the certificate was generated based on an external hostname (DNS), then update the external hostname in the **CERT_CN** field.
6. When you are done editing the fields, press the **Esc** key, then type `:wq` to save and quit the file. `$ appviewx --conf-sync`.

**Note:** In case of multi node, run the following command to update the changes across all the cluster nodes: `$ appviewx --conf-sync`

7. To generate a new certificate execute the following command: `$ appviewx --cert-gen`
   After the certificate generates successfully, the following message is displayed:

```
[appviewx@int-dev-8 scripts]$ ./appviewx --cert-gen
**********************************************************************************************************
                                        cert-gen
**********************************************************************************************************
New certificates created.
**********************************************************************************************************
```

8. Navigate to the scripts directory in the installation directory and enter the following commands:

   `$./appviewx --initialize all $./appviewx --restart gateway $./appviewx --restart web`

## Collect Logs from Nodes

This utility can be used to collect the logs of various components from all the nodes as an archive. Use the following commands to collect the logs for all the components, databases, plugins (all or individual), gateway, or web:

```
./appviewx --collect-logs

./appviewx --collect-logs <component>
```

The components are as follows:

- avx_platform_database
- plugins
- avx_platform_gateway
- avx_platform_web
- avx_platform_scheduler

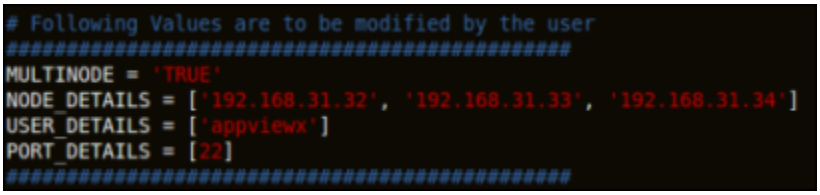
## Copy an SSH Key Across an Installation Node

AppViewX uses password-less authentication using SSH keys between the installation nodes for all communication. In order to update expired SSH keys:

1. Navigate to the following location: **$ cd <avx_installed_path>/scripts/Commons**
2. Execute the following command to modify the file: `$ vi copy_ssh_key.py`

3. Modify the following details:
   - **Node_details**: Provide the IP address of the nodes with comma separation.
   - **User_details**: By default ‘AppViewX’ will be used. Provide a username for each of the nodes with comma separation.
   - **Port_details**: By default port 22 will be used. Provide an ssh port for each of the nodes with comma separation.

4. When you are done editing the fields, press the **Esc** key, then type `:wq` to save and quit the file.
5. Execute the following command to initiate the SSH key exchange between the nodes: `$ <avx_installed_path>/Python/bin/python copy_ssh_key.py`

   A success message is displayed upon completion.

```
[appviewx@avx-31-32 installer]$ python copy_ssh_key.py
password for appviewx@192.168.31.32 :
password for appviewx@192.168.31.33 :
password for appviewx@192.168.31.34 :

 Success. RSA keys are copied to all the servers
```

## Change the Ulimit and Nlimit Configuration in the Node as a Root User

AppViewX always expects AppViewX installed users to have the number of open files (**Nlimit**) and the maximum number of process (**Ulimit**) executable in a Linux machine at a given time to be set to **65535**.

To change the configuration of this setting:

1. Navigate to the following location: **$ cd /etc/security**
2. Execute the following command to modify the file: `$ vi limits.conf`

```
[root@int-98-9 ~]# cd /etc/security/
[root@int-98-9 security]# vi limits.conf
```

3. Update the user details and the configuration to the required input.

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

4. When you are done editing the fields, press the **Esc** key, then type `:wq` to save and quit the file.
5. Open a new shell for the **Ulimit** and **Nlimit** changes to be reflected.

## Change the SSL Configuration

To secure AppViewX communication between Web, Gateway, Plugins, and Elastic, enable HTTPS by completing the following steps:

1. If an external certificate is provided, HTTPS is enabled using the same certificate. By default, the Open SSL self-signed certificate will be used to enable HTTPS and secure communication. Execute the following command to enable HTTPS: `$ appviewx --enable-https all`

   The following options are also allowed:

```
$ appviewx --enable-https avx_platform_web

$ appviewx --enable-https avx_platform_gateway

$ appviewx --enable-https database

$ appviewx --enable-https elastic
```

2. To disable a secure AppViewX communication between Web, Gateway, Plugins, and Elastic, complete the following steps: `$ appviewx --disable-https all`
   The following options are also allowed:

```
$ appviewx --disable-https avx_platform_web

$ appviewx --disable-https avx_platform_gateway

$ appviewx --disable-https plugins

$ appviewx --disable-https elastic
```

## Enable VIP for Web Access

To enable users to access an external VIP to land on the AppViewX web, complete the following steps, which will configure the external VIP configuration to a multi-node installed avx_platform_web plugin.

1. Navigate to the directory **<avx_installed_path>/conf**
2. Go to the **appviewx.conf** file using the following command to add the new plugin to be installed: `$ vi appviewx.conf`
3. Update **WEB_VIP_ENABLED** as **TRUE** and update the external VIP details under **APPVIEWX_WEB_VIP**

```
[WEB]
HOSTS=localhost:5004

##------------------------------------------------
## To enable secure connection in web, set APPVIEWX_WEB_HTTPS as TRUE
##
##------------------------------------------------

APPVIEWX_WEB_HTTPS=False

##------------------------------------------------
## To enable VIP for web, set WEB_VIP_ENABLED as TRUE
##
##------------------------------------------------
WEB_VIP_ENABLED=False
APPVIEWX_WEB_VIP=localhost:5004
APPVIEWX_WEB_VIP_HTTPS=False
```

4. When you are done editing the fields, press the **Esc** key, then type `:wq` to save and quit the file. `$ appviewx --conf-sync`.

> **Note:** In case of multi node, run the following command to update the changes across all the cluster nodes: `$ appviewx --conf-sync`

## Enable a VIP for Gateway Access

To enable external users to access the AppViewX exposed APIs using a VIP, complete the following steps, which will configure the external VIP configuration to a multi-node installed `avx_platform_gateway` plugin.

1. Navigate to the directory `<avx_installed_path>/conf`
2. Go to the `appviewx.conf` file using the following command to add the new plugin to be installed: `$ vi appviewx.conf`
3. Update `GATEWAY_VIP_ENABLED` as `TRUE` and update the external VIP details under `APPVIEWX_GATEWAY_VIP` and `APPVIEWX_GATEWAY_VIP_HTTPS = True` if SSL is enabled for VIP.

```
[GATEWAY]
HOSTS=localhost:5300
APPVIEWX_GATEWAY_KEY=f000ca01

##-----------------------------------------------
## To enable secure connection in gateway, set APPVIEWX_GATEWAY_HTTPS as TRUE
##
##-----------------------------------------------

APPVIEWX_GATEWAY_HTTPS=False

##-----------------------------------------------
## To enable VIP for gateway, set GATEWAY_VIP_ENABLED as TRUE
##
##-----------------------------------------------

GATEWAY_VIP_ENABLED=False
APPVIEWX_GATEWAY_VIP=localhost:5300
APPVIEWX_GATEWAY_VIP_HTTPS=False
```

4. When you are done editing the fields, press the **Esc** key, then type `:wq` to save and quit the file. `$ appviewx --conf-sync`.

> **Note:** In case of multi node, run the following command to update the changes across all the cluster nodes: `$ appviewx --conf-sync`

5. Initialize the VIP configuration change by executing the following commands:

```
$ appviewx --initialize all

$ appviewx --restart all
```

# Reset GUI Admin Password

Use the following command to reset the GUI admin user password: `./appviewx --reset-gui-password`

You must provide the admin database user password for authentication.

# Change the Port for a Plugin After Installation

1. Stop the plugin or component by entering the following: `$./appviewx --stop plugins <plugin_name>`
2. Go the **Installation** directory and in the **conf** directory, open the configuration file **appviewx.conf** by entering the following command: `vi appviewx.conf`
3. Switch to **Insert** mode by pressing the **Insert** key and then change the port in the **PLUGINS** section.
4. When you are done editing the fields, press the **Esc** key, then type `:wq` to save and quit the file. `$ appviewx --conf-sync`.

   **Note:** In case of multi node, run the following command to update the changes across all the cluster nodes: `$ appviewx --conf-sync`

5. Go to the **scripts** directory in the **Installation** directory.
6. Initialize every component by entering the following command: `$./appviewx --initialize all`
7. Start the plugin or component by entering the following: `$./appviewx --start plugins <plugin_name>`
8. After the plugin is in running state, refresh the gateway by entering the following command: `$./appviewx --gwrefresh`

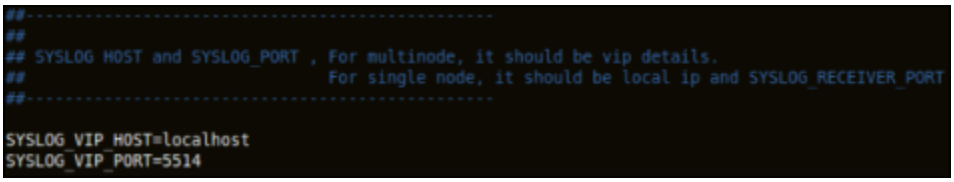
# Enable SYSLOGS Reception from Devices

To enable SYSLOG to be received from the devices added in the inventory, complete the following steps. In case of multi-node, run the following commands to update the changes across all the cluster nodes:

1. Navigate to the directory `<avx_installed_path>/conf`
2. Go to the **appviewx.conf** file using the following command to add the new plugin to be installed: `$ vi appviewx.conf`
3. Update **SYSLOG_RECEIVER_ENABLED** as **TRUE** and hosts where the logstash component has to be installed.

```
[SYSLOG]
HOSTS=localhost:5514
LOG_LEVEL=INFO

SYSLOG_RECEIVER_ENABLED=False
```

4. If SYSLOG reception is enabled using an external VIP, configure the VIP information in the configuration as shown below.




5. To enable Apache Kafka, update the following fields in the **appviewx.conf** file:
   - KAFKA_ENABLED=true
   - KAFKA_HOST
   - KAFKA_PORT
   - KAFKA_TOPIC
   - KAFKA_GROUP_ID
6. Add **plugins avx_platform_syslog** and **avx_platform_syslog_receiver** to the **ENABLED_PLUGINS** list. Add the host details for them, too.
7. When you are done editing the fields, press the **Esc** key, then type `:wq` to save and quit the file. `$ appviewx --conf-sync`.

> **Note:** In case of multi node, run the following command to update the changes across all the cluster nodes: `$ appviewx --conf-sync`

8. Initialize the configuration change using the following command: `$ appviewx --initialize all`
9. To install configuration changes for the log plugin, execute the following command:

```
$ appviewx --start plugins avx_platform_syslog

$ appviewx --start plugins avx_platform_syslog_receiver

$ appviewx --restart avx_platform_logs
```

10. To enable syslog subscription from required vendor plugins, execute the following command: `$ appviewx --restart plugins avx_vendor_f5`
11. Check that the logs and vendor plugins have a status of running by executing the following commands:

```
$ appviewx --status plugins avx_platform_logs

$ appviewx --status plugins avx_vendor_f5
```

12. To reflect the SYSLOG configuration changes in vendor devices, perform a config fetch in the inventory module for the required devices. Any new devices added after enabling this SYSLOG configuration, are automatically registered in the vendor devices.
    The following plugins are associated with an elasticsearch:

- avx_platform_syslog
- avx_platform_syslog_receiver

**Note:** Make sure that these plugins are in an enabled state.

## Execute Command on All Nodes

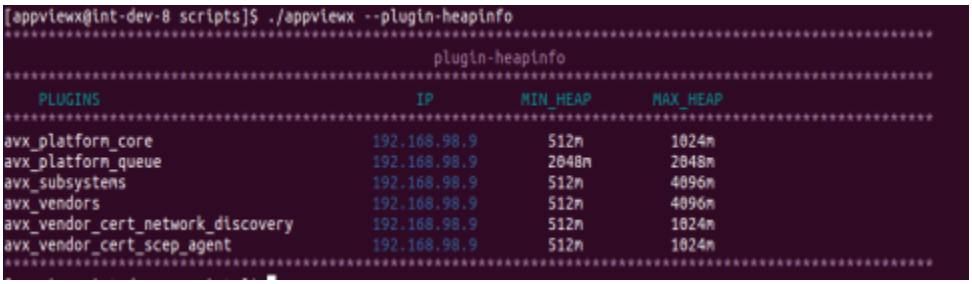
To execute one command on all nodes, use the following command: `appviewx --execute-command`

**Note:**

- Placeholders can be used in **AVX_DIR** for the AppViewX installation directory and **AVX_USER** for the AppViewX user.
- If the command contains `rm` or `mv`, the user will be prompted for confirmation.

## View Heap Size of the Plugins

1. To retrieve the maximum and the minimum heap sizes of the available plugins in all the cluster nodes, execute the following command: `$ appviewx --plugin-heapinfo`

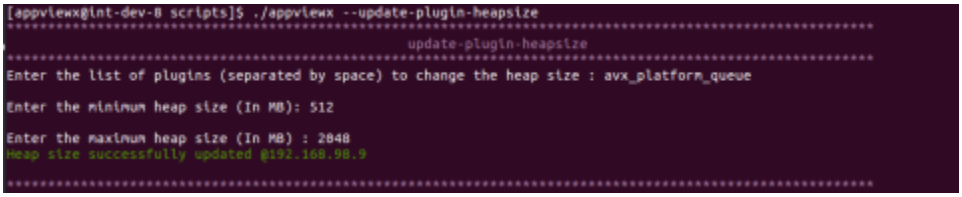



2. To retrieve the maximum and the minimum heap sizes of all the plugins available in a specific node, execute the following command: `$ appviewx --plugin-heapinfo <IP>`
3. To retrieve the maximum and the minimum heap sizes of a particular plugin in all the cluster nodes, execute the following command: `$ appviewx --plugin-heapinfo <plugin_name>`
4. Run the following command: `appviewx --gateway-refresh`

## Update the Heap Size of the Plugins

**Note:** Tab completion is supported for all user inputs.

1. Execute the following command to update the head size of the plugins: `appviewx --update-plugin-heapsize`




You will be prompted to provide the following details:

- **Plugins**: The plugin for which you want to update the heap size. When multiple plugins are provided, ensure that they are space-separated.
- **Minimum heap size**: The minimum heap size should be provided in MB. For example, 512 MB.
- **Maximum heap size**: The maximum heap size should be provided in MB. For example, 1024 MB
- **Node IP(s)**: The IP address of the node, where you want the change to be reflected. When multiple node IP(s) are provided as input, ensure that they are space-separated.

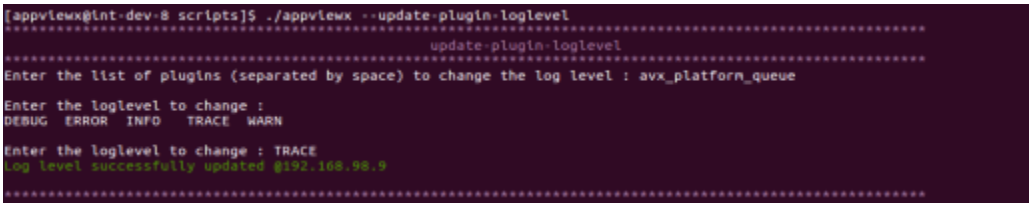
> **Note:** The Node IP details are required only in a multi-node scenario.

2. Execute the following command to restart the plugins: `$ appviewx --restart plugins <plugin_name>`

## Update Log Level of the Plugins

> **Note:** Tab completion is supported for all user inputs.

1. Execute the following command to update the log level of the plugins: `appviewx --update-plugin-loglevel`




Youwill be prompted to provide the following details:

- **Plugins**: The plugin for which you want to update the log level. When multiple plugins are provided, ensure that they are space-separated.
- **Log Level**: The following are the log level values
  - TRACE
  - DEBUG
  - INFO

- WARN
- ERROR

- **Node IP(s)**: The IP address of the node, where you want the change to be reflected. When multiple node IP(s) are provided as input, ensure that they are space-separated.

> **Note:** The Node IP details are required only in a multi-node scenario.

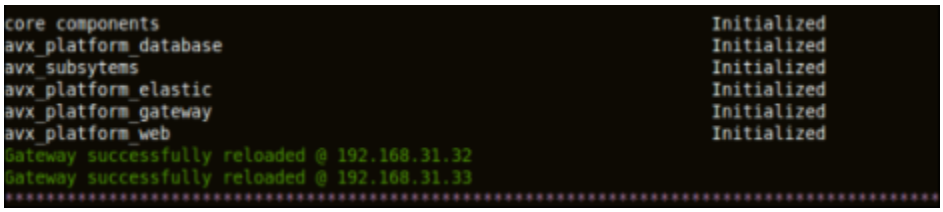
2. Execute the following command to restart the plugins: `$ appviewx --restart plugins <plugin_name>`
3. Run the following command: `appviewx --gateway-refresh`

## Configure an Elasticsearch

1. Navigate to the directory **<avx_installed_directory>/conf**
2. Open the configuration file **appviewx.conf** by entering the following command: `vi appviewx.conf`
3. Press the **Insert** key to switch to the insert mode, then set the value for the **ENABLE** field under **ELASTIC** section as follows:
   - **ENABLE=TRUE** to enable elasticsearch
   - **ENABLE=FALSE** to disable elasticsearch
4. If the elastic search is enabled in the **appviewx conf** file, make sure that the following details are updated:
   - **HOSTS:** The input should be in the following format: <IP1>:<PORT>, <IP2>:<PORT>, <IP3>:<PORT>
   - **TRANSPORT_PORT:** By default, the port **5550** is used for configuration. You can change the port if necessary. Only one port should be provided to use as the transport port for elastic configured nodes.
   - **ELASTIC_HTTPS:**Set the value as follows:
     - **ELASTIC_HTTPS=FALSE** to disable secured communication
     - **ELASTIC_HTTPS=TRUE** to enable secured communication
5. If the **ELASTIC_HTTPS** field in the **appviewx.conf** file is set to TRUE, update the following fields:
   - **EXTERNAL_CERTIFICATE=TRUE** to use the external certificate for secured communication.
   - **EXTERNAL_CERTIFICATE=FALSE** to use the self-signed certificate for secured communication.
6. When you are done editing the fields, press the **ESC** key, then type `:wq` to save and quit the file. `$ appviewx --conf-sync`.

> **Note:** In case of multi node, run the following command to update the changes across all the cluster nodes: `$ appviewx --conf-sync`

7. Initialize the configuration change by entering the following commands:
   - `appviewx --initialize all`




   - `appviewx --restart avx_platform_elastic`


The following plugins are associated with elasticsearch:

- **avx_insight_subsystem_adc**
- **avx_insight_statistics_bot**

**Note:** These plugins will start only when the elasticsearch is up and running. The plugin **avx_insight_statistics_bot** will be enabled by default and it will initiate the statistics collection. If you want to collect the statistics through the **plugin avx_insight_vendor**, then you must disable the plugin **avx_insight_statistics_bot** and **enable avx_insight_vendor**.

## Modify an Elasticsearch

Elasticsearch is configured within 5 shards and 1 replica by default. In the case of multinode, data loss may occur when two elasticsearch nodes are down and not available.

For example, if elasticsearch is configured in 3 nodes execute the following command to prevent the data loss:`curl -k -X PUT https://admin:admin@<Elastic_ip:Elastic_port>/_template/template_1 -d '{"template":"*","index_patterns": ["statistic*"],"settings":{"number_of_replicas": 2}}'`

This will configure the replicas in the elasticsearch servers and will prevent the data loss.

**Note:** The disadvantage of the replica is that it will consume large disk spaces.

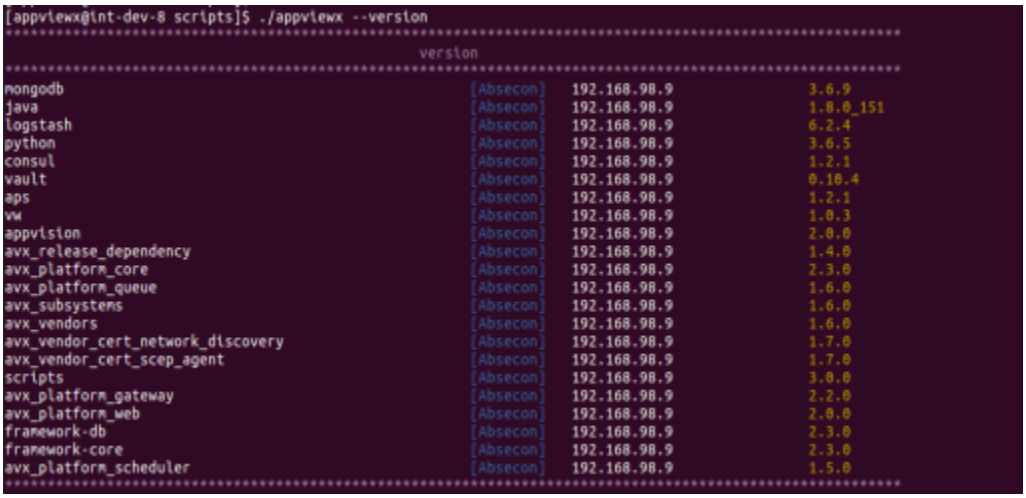
## Backup and Restore of an Elasticsearch

1. Execute the following command to take a backup of the elasticsearch data: `appviewx --elastic-backup`
2. The backup file will be available in the following location: `<appviewx installed directory>/es_backup`
3. You can maintain a maximum of 5 backup files.

4. Execute the following command to restore the elasticsearch data./appviewx --elastic-restore.
5. You will be prompted with the list of backup files.
6. Select the backup file which you want to restore.

## View the Version of AppViewX Components

1. Execute the following command to retrieve the version of all the components: `appviewx --version`

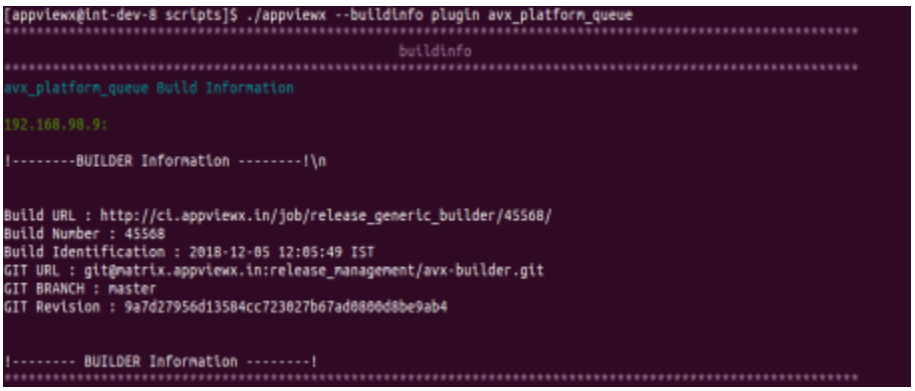



2. Execute the following command to retrieve the version of AppViewX components: `appviewx --version avx-components`
3. Execute the following command to retrieve the version of third party components: `appviewx --version third-party`
4. Execute the following command to retrieve the build information of AppViewX components: `appviewx --buildinfo <avx_components>`
5. Execute the following command to retrieve the version of the AppViewX plugins: `appviewx --buildinfo plugin <plugin>`

   The following are the allowed components:

   - avx_platform_gateway
   - avx_platform_web
   - Scripts
   - avx_platform_scheduler
   - framework-db
   - framework-core

- APS
- avx_release_dependency




## Set the Location for Database Backup

1. Go to the directory: **<avx_installed_directory>/conf**
2. Open the configuration file **appviewx.conf** by entering the following command: `vi appviewx.conf`
3. To store the DB backup in an FTP server, set the value for **ENABLE_FTP_UPLOAD** to **TRUE**.
4. Update the following details of the FTP server in the configuration file:
   - **FTP_USERNAME**
   - **FTP_PASSWORD**
   - **FTP_SERVER**
   - **FTP_PORT**
   - **FTP_REMOTE_DIR**
   - **PROTOCOL=sftp/scp**
   - If **FTP_ENABLED** is set to **TRUE**, the backup will be stored in the configured FTP location.
   - If the **FTP_ENABLED** is set to **FALSE**, the backup will be stored in the **<appviewx installed directory>/dbbackup**
5. When you are done editing the fields, press the **ESC** key, then type `:wq` to save and quit the file.`$ appviewx --conf-sync`.

> **Note:** In case of multi node, run the following command to update the changes across all the cluster nodes: `$ appviewx --conf-sync`

## Configure a TFTP Server

1. Go to the directory **<avx_installed_directory>/conf**
2. Open the configuration file **appviewx.conf** by entering the following command: `vi appviewx.conf`

3. Update the following details of the FTP server in the configuration file:
   - TFTP_SERVER_IP
   - TFTP_SERVER_USER_NAME
   - TFTP_SERVER_DOWNLOAD_PATH
   - TFTP_SERVER_PASSWORD
   - TFTP_SERVER_PASSWORD_KEY

> **Note:** An encrypted password and password key needs to be generated for the TFTP_SERVER_PASSWORD and TFTP_SERVER_PASSWORD_KEY

4. To generate an encrypted password and password key, complete the following steps:

   a. Execute the command `appviewx --password-encrypt`.
   You will be prompted to enter the server password.

   b. Type the password and press **Enter** on your keyboard.
   This will display the encrypted password and password key.

## Configure the Test Data

1. Go to the directory **<avx_installed_directory>/appviewx/scripts**
2. Run the following command:
   `appviewx --feed-elastic-data <type-of-data> <value-of-data>`

> **Note:** The value has to be less than or equal to 60 and should be a multiple of 5 in case of raw data.

For example: `appviewx --feed-elastic-data raw 10 appviewx --feed-elastic-data aggregate 90`

> **Note:**
> To support this feature ensure that the `avx_platform_elastic` plugin is enabled.

## Configure an SSL for the Database

> **Note:** In AppViewX v12.4.0, MongoDB has been upgraded from v3.0.7 to v3.6.2. There is a provision to configure secure communication(SSL) in the updated version and by default, it will be enabled for the database.

To enable or disable SSL for the database:

1. Go to the directory **<avx_installed_directory>/conf** .
2. Open the configuration file **appviewx.conf** by entering the following command: `vi appviewx.conf`

> **Note:** The MongoDB storage engine will be available in wiredtiger. (In the older version, it was available in MMAP).

3. To disable, set **ENABLE_SSL=False** in the **appviewx.conf** file.
4. To enable, set **ENABLE_SSL=True** in the **appviewx.conf** file.
5. Execute the following commands:
   - `appviewx -cert-gen`
   - `appviewx --initialize all`
   - `appviewx --restart all`

## Configure a Fat JAR Deployment

1. Go to the directory **<avx_installed_directory>/conf**
2. Open the configuration file **appviewx.conf** by entering the following command: `vi appviewx.conf`
   The available modules are **ADC**, **AUTOMATION**, **CERT**, **SECURITY**, and **OTHERS**.
3. To enable all the required modules, modify the **MODULES_ENABLED** field. For example, **MODULES_ENABLED=ADC**.

> **Note:** By default, all the modules will be enabled on Fat JAR deployment.

4. To enable a specific plugin from a module, modify the **VM_ENABLED** field. For example, **VM_ENABLED= avx_subsystem_adc, avx_vendor_f5**

> **Note:** If the **VM_ENABLED** field is empty, it will automatically consider all the plugins from the enabled modules. You can find the details of the available modules and the associated plugins from the `<avx_installed_directory>/scripts/templates/Plugins/vm_modules_txt` file.

## Change the Database Password

1. Execute the following command: `appviewx --change-db-password`.

   **Note:** You will require the database admin password to change the database password.

   You will be prompted to enter a user name and the new password.
2. Type the password and restart all the components by entering the following command: `appviewx --restart all`

   **Note:** You can skip this step while updating the database password for an admin user.

## Monitor the VIP Status

- After configuring the VIP(s) for **avx_platform_web** or **avx_platform_logs**, check if the call is routed to the server where the components are up and running.
- The monitoring endpoints on the servers are provided for the VIP(s) to determine whether or not the call should be routed to the server. The following command will be executed by the client on the device during the VIP configuration: `<host>:<web_port>/appviewx/VipRoutingStatus?component=<component>`
  - For example:

    https://192.168.98.23:5004/appviewx/VipRoutingStatus?component=avx_platform_web

    https://192.168.98.23:5004/appviewx/VipRoutingStatus?component=avx_platform_logs
  - The call will be successfully routed only if the endpoint response is up.

## Configure an AppViewX Git

AppViewX Git allows the user to get the latest checkouts of various projects from Github. After a new deployment/migration is completed:

1. Run the following command to set up the functionality: `appviewx --git`
2. To configure the projects, open the **<appviewx_dir>/appviewx_git/git_config.py** file and configure as follows:

```
GIT_REPO_URLS = [


{
```

```
"REPO_NAME": "Load-Balancers",

"REPO_URL": "https://github.com/AppViewX/Load-Balancers",

"REPO_TOKEN": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"

},

{

"REPO_NAME": "Certificates",

"REPO_URL": "https://github.com/AppViewX/Certificates",

"REPO_TOKEN": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"

}

]
```

3. To get the **REPO_TOKEN** for projects, please contact support@appviewx.com
4. To fetch the latest checkout of the projects mentioned in the config file, run the following command: `appviewx --git`

   The server where this command has been executed must have an access to connect to the github server either through a direct internet connection or through a proxy.

## Configure a CyberArk Agent

1. Execute the following command to move the **RHELinuxx64-Rls-v9.9.zip** package to **/root** in the AppViewX Server: `mv RHELinux x64-Rls-v9.9.zip /root`
2. Untar the package by entering the following command: `unzip RHELinux x64-Rls-v9.9.zip`
3. Rename the package by entering the following command: **mv RHELinux x64 RHELinux_64**
4. Navigate to the **RHELinux_64** directory and execute the following commands to change the Read/ Write/Execute permissions for the **CreateCredFile**:

```
cd RHELinux_64/

chmod 755 CreateCredFile
```

5. Enter the following command to provide the username and its associated password for the **CyberArk Vault Administrator:** `./CreateCredFile administrator.cf Password`

   **Note:** Leave the remaining parameters set to the default value.

6. Modify the following fields in the **/root/RHELinux_64/aimparms.sample** file by entering the command `vi aimparms.sample`:
   - AcceptCyberArkEULA=Yes
   - CreateVaultEnvironment=yes
   - LicensedProducts=AIM
   - CredFilePath=/root/RHELinux_64/administrator.cf
   - VaultFilePath=/root/RHELinux_64/Vault.ini
7. Click **Save**.
8. Copy the **aimparms.sample** file to **/var/tmp/aimparms** by entering the following command: `cp aimparms.sample /var/tmp/aimparms`
9. Modify the following fields in the **/root/RHELinux_64/Vault.ini** file by entering the command `vi Vault.ini`
   - VAULT = <"Vault name">
   - ADDRESS=<CyberArk vault address>
   - PORT=<CyberArk vault listening port>

   **Note:** These details must be fetched from the CyberArk team.

   **Note:** Before installation, ensure that the RPM package (such as the IP address and Port number) must be reachable from Agent Server.

10. Click **Save**.
11. Install the RPM package **CARKaim-9.80.0.85.x86_64.rpm** by entering the following command: `rpm -i CARKaim-9.80.0.85.x86_64.rpm`
12. After installing the RPM package, execute the following command to check the service status: `service aimprv status`.

    The response is displayed as **Cyber-Ark Application Password Provider is running**

After the service is up and running, it allows the agent server hostname as a member for all the safes on the CyberArk component, for which the credentials have to be retrieved from the vault.

## Configure a Proxy

To configure a proxy for Amazon Web Services (AWS) and iHealth, complete the following steps:

1. Enable a proxy by updating the following field in the **appviewx.conf** file: **IHEALTH_PROXY=TRUE**
2. Run the following commands:

```
appviewx --initialize all

appviewx --restart all
```

**Note:** This allows the AWS and iHealth feature to function through a proxy server.
For detailed information on how to configure a proxy, refer to the AppViewX 2019.4.0 User guide.

## Update the Node Password

The node password that has been changed must be updated in the configuration file for the HAProxy and NGinX vendors to be managed at AppViewX.

To update the node password for the HAProxy and NGinX device:

1. Execute the following command: `appviewx --update-node-password`
   You will be prompted to provide the changed password.
2. Type the password and press **Enter**.
3. Restart all the components by entering the following command: `appviewx --restart all`

## Reverse DNS lookup

1. Open the **appviewx.conf** file by entering the following command: `vi appviewx.conf`
2. Set the **REVERSE_LOOKUP_TRIGGER** property to **True**.
3. Run the following commands:

```
appviewx --initialize all

appviewx --restart all
```

## Syslogs

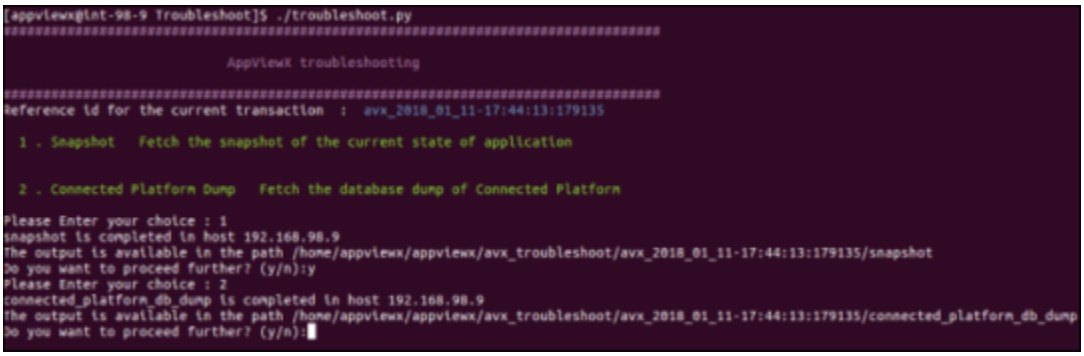
To receive the pending Syslogs from the **A10 v4.0.1** device, you must do the port forwarding from **514** to **5514** in the AppViewX node.

# Troubleshooting Utility

1. Navigate to the directory **<avx_installed_directory>/appviewx/scripts**
2. Run the following script: `./troubleshoot.py`

   A prompt will appear with the following options:

   - **Snapshots:** To fetch and capture the current state of an application. After the snapshot is captured, the location of the snapshot will be displayed.
   - **Connected platform dump:** To fetch the database dump of the Connected platform. After it is retrieved, the location of the database dump will be displayed.




# Prerequisites for SSH Deployment on CentOS 7

- The following utilities must be available:
  - OpenSSH v6.6.1p1
  - OpenSSL v1.0.1e-fips
  - PuTTYgen v0.70
    - This utility must have been installed on the server to download **.PPK** file of the private key.
    - For example, **TLS_CACERT /etc/certs/rootCA.cer**
- The attribute **sshPublicKey** must be available in the user profile of the customer's Active Directory to enable the **Publish to LDAP** feature to work.
- By default, the users must set their account in the /bin/bash shell to manage the keys.
- To use the public keys from the attribute sshPublicKeyin the user profile of the LDAP directory, ensure that the custom scripts are available and updated on the users' authorized_keys file.
- By default, the following basic utilities must be available in the client machines:
  - base64 (GNU coreutils) 8.22
  - xargs (GNU findutils) 4.5.11
  - grep (GNU grep) 2.16
  - sha256sum (GNU coreutils) 8.21
- If you want the functionalities to fetch the credential type **sudoer** and manage the hosts in AppViewX, the System Administrator or UNIX Server Support team must ensure that the sudoer account does not

have any restrictions (such as read and write, setting ownership, and file permissions) in the sudoers file.

- The F5 devices are capable to support the update known_host file only if they are added to AppViewX using the root/sudoer privileged credentials.

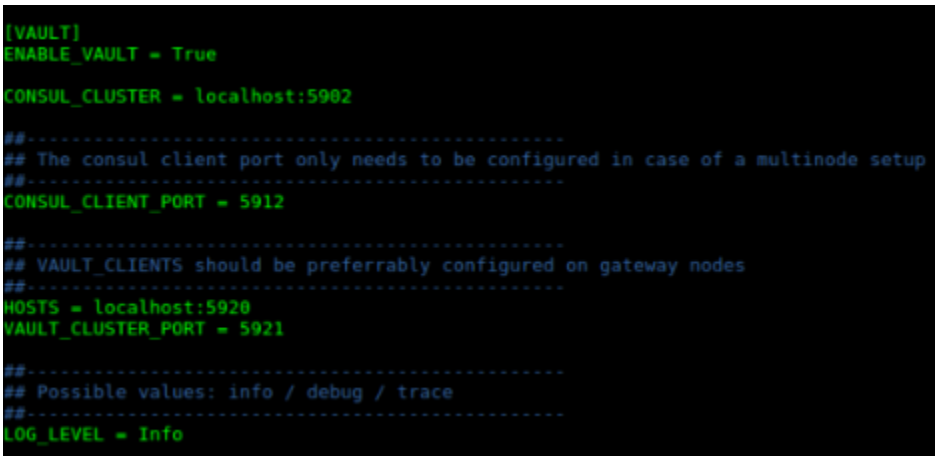
## Enable the Consul and Vault

The Consul and vault are enabled to store and manage the data in a secure manner. The Consul must have been configured in an odd number of nodes and the Vault must have been configured in the node where a consul is configured. Also, it is not mandatory to configure a vault in the database nodes.

> **Tip:** It is recommended to configure the vault in two nodes and the consul in three nodes for a multinode.

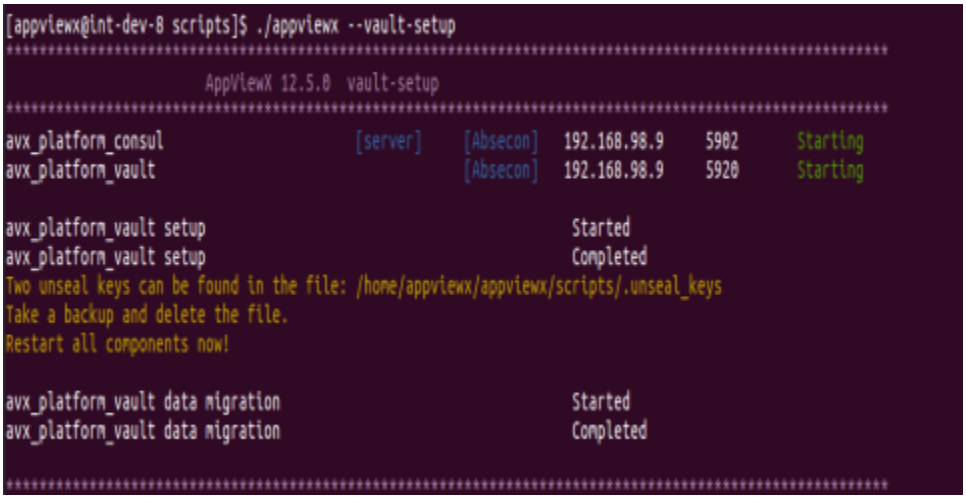
To enable the consul and vault:

1. Go to the **<avx_installed_path>/conf** directory.
2. Go to the **appviewx.conf** file.
3. Execute the following command to enable the consul and vault: `$ vi appviewx.conf`
4. Set the **ENABLE_VAULT** field in the **VAULT** section to **TRUE** and update the following fields to install the consul and vault components:
   - **CONSUL_CLUSTER**
   - **CONSUL_CLIENT_PORT**
   - **HOSTS**
   - **VAULT_CLUSTER_PORT**
   - **LOG_LEVEL**

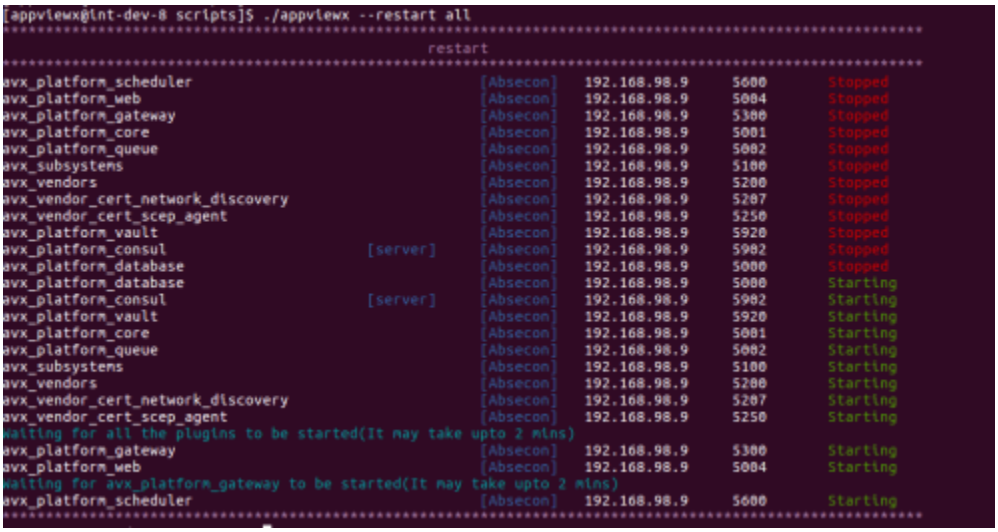


5. When you are done editing the fields, press the **ESC** key on your keyboard, then type `:wq` to save and quit the file.

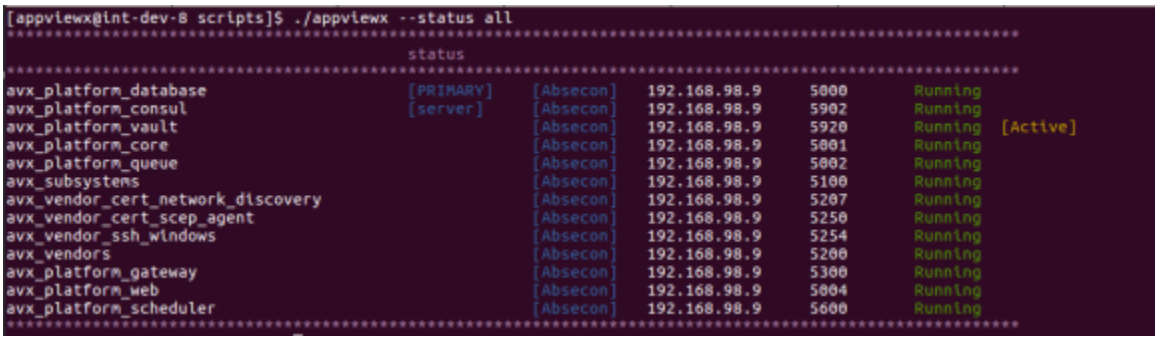
   In case of multinode, run the command `$ appviewx --conf-sync` to update the changes across all the cluster nodes.
6. Execute the following command to initialize the configuration changes: `$ appviewx --initialize all`
7. Execute the following command to set up and initialize the vault: `$ appviewx --vault-setup`




8. Restart all the components to start with the vault configurations by executing the following command: `$ appviewx --restart all`




9. Check the status of the components by executing the following command: `$ appviewx --status all`




**Note:** Ensure that the vault is not in a disabled state after you enable it.

# Restore a Database

- From a vault-enabled instance to another vault-enabled instance
- From a non-vault instance to a vault-enabled instance
- From a non-vault instance to another non-vault instance
- From a vault-enabled instance to a non-vault enabled instance

## From a vault-enabled instance to another vault-enabled instance

1. Execute the following commands on the instance from where the data must be taken:

```
appviewx --databasebackup

appviewx --vault-backup
```

2. Copy the latest archives from the following directories to the second instance:
   - **<appviewx_installed_dir>/db_backup**
   - **<appviewx_installed_dir>/vault_backup**
3. Execute the following commands on the second instance:

```
appviewx --databaserestore <absolute path of database backup file>

appviewx --vault-restore <absolute path of vault backup file>
```

4. Execute the following command to restart all the components on the second instance: `appviewx --restart all`

### From a non-vault instance to a vault-enabled instance

1. Execute the following command on the non-vault enabled instance: `appviewx --databasebackup`
2. Copy the latest archives from the following directories to the second instance: **<appviewx_installed_dir>/db_backup**
3. Execute the following command on the second instance: `appviewx --databaserestore <absolute path of database backup file>`
4. Execute the following command on the second instance to migrate the data to a vault enabled instance: `appviewx --vault-migration-jar`

### From a non-vault instance to another non-vault instance

1. Execute the following command on the first instance: `appviewx --databasebackup`
2. Copy the latest archives from the following directories to the second instance: **<appviewx_installed_dir>/db_backup**
3. Execute the following command on the second instance: `appviewx --databaserestore <absolute path of database backup file>`

### From a vault-enabled instance to a non-vault enabled instance

The data from a vault enabled instance cannot be restored to a non-vault enabled instance. However, you can enable a vault on the second instance and then, restore the data.

- For detailed information on how to enable a vault, refer to the Enable the Consul and Vault section of this guide.
- Repeat the steps that are mentioned in the From a vault enabled instance to another vault enabled instance section of this guide.

## Get the Certificate Information

Execute the commands below to get the certificate details (**Command Name (CN)**, **Subject Alternative Name (SAN)**, and **Certificate Authority (CA)**) of the following components:

- MongoDB
- Vault
- Plugins
- Gateway

- Web
- Elastic

**Commands**

- `./appviewx --cert-info`

  This command displays the certificate information of all the components.

- `./appviewx --cert-info <component>`

  This command displays the certificate information of the specific component.

- `./appviewx --cert-info <ip>`

  This command displays the certificate information of all the components on a specific node.

## Generate a New Certificate for the SSL Components

To generate and associate a new certificate with the SSL components:

1. Stop all the components by executing the following command: `appviewx --stop all`
2. Generate a new certificate by executing the following command: `appviewx --cert-gen`
3. Associate the certificate with the SSL components by executing the following command: `appviewx -initialize all`
4. Start all the components by executing the following command: `appviewx --start all`

## Windows Gateway Installer

AppViewX is a Linux based product and an intermediate agent is required to establish communication with any Windows machines to access functionalities such as discovering and pushing certificate to Windows Store, JKS for discovering, requesting a new certificate, renewal, and revocation of a certificate from the Microsoft CA.

- Prerequisites
- Current Implementation
- New Implementation - .exe file
- Steps to Integrate with AppViewX
- Agent Setup When the Service Account is not a Part of the Administrator Group
- Configuration Settings File

- LogOn Application
- Push Agent

## Prerequisites

For the client certificate authentication to work, the root and Intermediate certificates must not be mixed up. Such certificates must be deleted from the store. Also, the user performing the installation should have admin access.

| Component | Machine | Description | Scripts |
|---|---|---|---|
| .Net Framework 4.5 and above | Source and Target | Please download and install the framework from the URL https://www.microsoft.com/net/download/dotnet-framework-runtime | C:WindowsMicrosoft.NETFrameworkv4.0.30319>.MSBuild.exe -version |
| POWERSHELL 4+ | Source and Target | • PowerShell 4.0 installation URL: https://www.microsoft.com/en-in/download/details.aspx?id=42554<br>• Powershell IIS installation URL: x64 - https://www.microsoft.com/en-in/download/details.aspx?id=15488, x86 https://www.microsoft.com/en-in/download/details.aspx?id=7436 | Powershell $PSVersionTable.<br>• PSVersion IIS7.5 and above Get-Module -ListAvailable should display the WebAdministration<br>• IIS7 - Get-PSSnapin -Registered should display the WebAdministration |
| certadm.dll | Source and Target (Only CA) | Check if dll is available in the C:WindowsSystem32 folder or install the Microsoft Remote | cd C:WindowsSystem32 and then dir certadm.dll |

| Component | Machine | Description | Scripts |
|---|---|---|---|
| | | Server Administration Tools (RSAT) for the respective OS. For example, https://www.microsoft.com/en-in/download/details.aspx?id=45520 for Windows 10 | |
| CertUtil | Source and Target (Only CA) | Copy to the System32 folder if it is not available | Run certutil in the command prompt |
| netsh | Source | Copy to the System32 folder if it is not available | Run netsh in the command prompt |
| RPC | Source and Target | Start the Remote procedure call in the services | net start RpcSs |
| WMI | Source and Target | Start the Windows Management Instrumentation in the services | net start Winmgmt |
| WinRM | Source and Target | Start the Windows Remote Management | net start WinRM |
| IIS | Target | | powershell "get-itemproperty HKLM:SOFTWAREMicrosoftInetStp \| select setupstring,versionstring" |
| User Permission | Target | When the users are added in the Group and the machine is not restarted a permission error will occur. Ensure | • Gwmi win32_groupuser -computer ptpll594\|? {$_.groupcomponent -like '*""Administrators""'} \| select PartComponent<br>• net localgroup administrators |

| Component | Machine | Description | Scripts |
|---|---|---|---|
| | | that the machine is restarted when the user is added to a group | • Check if user can access C$/windows/temp or admin$/Temp<br>• Local admin addition needs restart |
| File Operations | Target | | Check if user can access C$/windows/temp or admin$/temp. If you do not have c-drive then change the configuration to the available drive |
| Port | Source | Check if the port is already in use | • netstat -an \|find ""8999"<br>• Check the Firewall outbound rules for the port<br>• Ping test from AppViewX<br>• Antivirus block for the port<br>• Turn off the local firewall<br>• Check the server, client, root, and intermediate certificates<br>• Check if the C:Logs folder exists and the permissions<br>• If you check in the Internet Explorer then the enhanced security must be disabled in the server role localserver |
| Powershell Remoting | Target | | Enter-PSSession -ComputerName <computername> -Credential <username> |

The users configured in AppViewx or agent must have local administrator rights on the target machine. To do that, complete the following steps:

1. Search **Edit local Users and groups** through **Start** > **Run** command and click on **Groups**.
2. Click on **Administrators** and add the user to the administrator's group if it's not displayed in the administrator's group.

## Current Implementation

To communicate with a Windows machine or a Microsoft CA deployed in a Windows machine, AppViewX requires a Jump Box agent that must be deployed in an IIS server, which will enable communication with the other windows machines that are in the same domain. This implementation has the following drawbacks:

- An IIS server and a dedicated IIS website are required to deploy this agent. This brings up a problem of handling Windows devices where IIS is not available.
- Any changes in the agent's dynamic link library (DLL) file will require replacing it in the respective websites.

## New Implementation - .exe file

The DLL files of the agent are wrapped into an .exe file, which can be installed in any Windows machine without any dependency on the IIS server. This enables a service in the Windows machine to communicate with another Windows machine or a Microsoft CA deployed in it.

## Steps to Integrate with AppViewX

**Note:** By default, the AppViewX gateway installers (in .msi format) and the server/client certificates are shipped along with AppViewX.

1. In the Windows machine, download one of the following AppViewX gateway installers and complete the corresponding steps:
   - AppViewX.CertPlus.Setup.msi
   - AppViewX.CertPlus.ValidatorSetup.msi
2. Download the certificates and save them in the same folder as the setup files.

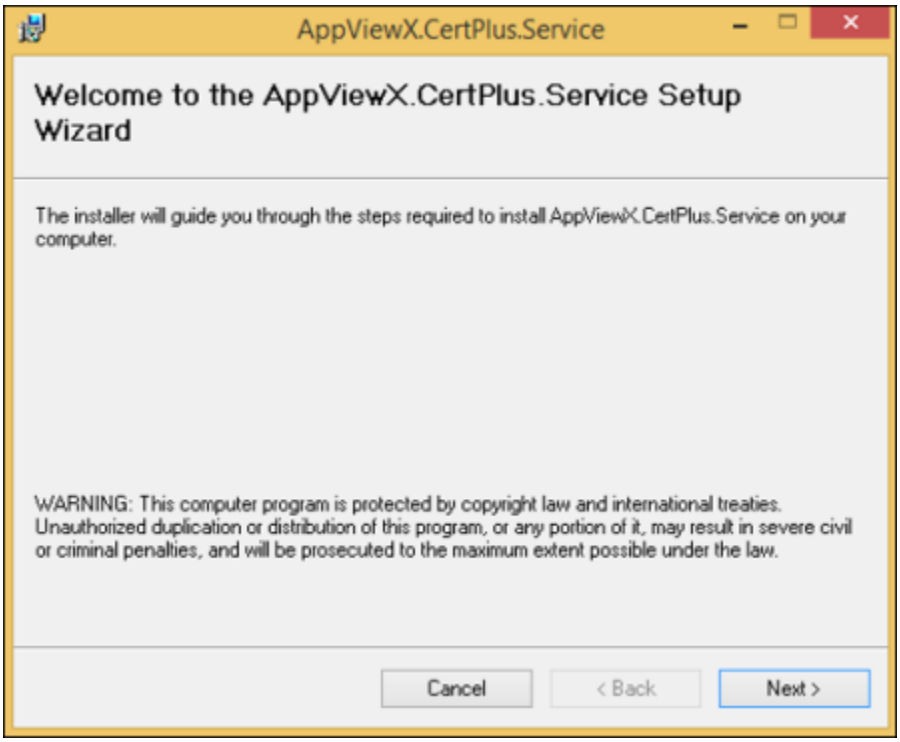
**Note:**

- If the user wants to use a different server and client certificate then replace the certificates in the same folder as the setup files.
- The setup file will add the certificates that are copied in the same directory to the store. Ensure that the name of the server and client certificate is in the ServerCertificateGateway.pfx and ClientCertificateGateway.pfx format.
- If the certificate is replaced, ensure that the respective password has been provided to add the certificate to the store. The incorrect password will cause the Agent to fail.

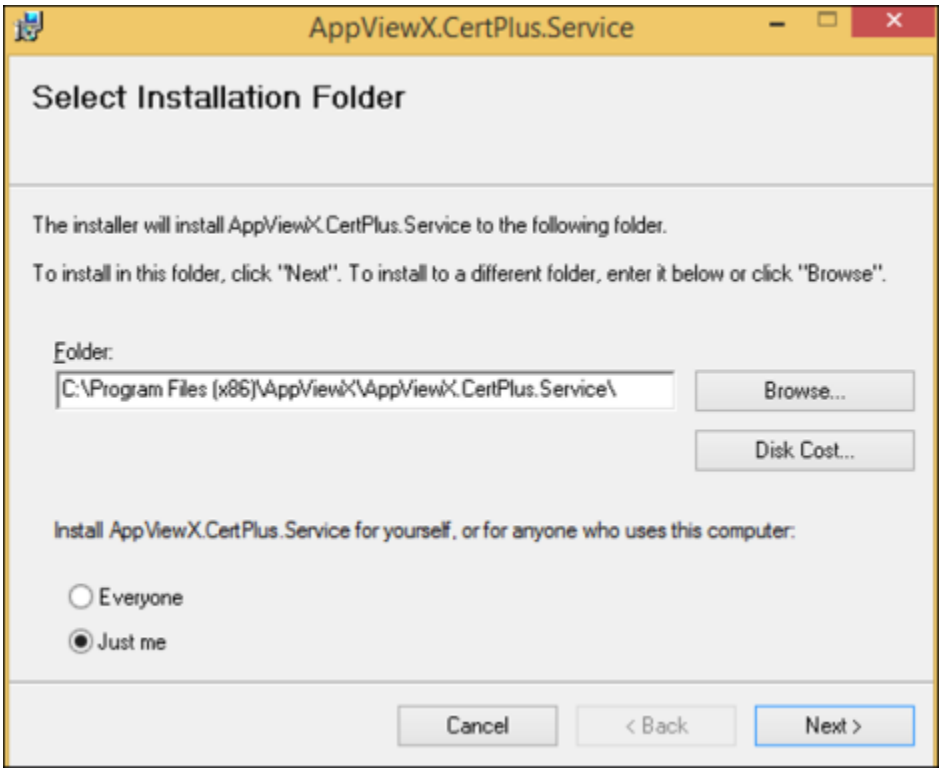
- AppViewX.CertPlus.Setup.msi
- AppViewX.CertPlus.CustomSetup.msi

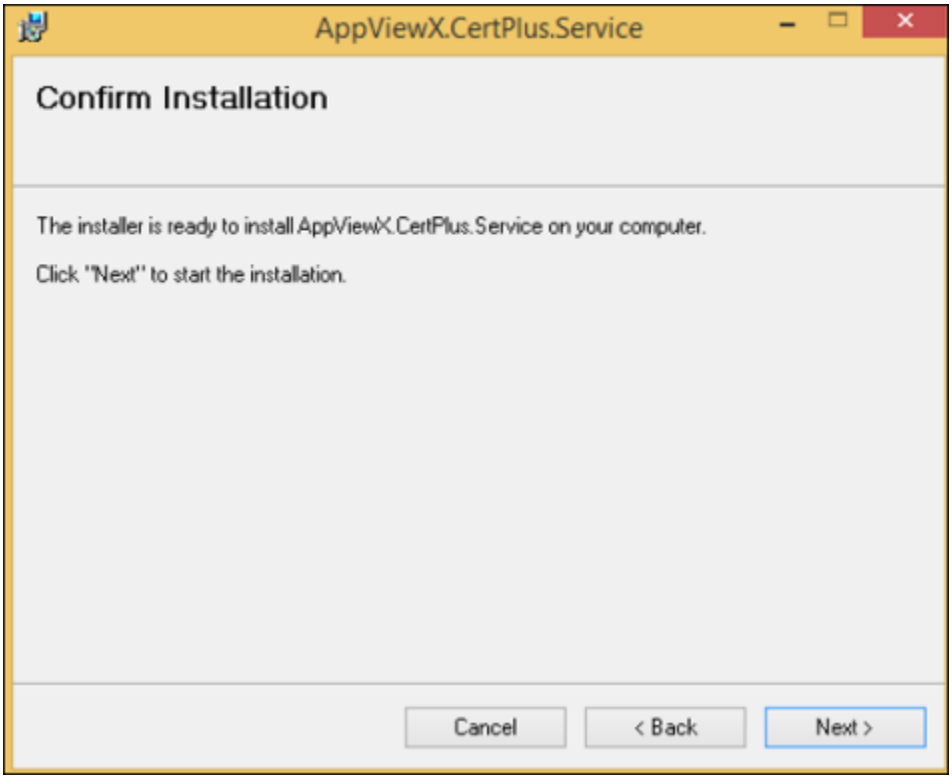
## AppViewX.CertPlus.Setup.msi

1. Click the **AppViewX.CertPlus.Setup.msi** file to install the AppViewX Windows Gateway.



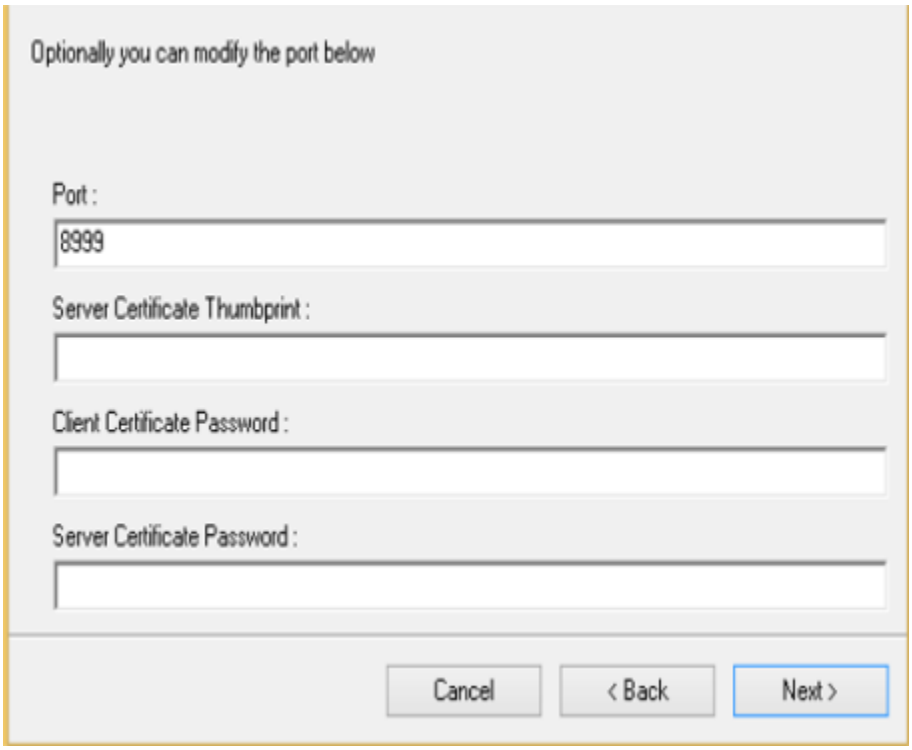

2. Select the users for the service by selecting **Everyone** if the service must be used by other user accounts such as an account apart from the login account used for installing the service.



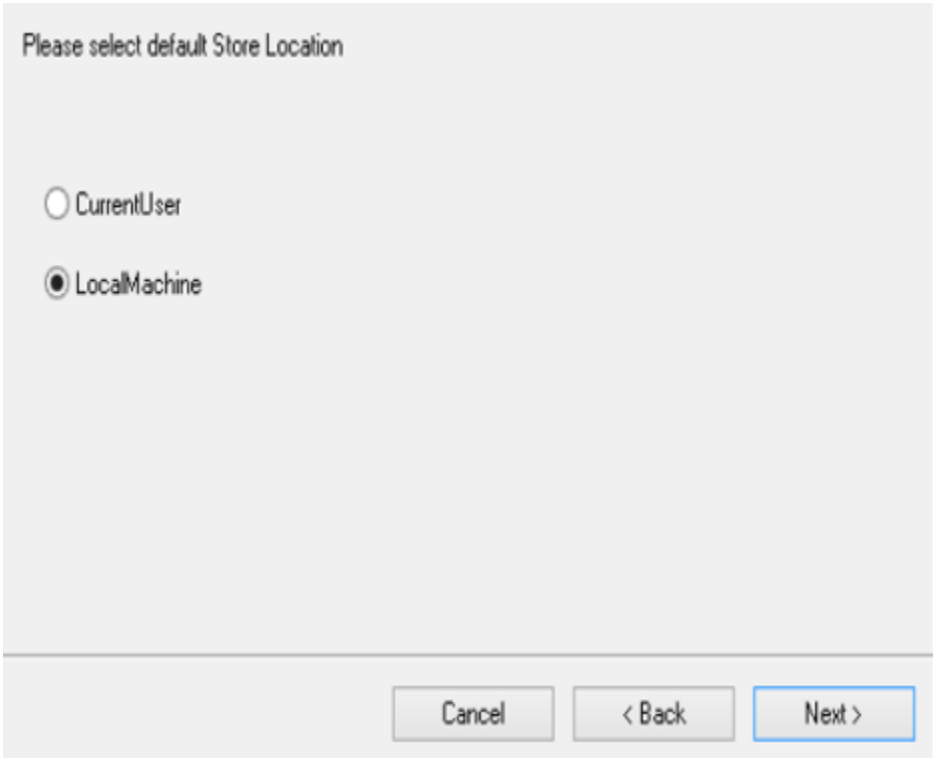

3. Enter a custom port for accessing the service.

The default value is 8999, which can be modified if required. By default, the **Thumbprint** value is the certificate shared with the installer. Enter a custom "Thumbprint" value while using a custom certificate.

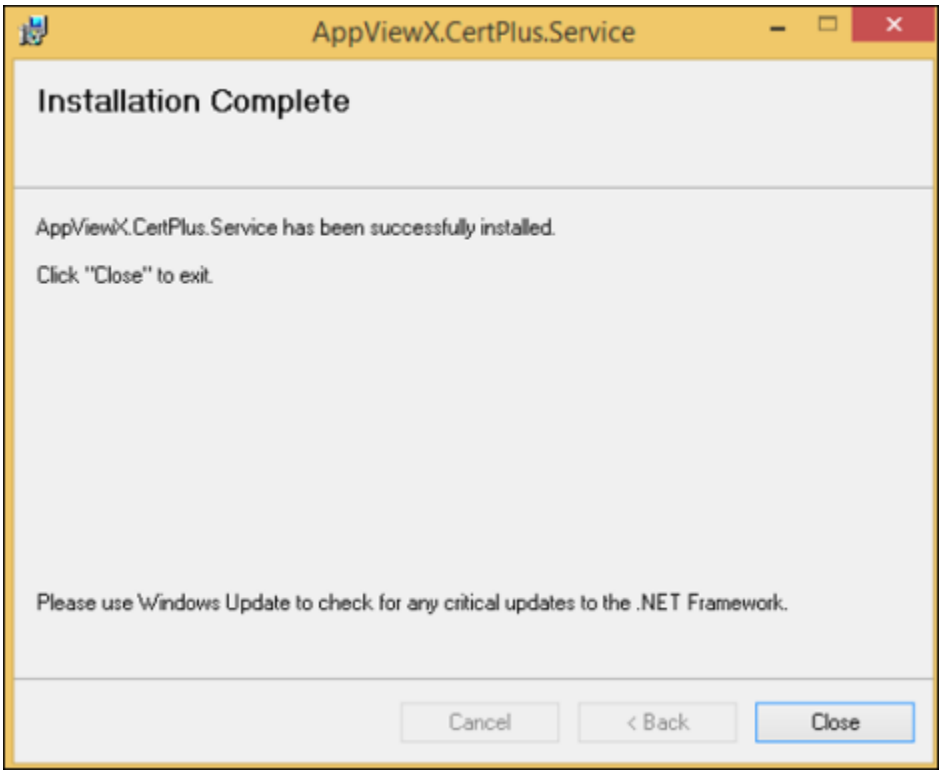



4. Select the certificate store from and to which the certificates must be discovered and pushed by AppViewX.
   This configures the gateway to communicate to the appropriate Certificate store.

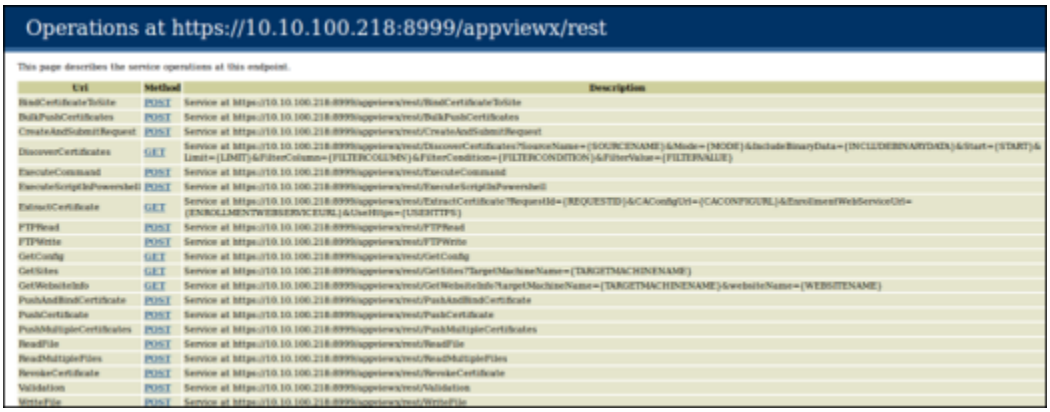
> **Note:** While managing the IIS servers, the certificates are placed in the local machine store.




5. Proceed with the installation to bind the service with the certificates installed.



6. To test the installation, go to the following URL: **https://<IP/Hostname>:<Port>/appviewx/rest/help**
   For example: **https://10.10.10.10:8999/appviewx/rest/help**



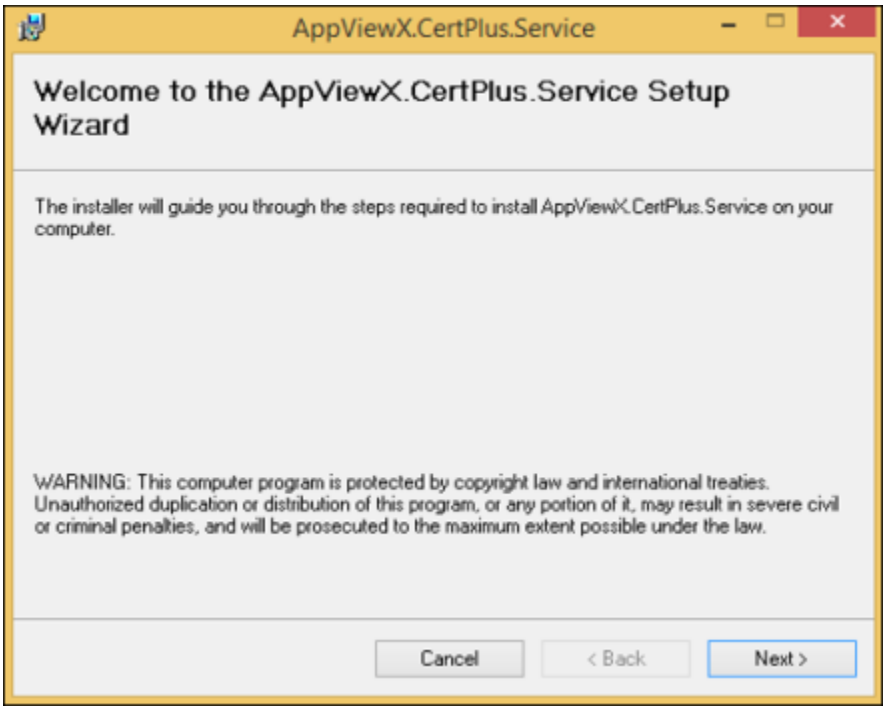

   The above page confirms the accessibility and installation of the service.

> **Note:** In case, a different client authentication certificate is being used, ensure that the CRL mentioned in the certificate is reachable from the AppViewX Windows gateway hosting server.

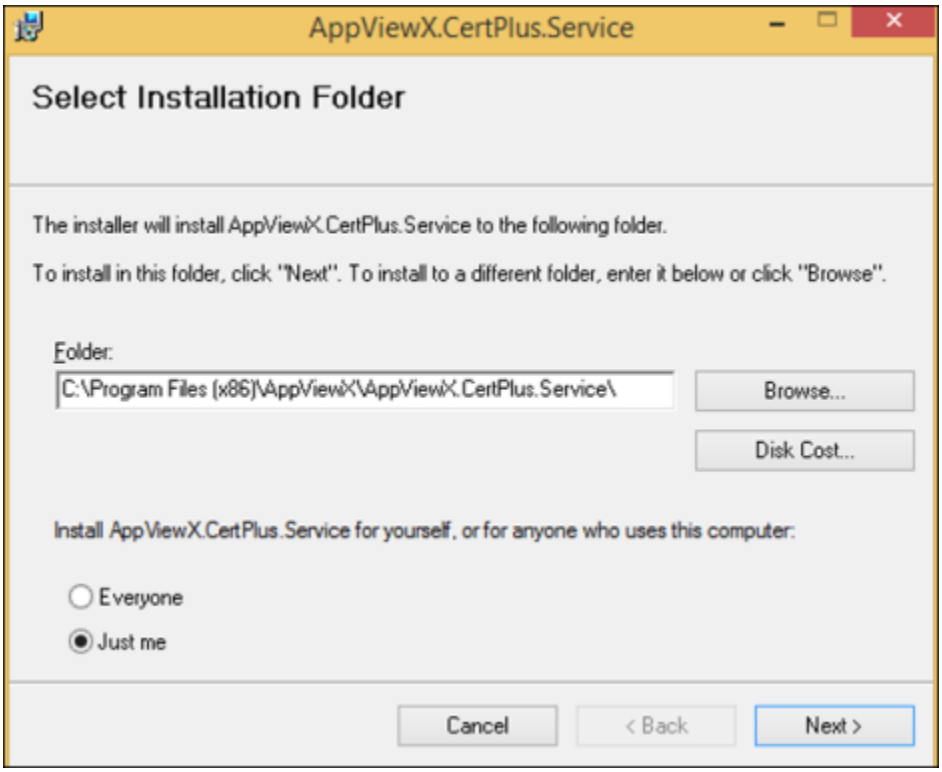
7. Go to **AppViewX** >> **Settings** >> **Certificate** to register the gateway with AppViewX.
8. Register the gateway with the URL format as follows: **https://<IP/Hostname>:<Port>/appviewx**
   For example: **https://10.10.10.10:8999/appviewx**

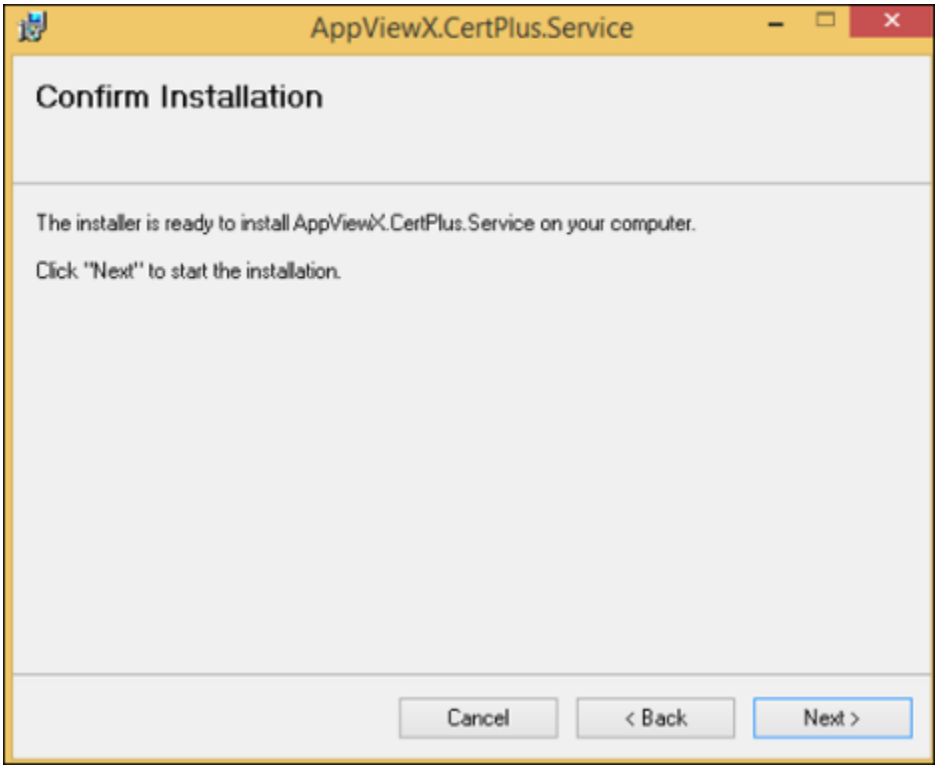
## AppViewX.CertPlus.CustomSetup.msi

1. Click the **AppViewX.CertPlus.CustomSetup.msi** file to install the AppViewX Windows Gateway.

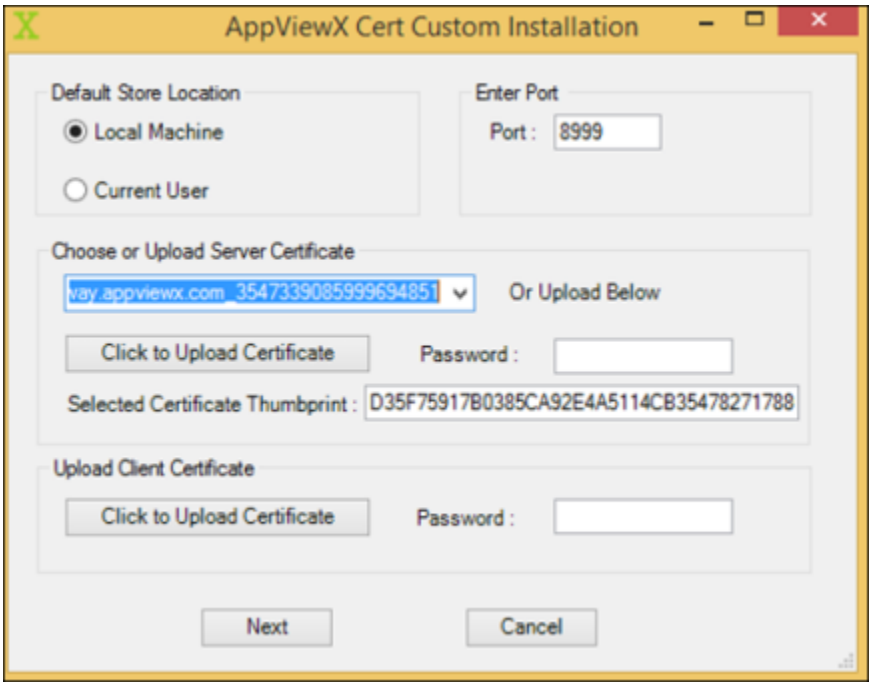



2. Select the users for the service by selecting **Everyone** if the service must be used by other user accounts such as an account apart from the login account used for installing the service.




3. On the Custom Installation screen that opens, do the following:




To allow the port on any antivirus, system and network firewalls for the service to be reachable:

a. Select the default certificate store from and to which the certificates must be discovered and pushed by AppViewX.
   This configures the gateway to communicate to the appropriate Certificate store. While managing the IIS servers, the certificates are placed in the local machine store.

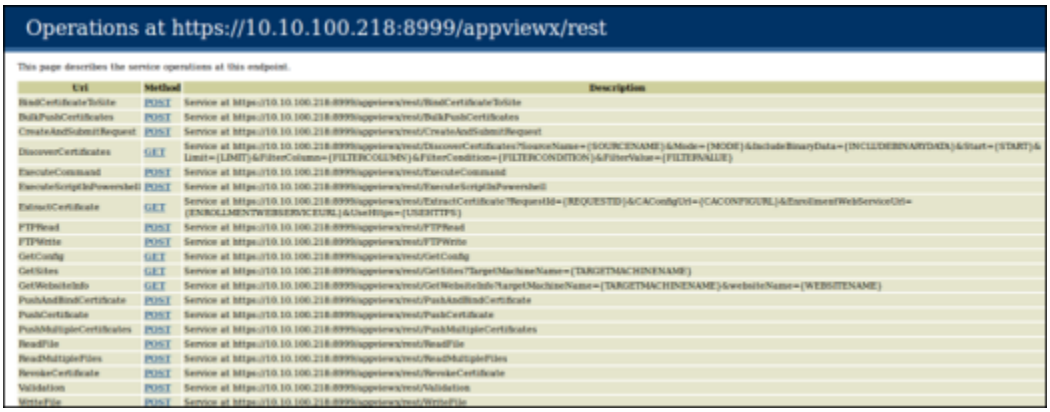
b. Enter a custom port for accessing the service.
   The default value is 8999, which can be modified if required.

c. Select one of the existing server certificates from the dropdown list.
   By default, the **Thumbprint** value for the selected certificate will be auto-populated.

d. Click the **Click to Upload Certificate** button to upload both the Server and Client certificate and enter the password in the corresponding fields.

e. Click **Next** to complete the installation and bind the service with the certificates installed.

4. To test the installation, go to the following URL: **https://<IP/Hostname>:<Port>/appviewx/rest/help**
   For example: **https://10.10.10.10:8999/appviewx/rest/help**




The above page confirms the accessibility and installation of the service. In case, a different client authentication certificate is being used, ensure that the CRL mentioned in the certificate is reachable from the AppViewX Windows gateway hosting server.

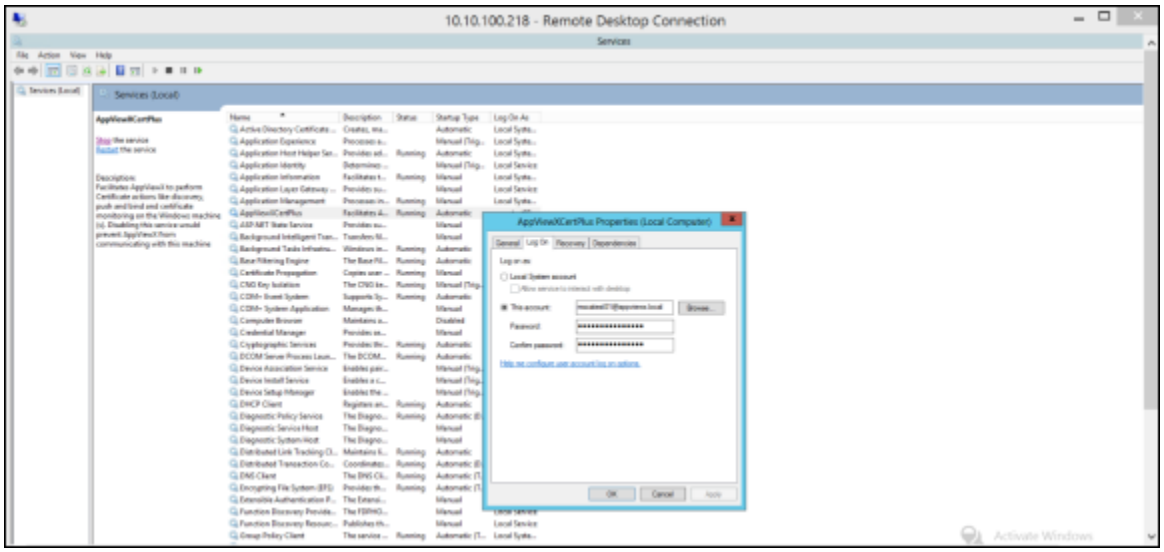
5. Navigate to **AppViewX** > **Settings** > **Certificate** to register the gateway with AppViewX.
6. Register the gateway with the URL format as follows: **https://<IP/Hostname>:<Port>/appviewx**
   For example: **https://10.10.10.10:8999/appviewx**

## Agent Setup When the Service Account is not a Part of the Administrator Group

The Windows gateway agent can be installed using a service account or an admin account.

If the customer has a policy that states that the service account cannot be part of the administrator group or that the service account is only a part of the user group, then:

- The gateway agent is installed using an admin account.
- The installed agent is associated to the service account in services.msc, by adding the account in the properties of the AppViewXCert Plus service. Refer the following image:




1. In this case, the following command has to be executed from the PowerShell: `netsh http add urlacl url=https://+:8999/appviewx/ user=appviewx.localmscatest01`

   In the above command, the value for **user = <domainserviceaccount>** and the URL must be changed respectively.
2. Once this is done, stop and start the **AppViewXCertPlus Service** in **services.msc**.

## Configuration Settings File

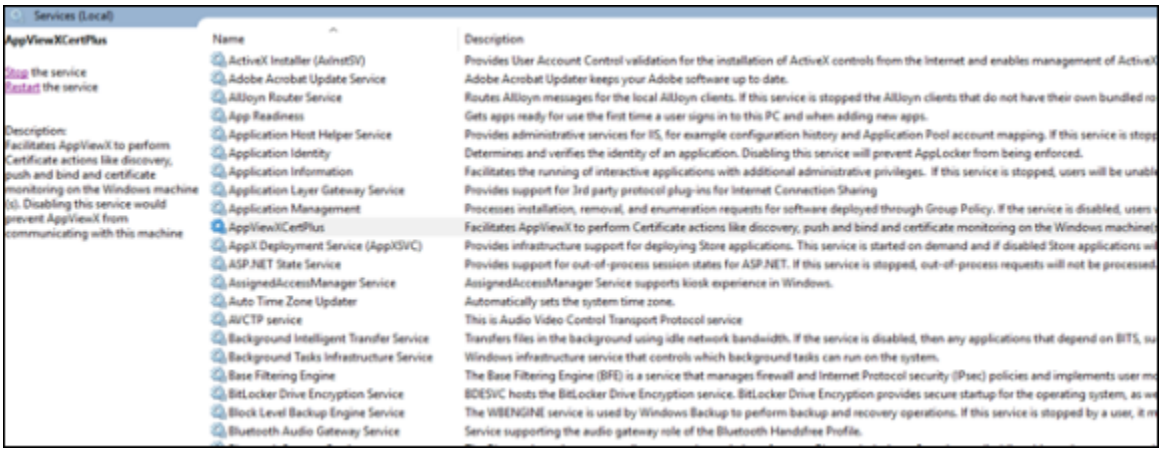
While installing a windows agent in the installation folder, the configuration file can be found in the name of 'config.xml'. The configuration details can be modified before installing the agent. The following are the details of the configuration:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <appSettings>
    <add key="Vendor" value="Microsoft PC" />
    <add key="CertficateStatus" value="Managed" />
    <add key="AppViewXGatewayUrl" value="https://192.168.97.99:5301/avxapi/" />
    <add key="PushAgentEnabled" value="Yes" />
    <add key="PC" value="Yes" />
    <add key="PushCertifcateSchedule" value="0 0 13 1/1 * ? *" />
  </appSettings>
  </configuration>
```

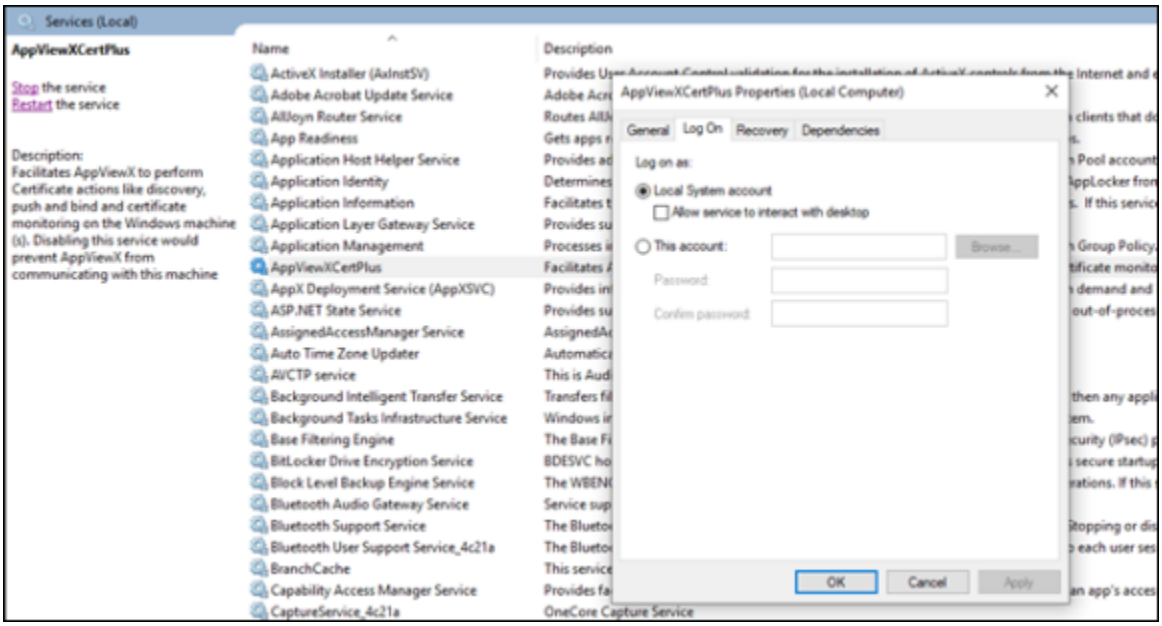
- **Vendor**: The vendor is the device category that has to be added to AppViewX. By default the value is 'Microsoft PC' for personal computers and 'IIS' in case the device is for Internet Information Service.
- **Certificate Status**: Monitored certificates can be just be monitored, while 'Managed' the certificate can be managed and monitored.
- **AppViewXGatewayURL**: The AppViewXGateway URL for pushing the certificate.
- **PushAgentEnabled**: When **Yes** certificates will be pushed to AppViewXGateway API. If **No** the push agent will be disabled.
- **PC**: If it is a personal computer the value is **Yes**. For the server, the value is **No**. On a server when the user makes an RDP connection, the user details will not be captured.
- **PushCertificateSchedule**: The time interval when the push agent gets triggered and the interval time is the cron expression.

## LogOn Application

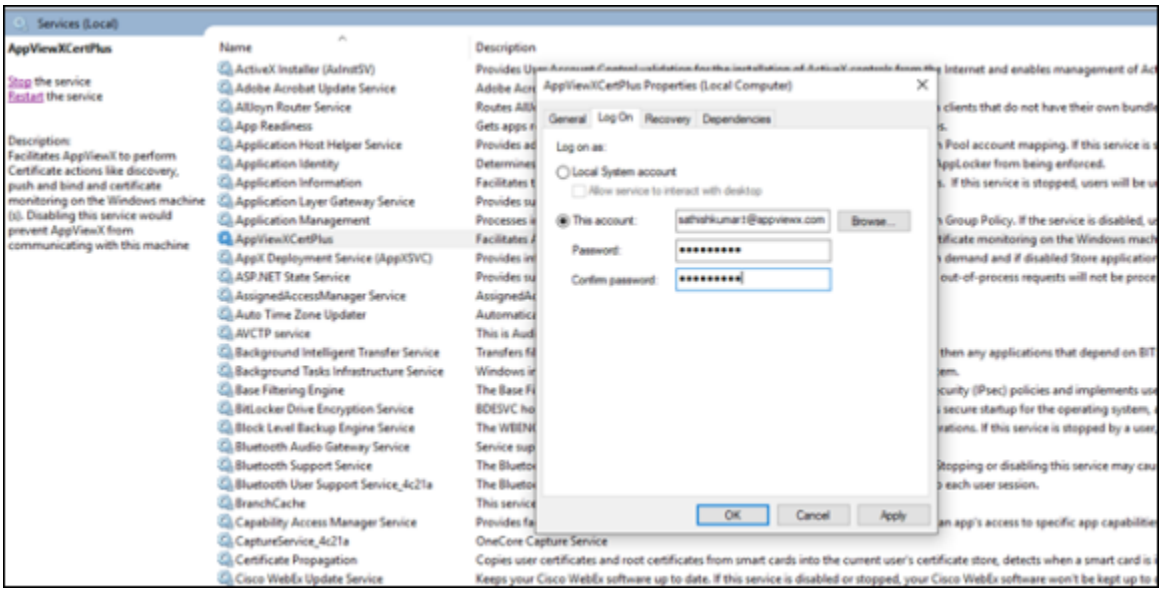
**Service Behavior:** When the Windows Agent is installed on the computer it runs as a Windows service. The agent service can be viewed in **services.msc**.




The Log On details can be viewed by navigating to **Properties** >> **Log On** (tab) as shown below:

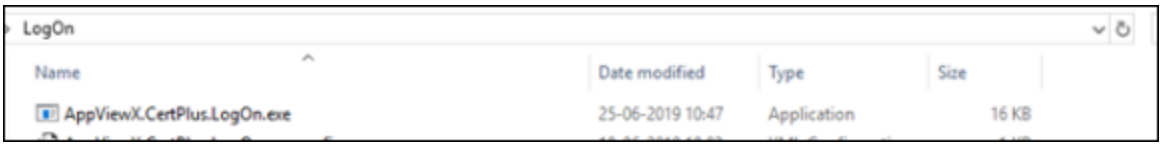



By default, the service runs under the **Local System Account**. If you want to fetch the current user certificates, then the account has to be changed to the current user as shown in the following image:

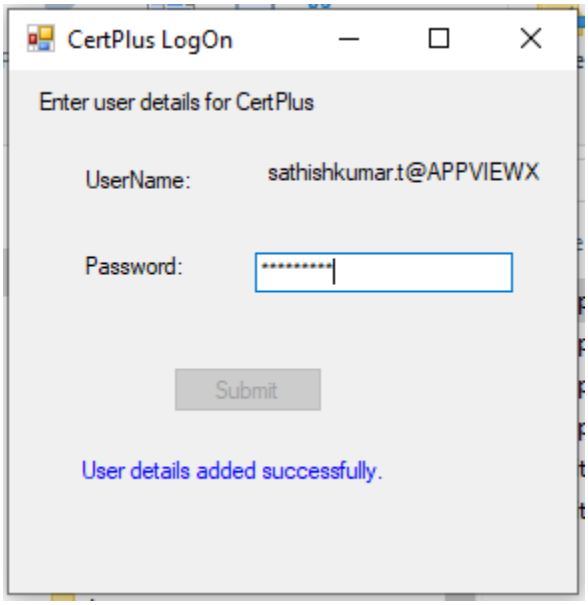



**Note:** A drawback of this approach is that the current user will not have access to **services.msc** and when the user password is changed, then **This account** details have to be updated or the service will fail to start.

So, for the convenience of the user, the **LogOn Application** is created and this LogOn application has to be called through GPO during every user login.




So for the first login the user will be prompted to enter the password as shown in the following image.

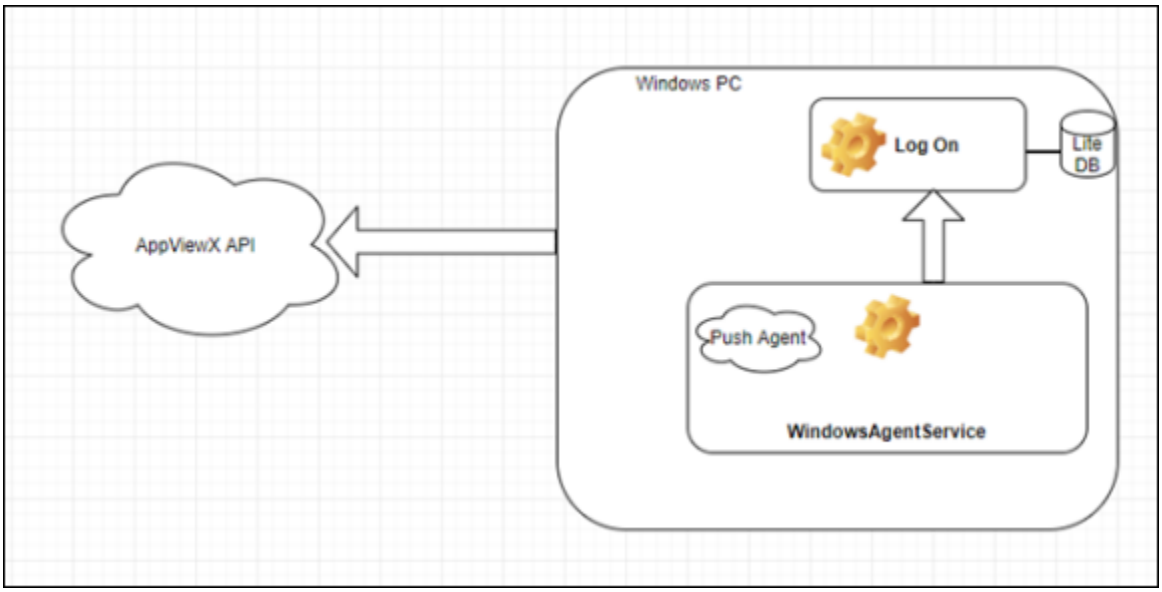



After saving it successfully, the user will not be prompted for a password until there is a change in the password. This is so that the user does not have to go to **services.msc** and configure the credential every time to fetch and push the current user certificate.

## Push Agent

The push agent is a feature specifically designed for the PC user, which sends the certificate automatically to AppViewX from the Windows agent when the user logs into the computer and also during the scheduled time.

**Note:** The schedule time can be configured in the config file before installing the AppViewX Windows Agent. This feature uses the LogOn application to get the user details and scans the current user certificates and sends it back to AppViewX. The following image is the architecture diagram for the push agent.






## Requirements

- PowerShell remoting needs to be enabled for config fetch and discovery.
- Config fetch, certificate discovery will work only when PowerShell remoting is enabled on the local computer.

## Enable PowerShell Remoting

In a PowerShell console running as administrator enable PowerShell Remoting: `Enable-PSRemoting -force`
This should be enough, but if you have to troubleshoot you can use the following commands:

**Troubleshooting:**

a. Make sure the WinRM service is set up to start automatically.
   - **Set start mode to automatic:** `Set-Service WinRM -StartMode Automatic`
   - **Verify start mode and state - it should be running:** `Get-WmiObject -Class win32_service | Where-Object {$_.name -like "WinRM"}`

b. Set all remote hosts to trusted.

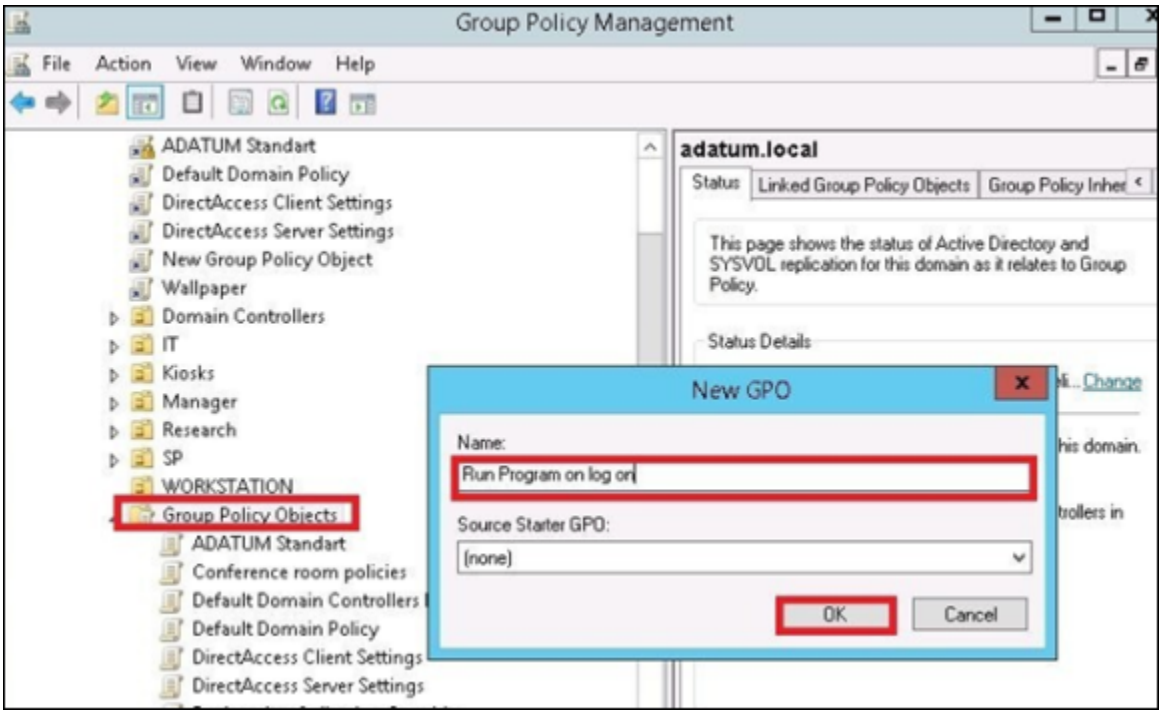
**Note:** You may want to unset this later.

- **Trust all hosts**: `Set-Item WSMan:localhostclienttrustedhosts -value *`
- **Verify trusted hosts configuration:** `Get-Item WSMan:localhostClientTrustedHosts`

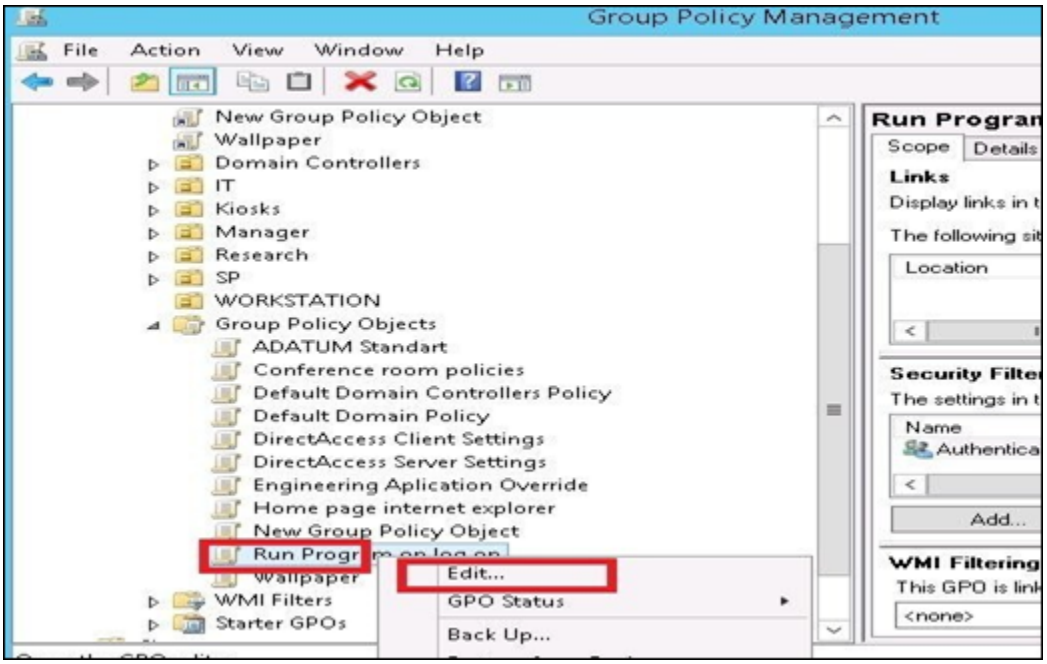
## GPO LogOn Settings

To run the LogOn Application during start-up, it has to be configured in the GPO. Following are the steps to configure it on the GPO:

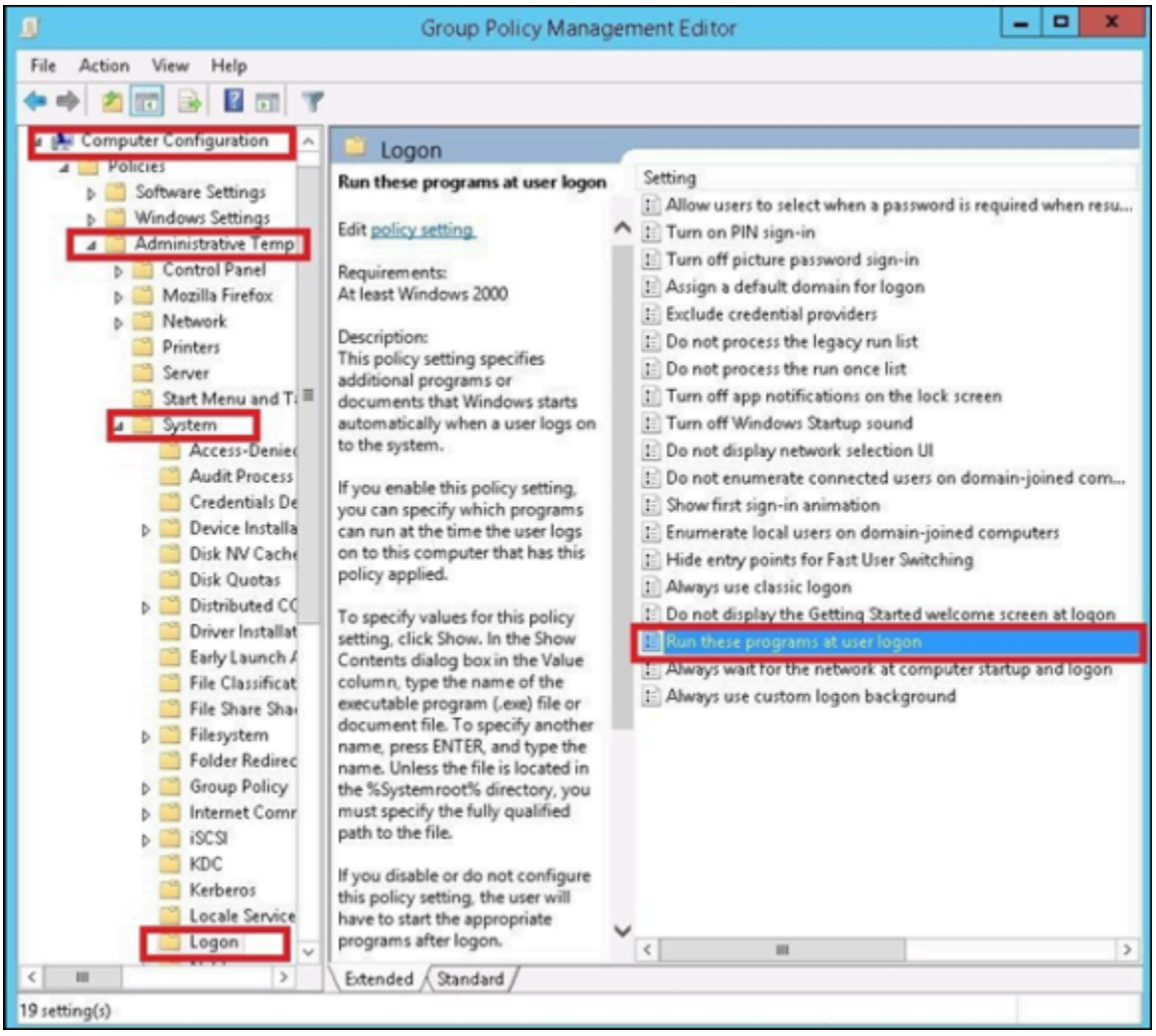
1. Log on to **Windows server and Open Group Policy Management** and create a new policy:



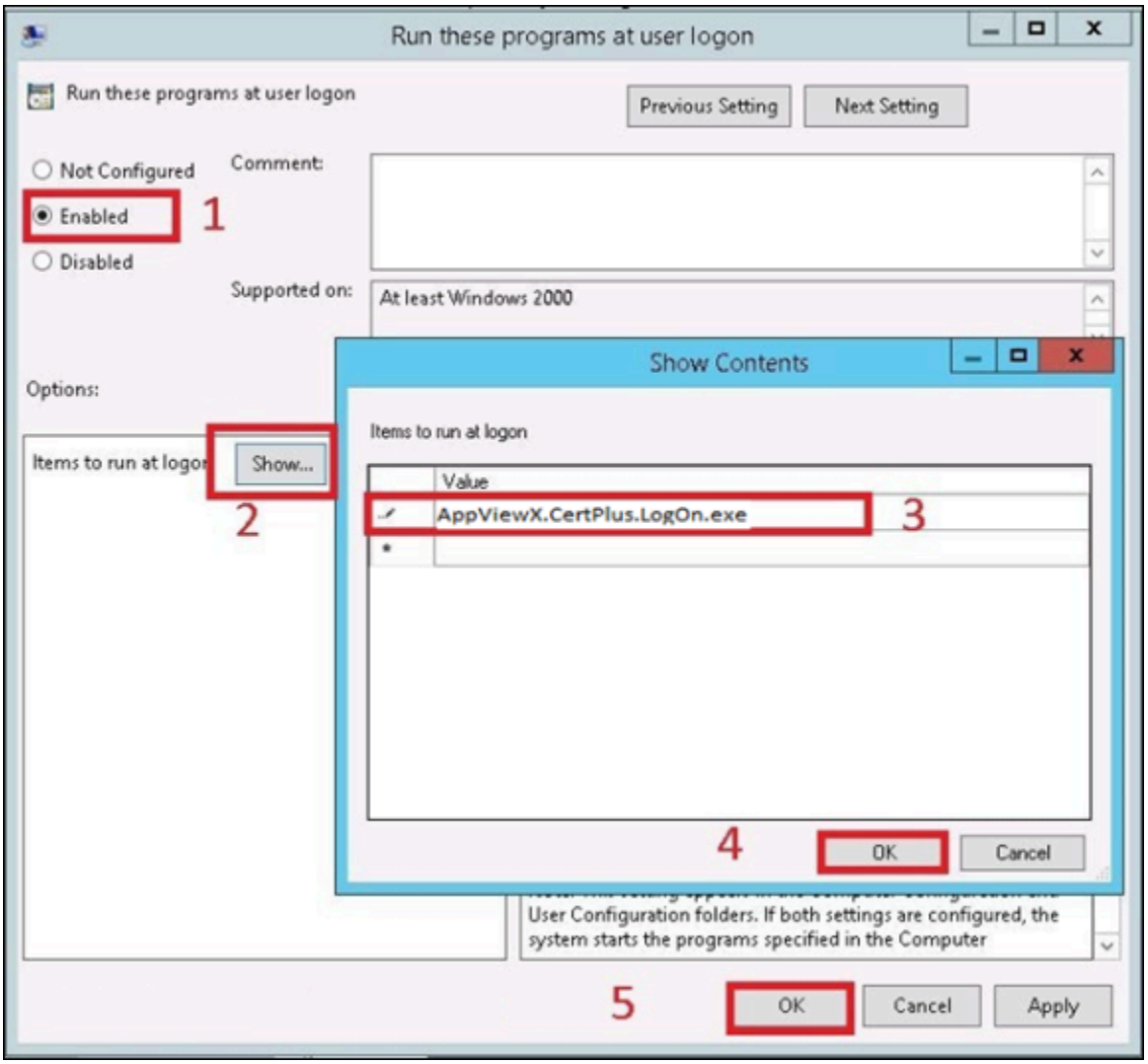

2. Right-click on the created GPO and click **Edit**.



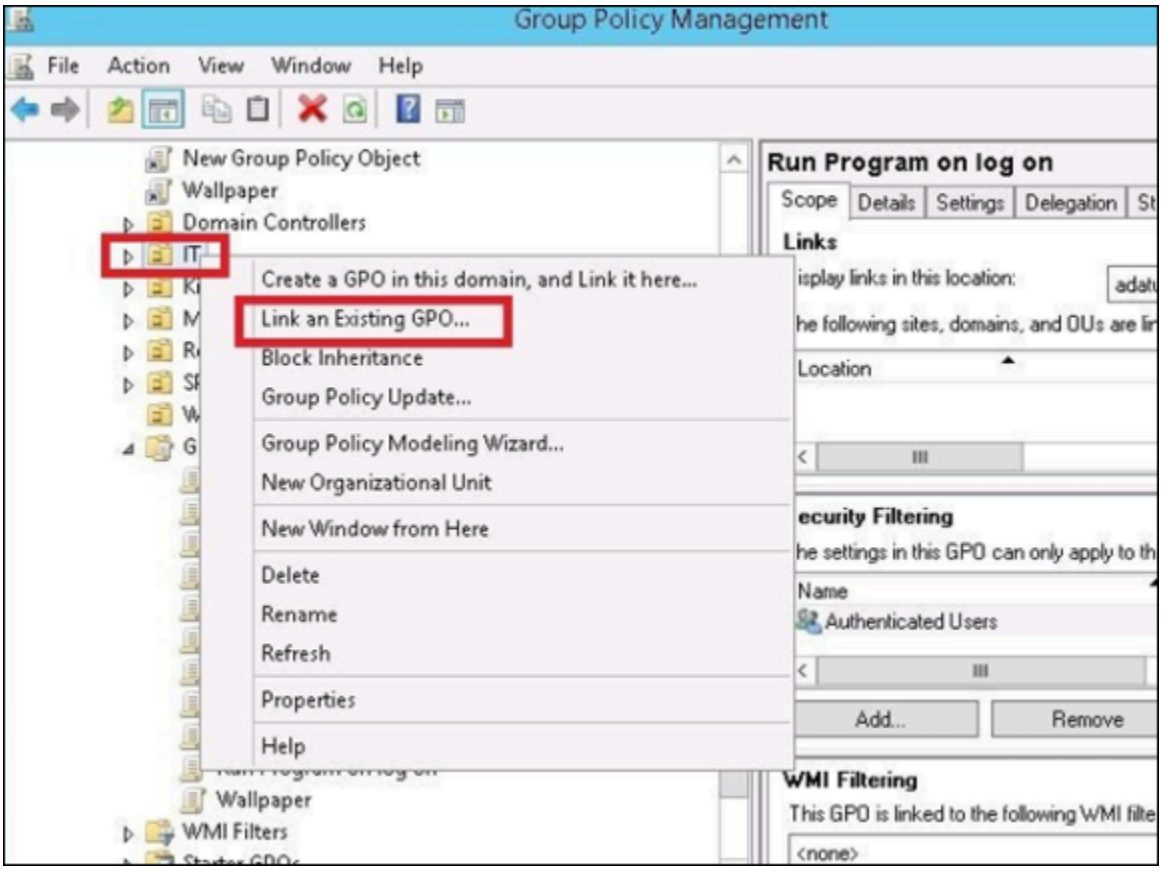

3. Go to **ConfigurationAdministrative TemplatesSystemLogon** and double click **Run** Following are the programs at the user log on:



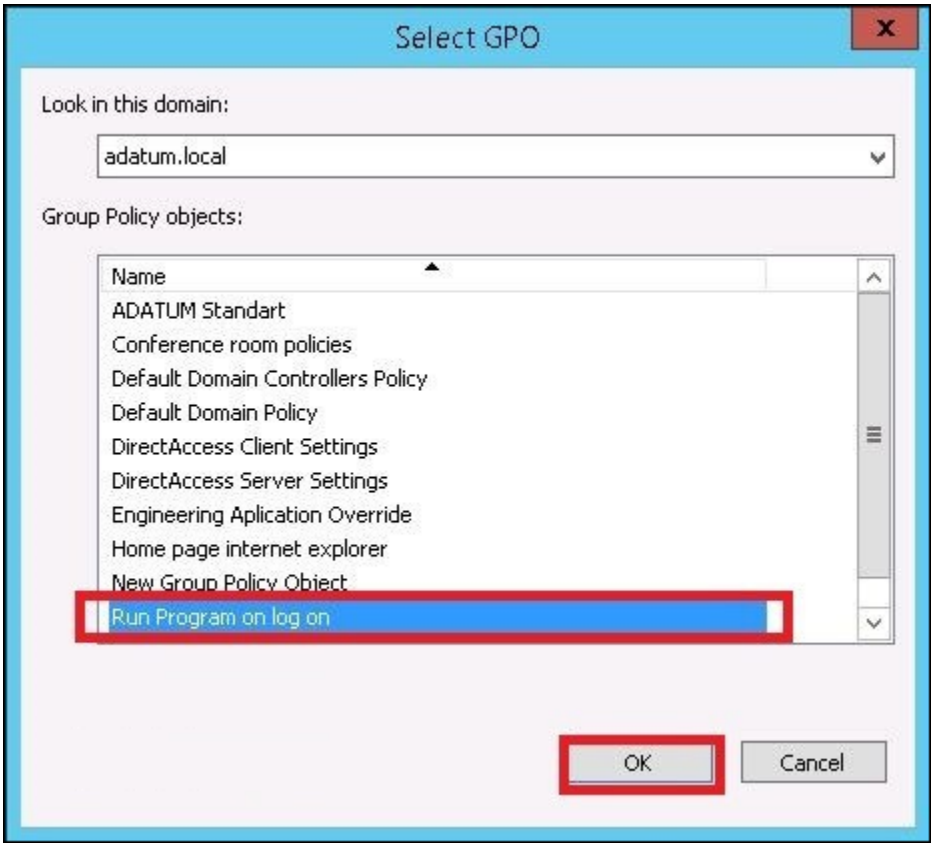

4. Click **Enable** and then click **Show**.
5. Enter the program that you want to run on a user logon (For example Internet Explorer) and click **OK** and then **OK** again.




6. Now you need to apply that GPO on OU that you need. So right-click **OU** and click **Link an Existing GPO**:



7. Choose **GPO Run Program on log on** and click **OK**.




## Upgrade a Web Component

To upgrade a subsystem based web component:

1. Execute the following command to check the status of the web component: `$ appviewx --status avx_platform_web`
2. Download the latest web component that you want to upgrade to the following location: **<user_home_directory>**
3. Execute the following command to initiate the web upgrade process: `$ appviewx --deploy-web <package_location>`
4. After the upgrade process is complete, execute the following command to check the status of the web component: `$ appviewx --status avx_platform_web`.

> **Note:** You need not restart the component for a web upgrade process.

# Apply Release Patch



## Apply Latest Patches Through Release Portal

Complete the following steps to apply all the latest patches released through the release portal:

1. Download the latest plugins from the release portal to a node and extract the archive.
2. Run the following command: `appviewx --apply-patch <path_of_extracted_patch_directory>`

   The following will happen during the execution:
   - The components release_scripts, properties, plugins, gateway, and the web will be updated.
   - The new plugin and release scripts will be executed.
   - If there is no database change for a plugin, the plugin will be restarted one instance at a time.
   - If the web binary is found in the patch, the web will be restarted. If only web plugins are patched, the web component will not be restarted.
   - If there is a change in the property directory, all the components will be restarted.

# Steps to Add Integration Libraries

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



## Prerequisites

- iControl
- CyberArk (javapasswordsdk)
- Thales (jutils,kmjava,nfjava)
- Safenet/Gemalto (jcprov)

- If the customer uses any of the Jars mentioned in the earlier versions of AppViewX and if the customer wants to use it in the 19.3.0 release, the corresponding jars should be downloaded and extracted in **Installer/external lib** folder before the migration/installation process.
- If any customer intends to use any functionality of the jar after the migration or installation, then the corresponding jars should be downloaded and extracted in **/home/appviewx/appviewx/external_libs** folder.

## iControl

1. Go to the following directory: **cd <user_home_dir>/installer/external_libs/**
2. Open your web browser (with an internet connection) and go to the following link: https://devcentral.f5.com/s/articles/iControl-Library-For-Java-With-Source
3. Click on the link **iControl Assembly for Java 13.0.0** from the listed iControl libraries to download the **iControlAssembly-13_0_0-Java.zip** file.
4. Extract the zip file using the following command: `unzip iControlAssembly-13_0_0-Java.zip`
5. Copy **iControl-13.0.0.jar** from the extracted package to the **external_libs** directory.
6. Restart the **avx_vendors** plugin followed by the gateway.

```
appviewx --restart plugins avx_vendors

appviewx --restart gateway

Thales(jutils, kmjava, nfjava)
```

## Thales (jutils, kmjava, nfjava)

Thales client installation should be performed in the node where AppViewX is installed.

1. Go to the directory where the Thales client is installed: **cd /opt/nfast/java/classes**
2. Copy **jutils**, **kmjava** and **nfjava** jars from the directory and paste it to the **external libs** folder in AppViewX.
3. Restart the **avx_vendors** plugin followed by the gateway:

```
appviewx --restart plugins avx_vendors

appviewx --restart gateway
```

## CyberArk (javapasswordsdk)

Cyberark client installation should be performed in the node where AppViewX is installed. After installation, follow the necessary steps listed below:

1. Go to the directory where CyberArk is installed: **cd /opt/CARKaim/sdk/**
2. Copy the **javapasswordsdk** jar from the directory and paste it to the external lib folder in AppViewX.
3. Restart platform core plugin followed by the gateway:

```
appviewx --restart plugins avx_platform_core

appviewx --restart gateway
```

## Safenet/Gemalto (jcprov)

Safenet client installation should be performed in the node where AppViewX is installed.

1. Go to the directory where Safenet/Gemalto is installed: **cd /usr/safenet/lunaclient/jcprov/lib**
2. Copy the **jcprov** jar from the directory and paste it to the **external lib** folder in AppViewX.
3. Restart **avx_vendors** plugin followed by the gateway:

```
appviewx --restart plugins avx_vendors

appviewx --restart gateway
```

# Chapter 14: OS Configurations



## Set Up sudoer Permissions

To set up the required sudoer permissions:

1. Log in to the host as a root user.
2. Execute the following command to create a new user account for AppViewX usage: `useradd -m avxssh -s /bin/bash passwd avxssh`

   You will be prompted to enter the password.
3. Type the password and then press **Enter** on your keyboard.
4. Enter visudo to provide a sudoer privilege and allow the root user to run the commands in the following order: `Cmnd_Alias AVX = /bin/ls, /bin/getent passwd, /bin/test, /bin/grep, /bin/rm, /bin/mv, /bin/cat, /bin/xargs, /bin/stat, /bin/su, /bin/cd root ALL=(ALL) ALL avxssh ALL=(root) AVX`

   **Note:** You can use the netstat package as the telnet package is not available in the disk images (OVA/VHD/QCOW2)

## Configure a Hostname

1. Use the following methods to configure the hostname of a machine running Redhat Linux:
   - The `hostname` command: `hostname appviewxeval.payoda.com`.

     **Note:** This creates a temporary or non-persistent configuration of hostname.

   - The `/etc/sysconfig/network` configuration file (preferred method).
2. To make the configuration persistent, you must configure it in the `/etc/sysconfig/network file`.
3. Open the file in an editor and change the following line:

```
NETWORKING=yes


HOSTNAME=appviewxeval.payoda.com
```

4. After modifying the configuration file, restart the network service to read the `/etc/init.d/network restart` file.

## Configure an IP Address

1. Modify the **/etc/sysconfig/network-scripts/ifcfgeth0** file and provide a static IP address as a root user in Redhat.

   It should be displayed as follows:

   - **DEVICE=eth0**
   - **BOOTPROTO=STATIC**
   - **IPADDR=192.168.0.5**
   - **NETMASK=255.255.255.0**
   - **GATEWAY=192.168.0.1**
   - **ONBOOT=yes**
2. After saving the file, execute the following commands to restart the network daemon:

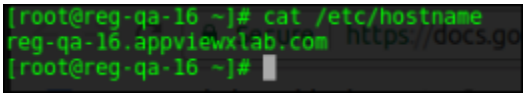
```
$ /etc/init.d/network stop

$ /etc/init.d/network start
```

   This will provide the IP address to the **eth0** interface and also, the `ifconfig` command will list eth0.

## Configure a DNS

1. Add the DNS entries in the **/etc/resolv.con** file as follows: **nameserver DNS-server-IP**
2. To set the hostname for CentOS 7.X and RedHat 7.X by any one of the following methods:

- Add an FQDN in the /etc/hostname file.




- Execute the following command and reboot a server: `# hostnamectl set-hostname your-new-hostname`

## Port Forwarding

1. For CentOS 7.X, run the following command and reboot the server:

```
systemctl start firewalld

firewall-cmd --zone=public --add-masquerade --permanent

firewall-cmd --zone=public --add-rich-rule='rule

family="ipv4" source address="0.0.0.0/0" accept' --permanent

firewall-cmd --zone=public

--add-forward-port=port=443:proto=tcp:toport=5004

--permanent

firewall-cmd --reload
```

2. For CentOS 6.X, add the following line in the **/etc/rc.local** file:

```
iptables -A INPUT -i eth0 -p tcp --dport 443 -j ACCEPT

iptables -A INPUT -i eth0 -p tcp --dport 5004 -j ACCEPT

iptables -A PREROUTING -t nat -i eth0 -p tcp --dport 443 -j REDIRECT --to-port 5004
```

3. Execute the following commands and reboot the server:

```


# chmod +x /etc/rc.local

# iptables -t nat -L
```

## Set the Time Zone

1. In the **/etc/localtime** file, the **localtime** is a link to or copy of a file containing information about your time zone.

> **Note:** The zone information files are available in the /usr/share/zoneinfo, based on your geographical distribution.

2. If the **localtime** file is directed to an incorrect **zoneinfo** file, you can modify it by browsing the directories in /usr/share/zoneinfo to find your country.

> **Note:** You must then find your city or a city belonging to the same time zone and link a **localtime** to it.

## Modify the Date and Time

1. Use the `date` command to view and modify the date and time.

```
root@PiTT ~ # date
Mon Nov  3 02:25:31 PST 2003
```

2. To modify the time, enter date followed by the month, day, hour, minute, and year.

```
root@PiTT ~ # date 110212572003
Sun Nov  2 12:57:00 PST 2003
```

> **Note:** Ensure that all the values you provide are in numerals with no spaces.

> **Note:** The hardware clock can be updated either in UTC (Coordinated Universal Time) or using your local time. However, the standard practice is to update the hardware clock in UTC.
>
> ```
> root@PiTT etc # hwclock --utc --systohc
> ```

## Install Network Time Protocol (NTP)

1. Install the NTP package on your server using the appropriate package management tool available on the Linux distro.
   For example, the NTP package is installed in the Redhat and CentOS by entering the following command: `yum install ntp`
2. To add the NTP server details, complete the following steps:

a. Add a local clock in the **ntp.conf** file to allow the NTP server to provide time from its local system clock when the NTP server is disconnected from the internet.

b. The NTP will connect to a server to fetch the atomic time and the hardware clock is updated by entering the command ntpdate followed by the local/public time server.
For example:

```
$ ntpdate "server DNS name or IP address"

4 Nov 22:31:28 ntpdate[26157]: step time server 209.81.9.7

offset 22317290.440932 sec $ hwclock -systohc
```

c. To maintain an accurate time, create a cron job by entering the following command: `ntpdate "server name" && hwclock -w`

**Note:** Ensure that -w option must be same as `-systohc`

## Configure a Cron Job

The **/etc/cron.allow** and **/etc/cron.deny** files are used to restrict access to cron. The access control files are read every time when a user tries to add or delete a cron job. Both the access control files should possess one username on each line with no whitespaces. It is not necessary to restart the cron daemon (**crond** when the access control files have been modified. The root user can always use cron, regardless of the usernames listed in the access control files.

If the **cron.allow** file exists, only users listed in it are allowed to use cron, and the **cron.deny** file will be ignored.

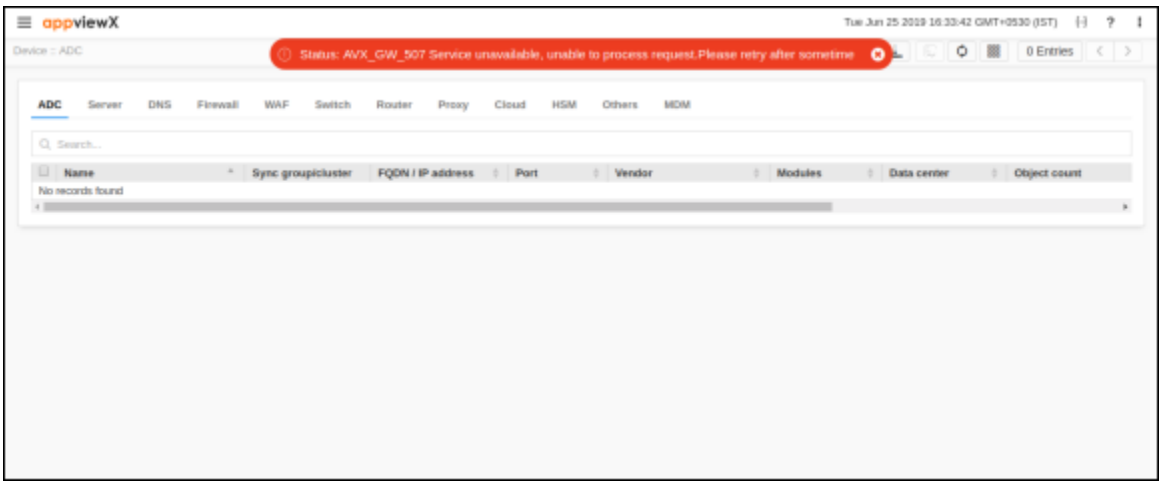
If the **cron.allow** file does not exist, users listed in **cron.deny** file will not be allowed to use cron.

# Chapter 15: Infrastructure Alerts

- Hard Disk Reaching Critical Limits

## Hard Disk Reaching Critical Limits

When the hard disk usage is more than 99% then the gateway will enter sleep mode. In this mode, no new requests will be entertained and the following error banner will be displayed in the Web UI.




When the gateway is in sleep mode, the API's will receive the following response:

**{"response":{"appStatusCode":"GW_507","message":"Gateway is in sleep mode. Disk usage is critically high. Please try after resolving the issue"}}**

# Chapter 16: Appendix A: AppViewX Operational Commands

For operational assistance in the Command Line Interface, use the following commands in the **scripts** directory: for example, **/appviewx/appviewx**.

| Command | Purpose |
|---|---|
| appviewx - - status all | Check the status of all plugins |
| appviewx - - start all | Start all plugins |
| appviewx - - stop all | Stop all plugins |
| appviewx - - start <plugin_ name> | Start Database, Web, or Gateway, Scheduler, Queue |
| appviewx - - stop <plugin_ name> | Stop Database, Web, or Gateway, Scheduler, Queue |
| appviewx - - start plugins <plugin_ name> | Start a particular plugin |
| appviewx - - stop plugins <plugin_ name> | Stop a particular plugin |
| appviewx - - start plugins | Start all plugins |
| appviewx - - stop plugins | Stop all plugins |
| appviewx - - start <plugin_name> $(hostname -i) | Start all plugin in the particular host machine |
| appviewx - - stop <plugin_name> $(hostname -i) | Stop all plugin in the particular host machine |
| appviewx - - start plugins <plugin_ name> $(hostname -i) | Start a particular plugin in the particular host machine |
| appviewx - - stop plugins <plugin_ name> $(hostname -i) | Stop a particular plugin in the particular host machine |
| appviewx - - start plugins $(hostname -i) | Start all plugins in the particular host machine |
| appviewx - - stop plugins $(hostname -i) | Stop all plugins in the particular host machine |
| appviewx - - conf-sync | Sync the configuration file from one host to another host |

| Command | Purpose |
|---|---|
| appviewx - - initialize all | Initialize all plugins after a specific configuration update |
| appviewx - - prerequisite | Check the prerequisite of all hosts |
| appviewx - - prerequisite $(hostname -i) | Check the prerequisite of a particular host |
| appviewx - - restart <plugin_name> | Restart a particular plugin |
| appviewx - - restart plugins <plugin_name> | Restart a particular plugin |
| appviewx - - restart all | Restart all plugins |
| appviewx - - enable-https <all> or <gateway> or <web> or <plugin> | Enable SSL on the plugins |
| appviewx - - disable-https <all> or <gateway> or <web> or <plugins> | Disable SSL on the plugins |
| appviewx - - databasebackup <Backup path> | Trigger a backup |
| appviewx - - databaserestore <Path of the backup archive> | Restore an already available backup archive |
| appviewx - - databaseimport fresh | Insert dbscripts for fresh installation |
| appviewx - - databaseimport upgrade <version> | Insert dbscripts for an upgrade from an older version |
| appviewx --change-db-password | Functionality to change the DB password |
| appviewx --update-node-password | Functionality to update the encrypted node password in the conf file |
| appviewx --db-execution | Functionality to execute a .js file |
| appviewx --sync-files | Functionality to sync directories/files across nodes |
| appviewx --plugin-heapinfo <IP> | Fetch the maximum and the minimum heap sizes of all the available plugins from a specific node |
| appviewx --plugin-heapinfo | Fetch the maximum and the minimum heap sizes of all the available plugins across all the cluster node setup |

| Command | Purpose |
|---|---|
| appviewx --plugin-heapinfo <plugin_name> | Fetch the maximum and the minimum heap sizes of a particular plugin across all the cluster node setup |
| appviewx --gw-api-form | Execute the gateway API |
| appviewx --update-plugin-heapsize | Add and update the heap size for each plugin across all the nodes |
| appviewx --update-plugin-loglevel | Update the log level for each plugin across all the nodes |
| --license host-fetch | Fetch the hostname on which the license needs to be generated |
| --license host-update | Update the hostname for the new license |
| --db-shell <username> | Provision to access and query the database |
| appviewx --start avx_platform_elastic | Start the elastic search component across all the cluster nodes |
| appviewx --start avx_platform_elastic <IP> | Start the elastic search component for a single node |
| appviewx --stop avx_platform_elastic | Stop the elastic search component across all the cluster nodes |
| appviewx --stop avx_platform_elastic <IP> | Stop the elastic search component for a single node |
| appviewx --restart avx_platform_elastic | Restart the elastic search component across all the cluster nodes |
| appviewx --restart avx_platform_elastic <IP> | Restart the elastic search component for a single node |
| appviewx --status avx_platform_elastic | Check the status of the elastic search component across all the cluster nodes |
| appviewx --status avx_platform_elastic <IP> | Check the status of the elastic search component for a single node |
| appviewx --elastic -passwd-update | Change the password of elastic search component for the admin user |
| appviewx --version | View the version of all the components used in the application |
| appviewx --view_logs <Transaction ID> | View the logs associated with a specific transaction ID |

| Command | Purpose |
|---|---|
| appviewx --gwrefresh mask | View the status of gateway components |
| appviewx --buildinfo <component> | View the build information of AppViewX components |
| appviewx --start avx_platform_vault | Start the vault component across all the nodes |
| appviewx --start avx_platform_vault <IP> | Start the vault component for a specific node |
| appviewx --start avx_platform_consul server | Start the avx_platform_consul server across all the nodes |
| appviewx --start avx_platform_consul client | Start the avx_platform_consul client across all the nodes |

# Chapter 17: Appendix B: AppViewX Stack Plugins List and Default Ports

| Plugin Name | Default Port | Type | Direction | Port Configurable |
|---|---|---|---|---|
| avx_platform_database | 5000 | TCP | Bi-directional | Yes |
| avx_platform_core | 5001 | TCP | Bi-directional | Yes |
| avx_platform_queue | 5002 | TCP | Bi-directional | Yes |
| avx_platform_web | 5004 | TCP | Bi-directional | Yes |
| avx_platform_syslog | 5005 | TCP | Bi-directional | Yes |
| avx_platform_syslog_receiver | 5006 | TCP | Bi-directional | Yes |
| avx_platform_report_generator | 5008 | TCP | Bi-directional | Yes |
| avx_platform_amc | 5009 | TCP | Bi-directional | Yes |
| avx_platform_gateway | 5300 | TCP | Bi-directional | Yes |
| avx_platform_scheduler | 5600 | TCP | Bi-directional | Yes |
| avx_platform_logs | 5514 | UDP | Bi-directional | Yes |
| avx_platform_elastic | 5500/5550 | TCP | Bi-directional | Yes |
| avx_platform_consul | 5902/5912 | TCP | Bi-directional | Yes |
| avx_platform_vault | 5920/5921 | TCP | Bi-directional | Yes |

| Plugin Name | Default Port | Type | Direction | Port Configurable |
|---|---|---|---|---|
| avx_subsystems | 5100 | TCP | Bi-directional | Yes |
| avx_subsystem_adc | 5101 | TCP | Bi-directional | Yes |
| avx_subsystem_automation | 5102 | TCP | Bi-directional | Yes |
| avx_subsystem_certificate | 5103 | TCP | Bi-directional | Yes |
| avx_subsystem_dns | 5104 | TCP | Bi-directional | Yes |
| avx_subsystem_misc_devices | 5105 | TCP | Bi-directional | Yes |
| avx_subsystem_others | 5106 | TCP | Bi-directional | Yes |
| avx_subsystem_router | 5107 | TCP | Bi-directional | Yes |

| Plugin Name | Default Port | Type | Direction | Port Configurable |
|---|---|---|---|---|
| avx_subsystem_security | 5108 | TCP | Bi-directional | Yes |
| avx_subsystem_ssh | 5109 | TCP | Bi-directional | Yes |
| avx_subsystem_switch | 5110 | TCP | Bi-directional | Yes |
| avx_subsystem_waf | 5111 | TCP | Bi-directional | Yes |
| avx_subsystem_proxy | 5112 | TCP | Bi-directional | Yes |
| avx_subsystem_cloud | 5113 | TCP | Bi-directional | Yes |
| avx_insight_subsystem_adc | 5114 | TCP | Bi-directional | Yes |
| avx_subsystems_ui | 5115 | TCP | Bi-directional | Yes |
| avx_subsystem_adc_ui | 5116 | TCP | Bi-directional | Yes |
| avx_vendors | 5200 | TCP | Bi-directional | Yes |
| avx_vendor_a10 | 5201 | TCP | Bi-directional | Yes |
| avx_vendor_amazonelb | 5202 | TCP | Bi-directional | Yes |
| avx_vendor_automation | 5203 | TCP | Bi-directional | Yes |
| avx_vendor_avi | 5204 | TCP | Bi-directional | Yes |
| avx_vendor_bigiq | 5205 | TCP | Bi-directional | Yes |
| avx_vendor_cert_app | 5206 | TCP | Bi-directional | Yes |
| avx_vendor_cert_network_discovery | 5207 | TCP | Bi-directional | Yes |
| avx_vendor_cert_ca | 5208 | TCP | Bi-directional | Yes |
| avx_vendor_cert_hsm_safenet | 5209 | TCP | Bi-directional | Yes |
| avx_vendor_cert_server | 5210 | TCP | Bi-directional | Yes |
| avx_vendor_citrix | 5211 | TCP | Bi-directional | Yes |
| avx_vendor_dns-bind | 5212 | TCP | Bi-directional | Yes |
| avx_vendor_dns-qip | 5214 | TCP | Bi-directional | Yes |
| avx_vendor_dns-infoblox | 5213 | TCP | Bi-directional | Yes |
| avx_vendor_f5 | 5215 | TCP | Bi-directional | Yes |

| Plugin Name | Default Port | Type | Direction | Port Configurable |
|---|---|---|---|---|
| avx_vendor_fortigate | 5216 | TCP | Bi-directional | Yes |
| avx_vendor_fortimanager | 5217 | TCP | Bi-directional | Yes |
| avx_vendor_fw-checkpoint | 5218 | TCP | Bi-directional | Yes |
| avx_vendor_fw-cisco-asa | 5221 | TCP | Bi-directional | Yes |
| avx_vendor_fw-f5-afm | 5222 | TCP | Bi-directional | Yes |
| avx_vendor_fw-juniper | 5223 | TCP | Bi-directional | Yes |
| avx_vendor_fwstat_southbound | 5224 | TCP | Bi-directional | Yes |
| avx_vendor_haproxy | 5225 | TCP | Bi-directional | Yes |
| avx_vendor_misc_devices | 5226 | TCP | Bi-directional | Yes |
| avx_vendor_nginxplus | 5227 | TCP | Bi-directional | Yes |
| avx_vendor_others | 5228 | TCP | Bi-directional | Yes |
| avx_vendor_paloalto | 5229 | TCP | Bi-directional | Yes |
| avx_vendor_panorama | 5230 | TCP | Bi-directional | Yes |
| avx_vendor_router_cisco | 5231 | TCP | Bi-directional | Yes |
| avx_vendor_router_juniper | 5232 | TCP | Bi-directional | Yes |
| avx_vendor_ssh_aws | 5233 | TCP | Bi-directional | Yes |
| avx_vendor_ssh_f5 | 5234 | TCP | Bi-directional | Yes |
| avx_vendor_ssh_linux | 5235 | TCP | Bi-directional | Yes |
| avx_vendor_switch_arista | 5236 | TCP | Bi-directional | Yes |
| avx_vendor_switch_cisco | 5237 | TCP | Bi-directional | Yes |
| avx_vendor_switch_juniper | 5238 | TCP | Bi-directional | Yes |
| avx_vendor_waf-f5 | 5239 | TCP | Bi-directional | Yes |
| **avx_subsystems_syn** | 5117 | TCP | Bi-directional | Yes |
| **LOGGING_TOOL_PORT** | 4712 | TCP | Bi-directional | Yes |
| avx_vendor_cloud | 5240 | TCP | Bi-directional | Yes |
| **avx_platform_logforwarding** | 5010 | TCP | Bi-directional | Yes |

| Plugin Name | Default Port | Type | Direction | Port Configurable |
|---|---|---|---|---|
| **avx_visual_page_builder** | 5011 | TCP | Bi-directional | Yes |
| avx_vendor_proxy_squid | 5241 | TCP | Bi-directional | Yes |
| avx_vendor_cisco_ace | 5242 | TCP | Bi-directional | Yes |
| **avx_vendor_cert_est_agent** | 5013 | TCP | Bi-directional | Yes |
| **avx_vendor_cert_intune_agent** | 5014 | TCP | Bi-directional | Yes |
| **avx_vendor_cert_acme_agent** | 5015 | TCP | Bi-directional | Yes |
| avx_vendor_cert_scep_agent | 5250 | TCP | Bi-directional | Yes |
| avx_vendor_cert_mdm | 5252 | TCP | Bi-directional | Yes |
| avx_vendor_akamai | 5253 | TCP | Bi-directional | Yes |
| avx_vendor_ssh_windows | 5254 | TCP | Bi-directional | Yes |
| avx_insight_vendor | 5243 | TCP | Bi-directional | Yes |
| avx_insight_vendor_f5 | 5244 | TCP | Bi-directional | Yes |
| avx_insight_vendor_citrix | 5245 | TCP | Bi-directional | Yes |
| avx_insight_vendor_a10 | 5246 | TCP | Bi-directional | Yes |
| avx_insight_statistics_bot | 5247 | TCP | Bi-directional | Yes |

# Chapter 18: Appendix C: Firewall Rules

| Source Component | Source Port | Source IP | Destination Service | Destination Port | Protocol | Type |
|---|---|---|---|---|---|---|
| Big IP LTM/GTM IP | Any | Big IP LTM/ GTM IP IPs | avx_platform_logs(logstash) | 5514(Default port) | UDP | Inbound |
| Web Load balancer(VIP) | Any | LoadBalancer of Web VIP | AppViewX Web nodes | 5004 | TCP | Inbound |
| Logstash(avx_platform_logs) | 5512 | Logstash IP | KAFKA node | Any | UDP | Outbound |
| SCEP supported network devices | Any | Network Device IP | AppViewX SCEP Plugin | 5250 | TCP | Inbound |

Open the below port to access the web application:

| Component | Port | Protocol | Type |
|---|---|---|---|
| Web | 5004 | TCP | Inbound |

The below rules are for internal communication between AppViewX components:

| Component | Port | Protocol | Type |
|---|---|---|---|
| MongoDB | 5000 | TCP | Inbound |
| avx_platform_consul | 5902 and 5912 | TCP | Inbound |
| avx_platform_vault | 5920 and 5921 | TCP | Inbound |
| plugins | 5001, 5002, 5008, 5100, 5200, 5207, and 5250 | TCP | Inbound |
| avx_platform_gateway | 5300 | TCP | Inbound |

The below rules are for syslog communication between AppViewX components:

| Component | Port | Protocol | Type |
|---|---|---|---|
| Logstash | 5512 and 5514 | TCP | Inbound |

| Component | Port | Protocol | Type |
|---|---|---|---|
| avx_platform_syslog | 5005 | TCP | Inbound |
| avx_platform_syslog_receiver | 5006 | TCP | Inbound |

The below rules are for insight operation between AppViewX components:

| Component | Port | Protocol | Type |
|---|---|---|---|
| Elasticsearch | 5500 and 5550 | TCP | Inbound |
| avx_insight_subsystem_adc | 5114 | TCP | Inbound |
| avx_insight_statistics_bot | 5247 | TCP | Inbound |
| **Note:** Make sure the outbound ports are not open. | | | |

# Chapter 19: Appendix D: General Setup Default Ports

| Plugin Name | Default Port | Port | Destination Type | Customizable |
|---|---|---|---|---|
| HTTPS | 443 | TCP | Bi-directional | No |
| SSH | 22 | TCP | Bi-directional | Yes |
| LDAP | 386 | TCP | Inbound | No |
| LDAPS | 636 | TCP | Inbound | No |
| Radius Authentication | 1812, 1646 | UDP | Inbound | No |
| Radius Account | 1813, 1646 | UDP | Inbound | No |
| TACACS | 49 | TCP | Inbound | No |
| SYSLOG | 9514 | TCP | Bi-directional | No |
| TRAP | 9164 | TCP | Bi-directional | No |
| SMTP (Email) | 25 | TCP | Bi-directional | Yes |
| SNMP | 4162 | TCP | Inbound | No |

# Chapter 20: Appendix E: Error Codes



## HTTP Codes

- 401 - Unauthorized
- 403 - Forbidden
- 404 - Requested API is not found
- 429 - VM cannot serve any more requests
- 500 - Internal server error
- 501 - Not implemented
- 503 - Service unavailable
- 504 - Gateway timeout
- 507 - Insufficient storage

## License Error Codes

- 700 - Success
- 701 - Date based license is going to expire warning
- 702 - Date based license expired error
- 703 - License is not found
- 704 - Firewall device count exceeded

# Chapter 21: Appendix F: AppViewX Component Descriptions

| Component Name | Subsystem | Description |
|---|---|---|
| avx_vendor_switch_cisco | South Bound | The plugin contains the Cisco switch specific communication logic |
| avx_vendor_switch_juniper | South Bound | The plugin contains the Juniper switch specific communication logic |
| avx_vendor_waf-f5 | South Bound | Communicates with the F5 device to manage ASM policies and holds parsing logic |
| avx_vendor_cloud | South Bound | Communicates with the cloud in general |
| avx_vendor_proxy_squid | South Bound | Communicates with the proxy squid system |
| avx_vendor_cisco_ace | South Bound | The plugin contains the Cisco ACE vendor specific communication and parsing logic for all versions |
| avx_insight_vendor | South Bound | The plugin contains the statistics parsing logic of the F5,Citrix, and A10 vendors |
| avx_insight_vendor_f5 | South Bound | The plugin contains the statistics parsing logic of F5 vendor |
| avx_insight_vendor_citrix | South Bound | The plugin contains the statistics parsing logic of Citrix vendor |
| avx_insight_vendor_a10 | South Bound | The plugin contains the statistics parsing logic of A10 vendor |
| avx_insight_statistics_bot | South Bound | The plugin responsible for complete statistics collection process and it runs as a separate service |
| avx_vendor_fwstat_southbound | South Bound | The plugin used for collecting firewall statistics from the devices |
| avx_vendor_cert_scep_agent | South Bound | **Certificate auto enrollment for scep protocol** |

| Component Name | Subsystem | Description |
|---|---|---|
| avx_vendor_cert_mdm | South Bound | The plugin resposible to interact with the MDM end points. |
| avx_vendor_akamai | South Bound | The plugin responsible to hold the akamai vendor specific communication and parse the logic for all the vendors. |
| avx_vendor_ssh_windows | South Bound | Interacts with the windows machine to retrieve the details and will perform all the SSH communications. |