# AppViewX Application Upgrade Guide

**Version: 2022.1.0 FP5**

# Copyright AppViewX, Inc.



## Trademarks



## 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 |
|---|---|---|
| v1.0 | Application Upgrade Guide for Release 2022.1.0 FP5 | May 2025 |

## About this Guide

The document describes the steps to upgrade AppViewX to the latest version 2022.1.0 FP5

## Audience

This guide is intended for AppViewX's customers deploying its products on Windows-based machines.

## Third-Party Software Acknowledgements

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

For example,

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

## Text Conventions

The following text conventions are used in this document:

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

# Chapter 1: Prerequisites

## General Prerequisites

1. If you are currently using CentOS operating system, refer to the CentOS Migration Guide.
2. Nodes must have the following OS:
   - RHEL 8.7, 8.8, 8.10, 9.5
   - Ubuntu 20.04
3. Keep the following file locations ready -
   - old installer file location, for example - /home/appviewx/FP2/appviewx_kubernetes
   - installed location, for example - /home/appviewx/appviewx_cluster
4. Location to save the new installer file - /home/appviewx/appviewx/Application_upgrade (Assign the folder name as required).
5. To list the nodes in the cluster, execute the command below and save the output for further reference.

   ```
   kubectl get nodes --show-labels
   ```

   **Note:** If custom labels are detected add them to the custom_changes.yaml file. Refer to the chapter *Adding Custom Pod Configuration* of the section **Monitoring and Maintaining AppViewX** in the **Install, Upgrade, and Maintenance Guide**.

6. To get the status of HPA, execute the command below and save the output for further reference

   ```
   kubectl get hpa
   ```

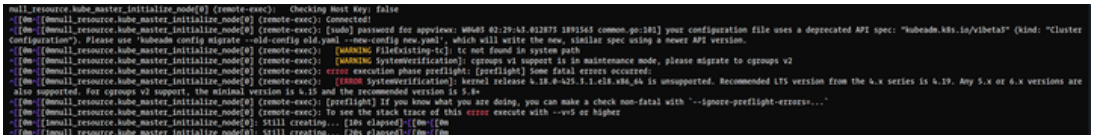
7. Keep a backup of the iControl jar files available at location - /home/appviewx/appviewx/appviewx_dependencies/external_libs/iControl-13.1.0.jar for iControl to be done after the upgrade.
8. Check for any other custom changes that may have been done specifically for the customer.
9. Check the enabled plugins in appviewx.conf file of the old installer (previous version), in case there are plugins that are not present but are required to be installed in the latest versions, please add them, example
   - avx_pkiaas_cert_ocsp_generator
   - avx_pkiaas_cert_ocsp_serveravx_pkiaas_cert_ocsp_server
   - avx_platform_hsm

## Updating the Kernel version

Following the upgrade to **Kubernetes v1.32.2**, the `kubeadm init` setup is no longer compatible with **older Linux kernel versions**. This is primarily due to enhanced reliance on **cgroups** (control groups), a Linux

kernel feature that manages resource isolation and allocation. This incompatibility typically affects nodes running older operating system versions, particularly those in the **RHEL 8 series** (e.g., 8.5, 8.6, or even 8.10), depending on the specific kernel version present on the node.

Figure 1. Error log from kubernetes



**Cgroups Overview:**

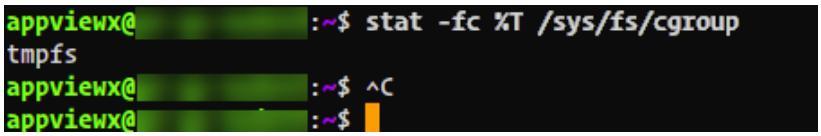
Control Groups (cgroups) are a kernel-level feature that enables the limitation, prioritization, and isolation of resource usage (CPU, memory, I/O, etc.) among process groups. Kubernetes leverages cgroups extensively for container orchestration.

**Determining the Cgroups Version in Use:**

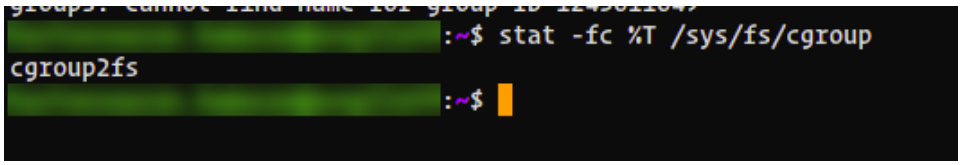
To identify the active cgroups version on a Linux system, run the following command:

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

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




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



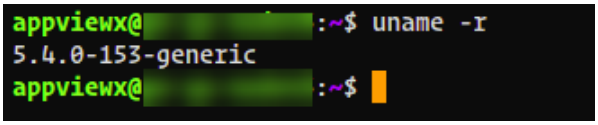

**Kernel Version Requirements Based on Cgroups Version**

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

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

**To verify the kernel version**, execute the command:

```
uname -r
```




**Solution provided:**

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

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

## Points to Remember when Upgrading to 2022.1.5

Know the following before proceeding with the upgrade.

1. If an external CA certificate is configured for kubernetes, the infra upgrade will overwrite the certificates.
2. A manual elastic restore must be performed post the upgrade. Post the application upgrade the back of the elastic search will be stored at the following location - INSTALLER_PATH/appviewx_kubernetes/statistical_data_backup. The steps for Elastic Restore are explained in the section elastic restore

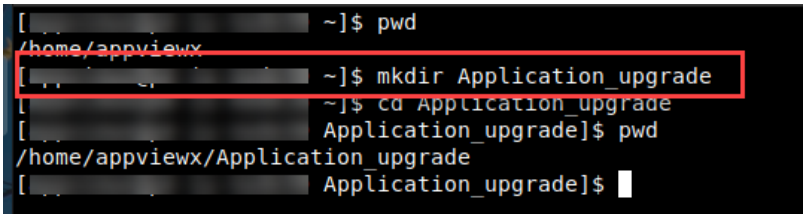
# Chapter 2: Upgrading AppViewX to 2022.1.0 FP5

You can now upgrade to AppViewX version 2022.1.0 FP5 if you are currently using the following versions of AppViewX in RHEL (8.7, 8.8, 8.10, 9.5) or Ubuntu (20.04) OS:

- 2020.3.0 FP10-F11
- 2021.1.0 FP3
- 2022.1.0 FP1-FP4

1. Log in to the release portal and download the installer and addons file –
   - **appviewx_kubernetes_2022.1.5.tar.gz**
   - **appviewx_kubernetes_addons_2022.1.5.tar.gz**
2. Create a new folder in the same location as the existing installer directory.

   **Example**: /home/appviewx/Application_upgrade/




3. Copy the installer file **appviewx_kubernetes_2022.1.5.tar.gz** into the new folder location /home/appviewx/Application_upgrade/.
4. Untar the installer file using the command below:

   ```
   tar –xf appviewx_kubernetes_2022.1.5.tar.gz
   ```

   After the command is executed, the **appviewx_kubernetes** folder is created: /home/appviewx/Application_upgrade/appviewx_kubernetes/
5. Copy the **appviewx_ kubernetes_addons_2022.1.5.tar.gz** file to the **appviewx_kubernetes** folder using the command:

   ```
   mv appviewx_kubernetes_addons_2022.1.5.tar.gz appviewx_kubernetes/
   ```
6. Navigate to the **scripts** directory in the **appviewx_kubernetes** folder (/home/appviewx/Application_upgrade/appviewx_kubernetes) using the command:

   ```
   cd /home/appviewx/Application_upgrade/appviewx_kubernetes/scripts
   ```

   The **scripts** folder contains the **appviewx.conf.template** file.
7. Update the conf parameters in the appviewx.conf file as mentioned in the Install and Upgrade Maintenance guide after copying the **appviewx.conf.template** file as **appviewx.conf**

   To copy, use the command:

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

OR

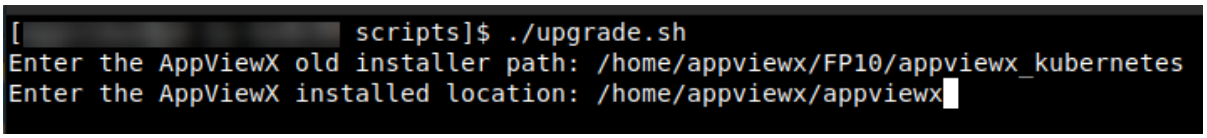
Skip the above step to use the conf merge feature as part of step 10b.

> **Note:**
> - For the complete list of the appviewx.conf file parameters refer the *Configuring the appviewx.conf File to Install Appviewx* section in the **Install and Upgrade Maintenance Guide**.

8. From the /home/appviewx/ApplicationUpgrade/appviewx_kubernetes/scripts directory execute the upgrade command below:

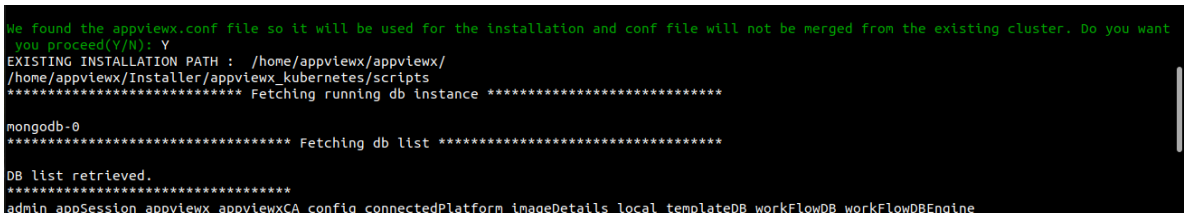
```
./upgrade.sh
```

9. Provide the input of the older installer directory and the directory where the application is currently installed.

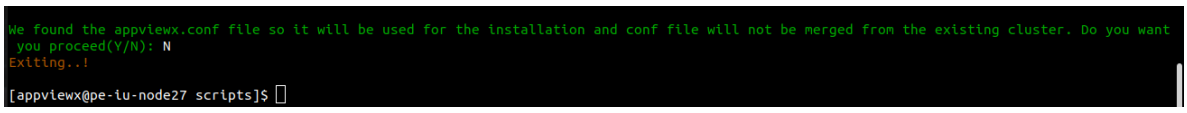



After entering both inputs, the system checks for newly introduced conf file parameters.

10. You will now be prompted with the message about the presence of the conf file, answer Y/N as follows:
    a. If the updated conf file is available in the installer folder, and you choose **Y**, the upgrade proceeds.



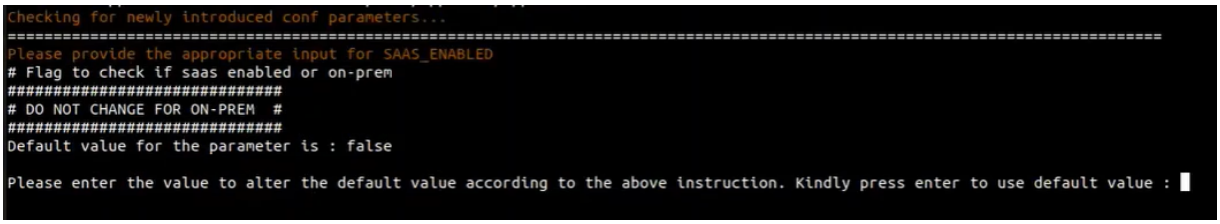

    b. If the updated conf file is available in the installer folder, and you choose **N**, the upgrade stops/exits.




To continue with the upgrade

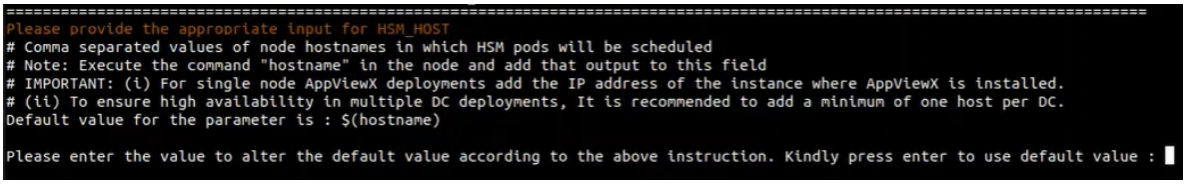
- Edit the conf file and resume the upgrade.
- Delete the conf file from the installer location and resume the upgrade (the upgrade script will handle the merging of the new conf parameters).

11. Enter the appropriate value to alter the default value OR hit the enter key (*recommended*) to use the default value. An example is shown below.




a. To enable msp, the default value is False. Hit the enter key to select the default value and continue.

b. For the parameters HSM_HOST and REDIS_HOST enter the value as follows:




   i. If you have a single node, hit Enter for the default value or the IP address of the instance where AppViewX is installed.
   ii. If you have a multi-node setup, you must enter one hostname per DC of the worker nodes.

c. To configure the BACKUP_CRONJOB_SCHEDULE, enter the values in double quotes. For example, “0 4 * * * states that the cron job will run at 4:00 AM every day.

d. To configure BACKUP_CRONJOB_RETENTION, enter an inter value. For example, 5, which means that the system will keep the last 5 backups and delete any older ones.

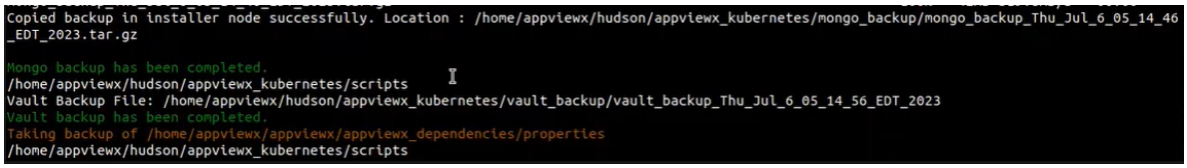
e. To configure the SECONDARY_DB_BACKUP, set the value to true, if DB backup has to be taken from the secondary shared DB.

f. To configure EXTERNAL_GATEWAY_HOST, enter one of the ingress host's hostname in which the external gateway is to be deployed.

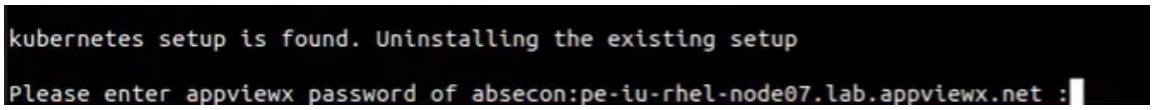
g. For SENTINAL_DC enter the value as follows (only for 2-DC setup):
   i. If it's not a 2-DC setup, enter any one of the DCs.
   ii. If you have a multi-node 2-DC setup, enter the DC which has less number of redis instances than the other DCs.

> **Note:** Ensure you read all the instructions specified in the conf parameters before entering the values.

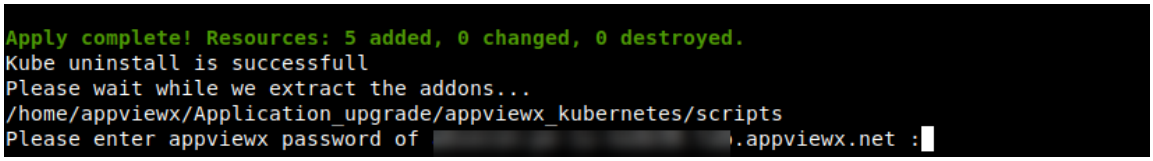
12. The upgrade continues and the following operations are carried out during the process.
    a. **Taking backups of mongo and vault**



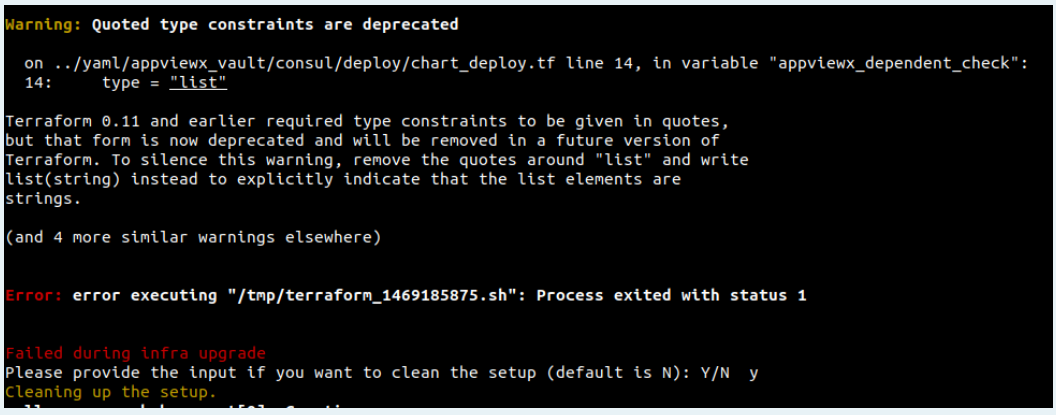

    b. **Uninstalling the old version**
        i. You will be prompted to enter the node password.



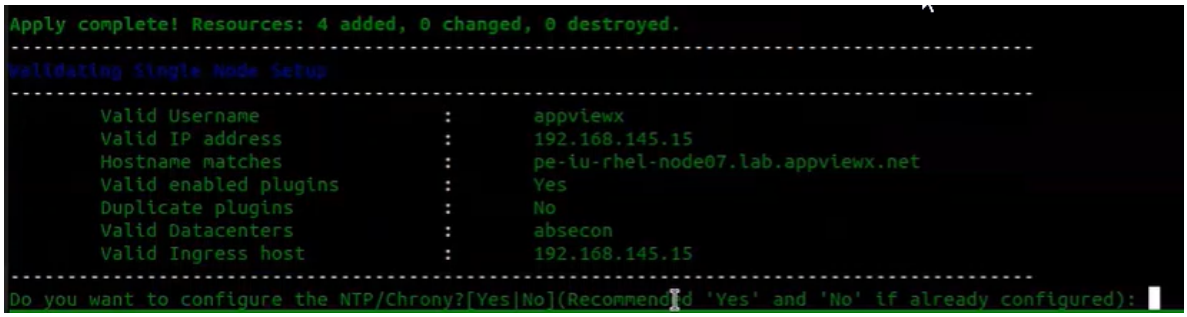

        ii. After the uninstall is complete, you will be prompted to enter the password for the DC host.




> **Note:**
>
> i. In case of upgrade failures, resume the upgrade by executing the command:
>
> ```
> /upgrade.sh
> ```
>
> ii. In case of Infra upgrade failure, the script will prompt a question to clean the setup as shown below. Enter ‘Y’ (yes) to proceed with the clean-up.

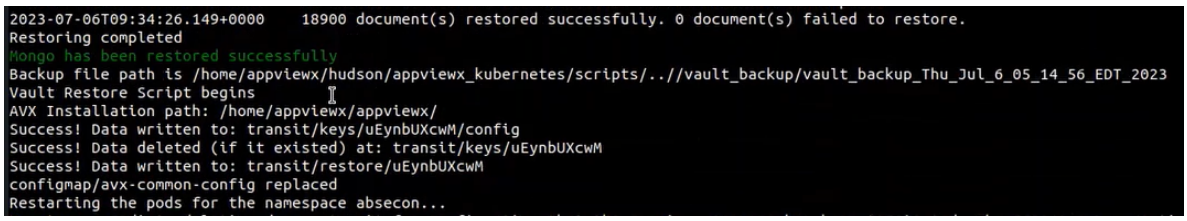



    c. **Time Sync (NTP/Chrony)**




- For a single node - Enter **No** as we do not have to sync time.
- For multi-node - If time sync is already configured before the upgrade then enter **No**. If the time sync for nodes has to be configured then enter **Yes**.

d. **Installing the new version and restoring the backups**




**Note:** In the case of Mongo restore, if the restore operation is stuck or takes more time than usual, then stop the installation process and increase Mongo's **wiredTiger** cachesize of the **mongodb** or **mongo-shareddb statefulset**. (The **mongodb** is for single node setup and **mongo-shareddb statefulset** is for multi-node setup). Use the commands below.

Single Node:

```
kubectl edit statefulset mongodb -n avx
```

Multi node:

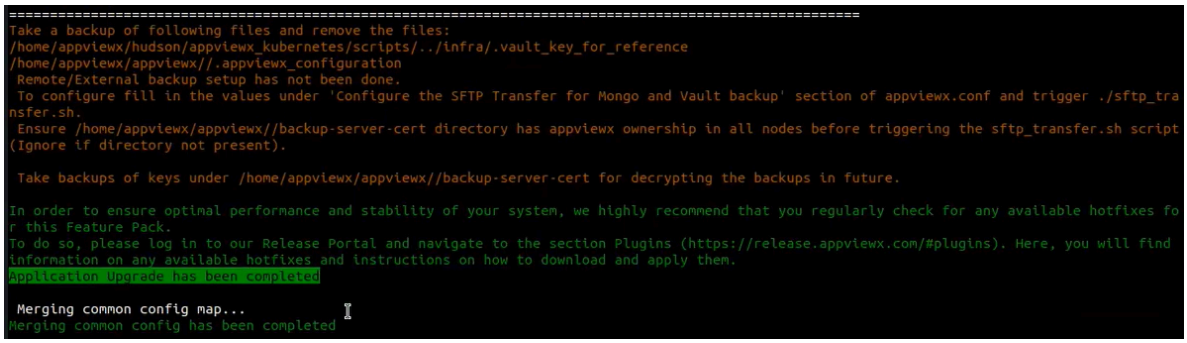
```
kubectl edit statefulset mongo-shareddb -n avx
```

Navigate to **MONGO_CACHE_SIZE** key value of the **env** field and increase the cache size value by 1 to 2 GB

```
        env:
        - name: MONGO_CACHE_SIZE
          value: "0.25"
        image: mongo:4.2.18
        imagePullPolicy: Never
```

After making the required changes run command `./upgrade.sh` to resume the upgrade.

e. **Merging the common config map**




f. After the installation is complete, take a backup of the below files and copy it to a secure location. Then, remove it from the installer location. The files are

- <installer location>/infra/.vault_key_for_reference
- <installer location>/.appviewx_configuration

13. Check the upgraded version and the pods running status.

a. To check the upgraded version, run the following:

```
kubectl get no
```

```
[                              ~]$ kubectl get nodes
NAME                                STATUS   ROLES           AGE   VERSION
                    .appviewx.net   Ready    control-plane   19h   v1.29.1
```

b. To check the pods running status, run the following:

```
kubectl get po -A
```

```
[                          scripts]$ kubectl get po -A
NAMESPACE          NAME                                                READY   STATUS      RESTARTS   AGE
absecon            avx-commons-54885b9d88-vhxdc                        3/3     Running     0          12m
absecon            avx-config-server-6bb868f559-pkc2f                  3/3     Running     0          11m
absecon            avx-platform-core-c4dc4976-296dd                    3/3     Running     0          10m
absecon            avx-platform-logforwarding-5cff778655-txvvg         3/3     Running     0          12m
absecon            avx-platform-queue-bbdfd565c-jljvv                  3/3     Running     0          11m
absecon            avx-platform-report-generator-b7ff97c5d-wlwb8       2/2     Running     0          12m
absecon            avx-subsystems-6f584c6656-n5bsq                     3/3     Running     0          5m
absecon            avx-subsystems-6f584c6656-vgj5n                     3/3     Running     0          5m
absecon            avx-subsystems-sync-678d54df58-tjkwk                3/3     Running     0          12m
absecon            avx-vendor-cert-network-discovery-74bdfcd76d-wpfhg  3/3     Running     0          12m
absecon            avx-vendors-7f6644d889-q5qtl                        3/3     Running     0          12m
absecon            avx-visual-page-builder-65f8c55b4f-mwzrq            2/2     Running     0          12m
avx-jobs           mongoutil-mongoseed-xwcqt                           0/1     Completed   0          20m
avx                avx-config-server-23.1.0.0-db-migration-x66h4       0/1     Completed   0          12m
avx                avx-crontab-5b576c4b59-tq9wc                        3/3     Running     0          12m
avx                avx-platform-core-23.1.0.0-db-migration-d48wf       0/1     Completed   0          12m
avx                avx-platform-gateway-6b6947746b-s8t66               2/2     Running     0          3m16s
avx                avx-platform-queue-23.1.0.0-db-migration-kxs46      0/1     Completed   0          12m
avx                avx-platform-web-8496bdc97f-x4g24                   2/2     Running     0          11m
avx                avx-subsystems-23.1.0.0-db-migration-8hfpt          0/1     Completed   0          12m
avx                crypt-migration-job-hhzhg                           0/1     Completed   0          4m39s
avx                logs-daemon-bs9pw                                   2/2     Running     0          18m
avx                mongodb-0                                           2/2     Running     0          22m
avx                prune-pod-mg4kx                                     2/2     Running     0          12m
avx                redis-0                                             4/4     Running     0          22m
avx                vault-0                                             2/2     Running     0          19m
default            cryptutilencrypt-8t7mx                              0/1     Completed   0          15m
istio-operator     istio-operator-5b6f47d749-twmmb                     1/1     Running     0          24m
istio-system       istio-ingressgateway-5d7cb55c7f-sct97               1/1     Running     0          24m
istio-system       istiod-74d6fc9995-wdr6l                             1/1     Running     0          24m
```

# Chapter 3: Avoiding Upgrade Errors

The following actions must be taken to avoid any post-upgrade errors, if they are applicable in your respective setup:

1. **Loss of Mongo replica set priority configurations**

   During application upgrade, mongodb is freshly set up with the latest upgraded versions. The existing replicaset configurations such as replicaset priorities will not be taken ahead and hence have to be re-configured. High latency customers must perform the following step:

   a. Configure the parameters OPTIMISE_ROUTING_FOR_LATENCY and PREFERRED_DEFAULT_DC in the appviewx.conf
   b. Re-trigger the `plugins_install.sh` to change the configurations.
2. **Custom changes**

   If the custom_changes.yaml, custom_vm_args.conf are present and updated in the custom changes, then the custom changes will be persistent. Any of the custom changes that may have been done specifically for the customer as noted in the prerequisites will not be present if the above mentioned files are not updated with this configuration.
3. **External web cert is not upgraded**

   To update the external CA web certificate, execute the command below:

   ```
   ./appviewx.sh --update-web-cert
   ```

   The following prompts will be displayed:
   • Enter the absolute path of external cert file:
   • Enter the absolute path of external key file:

   Enter both the values to proceed. Once the cert upgrade is completed, restart the gateway and web.
4. Download the **icontrol.jar**, **axis.jar**, and **javax.xml.soap-api-1.4.0.jar** files and copy them to the **external_libs**directory. For details, refer to the section iControl F5 Integration of the *AppViewX Install, Upgrade and Maintenance Guide*.
5. **Setting the ELASTIC_ENABLE as True in the Statistics Configuration**

   There are two ways to do it, choose from either the command prompt (a) or from the management console UI (b).

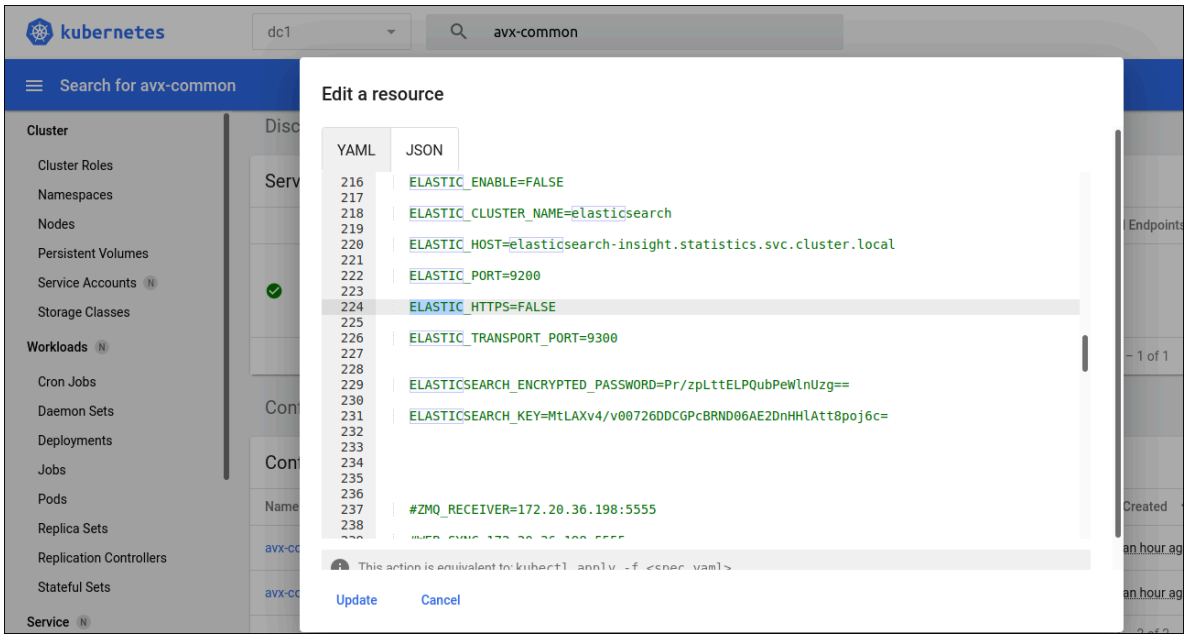
   a. Execute the command

   ```
   kubectl edit configmaps -n <Datacenter1>
   ```

Search for the keyword as Elastic and set ELASTIC_ENABLE as True and below params should have default values as below

```
ELASTIC_ENABLE=TRUE

ELASTIC_CLUSTER_NAME=elasticsearch

ELASTIC_HOST=elasticsearch-insight.statistics.svc.cluster.local

ELASTIC_PORT=9200

ELASTIC_HTTPS=FALSE

ELASTIC_TRANSPORT_PORT=9300
```

b. Login to management console >> Search for the namespace with the configured DC >> Search for avx-common-config in config maps >> Click on Edit and search for Elastic >> Set as True and give as update as shown below.




- Elastic Restore
- Enabling HSM

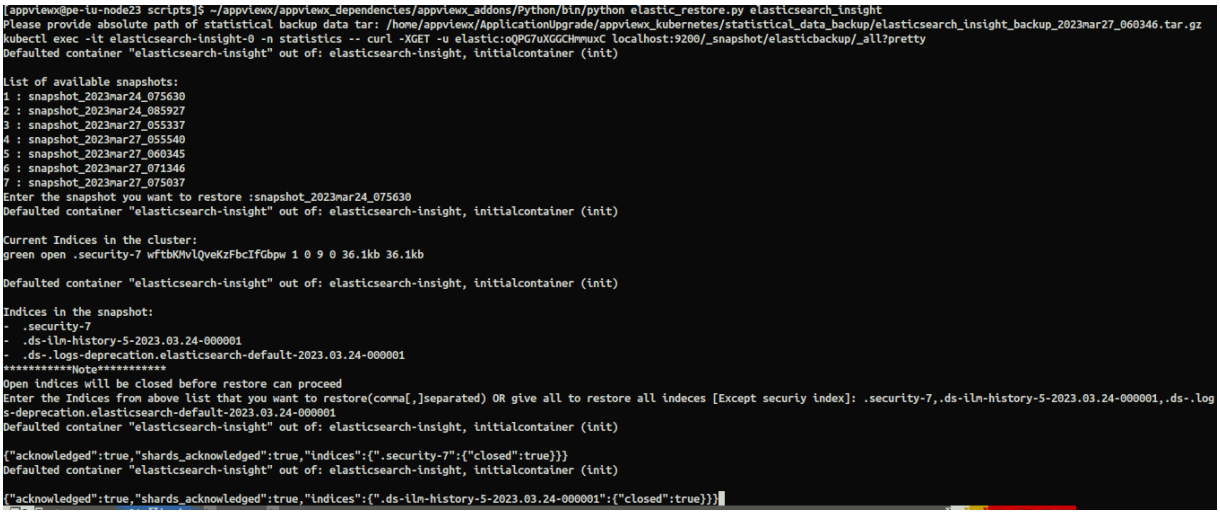
# Elastic Restore

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

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

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

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




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

# Enabling HSM

**Note:** Refer this section only if you have upgraded from AppViewX v2020.3.0.

**Prerequisites**

- To configure Fortanix and Utimaco, the **.so** file and **config** file must be present in the current Appviewx version.
  - The **.so** file is essential for communicating with the HSM using the PKCS11 interface.
  - The **config** file facilitates communication between the HSM and Appviewx.

After the successful upgrade, proceed with the steps below to enable HSM.

1. Ensure the HSM pod is operational and running in the required datacenters and that the HSM node is specified in the appviewx.conf file, execute the following command:

```
kubectl get pods -A -o wide |grep hsm
```

2. From the command line interface, navigate to the properties folder path {APPVIEWX_INSTALLATION_PATH}/appviewx_dependencies/properties

**For Fortanix**

a. Open the HSM file using the following command:

```
vi hsm
```

b. Check and confirm if the HSM file has the following lines. If not, uncomment the following lines:

```
export FORTANIX_PKCS11_CONFIG_PATH= /appviewx/dependencies/hsm/fortanix/pkcs11.conf
```

```
echo "FORTANIX Config Path : $FORTANIX_PKCS11_CONFIG_PATH"
```

c. If the file is edited, restart the **avx-platform-hsm** pod, using the following commands:

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

```
kubectl delete pods -n <namespace> <PodName> --force
```

**For Utimaco**

a. Open the HSM file using the following command:

```
vi hsm
```

b. Check and confirm if the HSM file has the following lines. If not, uncomment the following lines:

```
export CS_PKCS11_R2_CFG=/appviewx/dependencies/hsm/utimaco/cs_pkcs11_R2.cfg
```

```
echo "UTIMACO Config Path : $CS_PKCS11_R2_CFG"
```

c. If the file is edited, restart the **avx-platform-hsm** pod, using the following commands:

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

```
kubectl delete pods -n <namespace> <PodName> --force
```

3. Once the HSM pod is back to running state, login to AppViewX and navigate to **Platform** > **Vault & Security** > **HSM**.
4. Access the required HSM.

- Configuring Fortanix
- Configuring Utimaco
- Verifying/Modifying HSM Configuration for Private Key Encryption

## Configuring Fortanix

1. If the added HSM is Fortanix, upload the **.so** file and **config** file as follows:
   To upload the **.so** file,

a. Click **Browse**.
b. Navigate to the location of the .so file.
c. Select the .so file and click **Open**.

To upload the **.cfg** (config) file,

a. Click **Browse**.
b. Navigate to the location of the .cfg file.
c. Select the .cfg file and click **Open**.

2. Choose the respective datacenter.
3. Update each HSM available in the inventory one by one, ensuring that the HSM is moved to the **Available** status.

## Configuring Utimaco

1. If the added HSM is Utimaco, upload the **.so** file and **config** file as follows:

   To upload the **.so** file,

   a. Click **Browse**.
   b. Navigate to the location of the .so file.
   c. Select the .so file and click **Open**.

   To upload the **.cfg** (config) file,

   a. Click **Browse**.
   b. Navigate to the location of the .cfg file.
   c. Select the .cfg file and click **Open**.

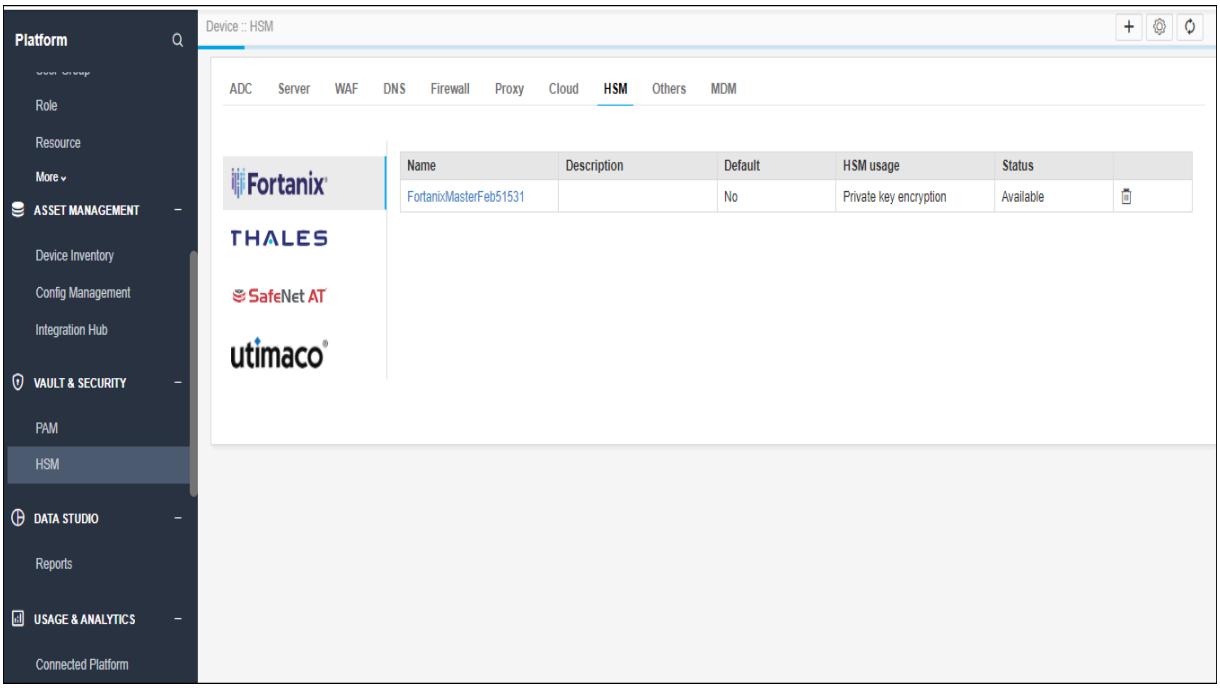
2. Choose the respective datacenter.
3. Update each HSM available in the inventory one by one, ensuring that the HSM is moved to the **Available** status.

If you have previously configured Thales GPN, Thales DPoD, and Thales TCT in the AppViewX v2020.3.0 and have performed the Application Upgrade, then freshly configure the same in the upgraded version. Refer the links below for the configuration:

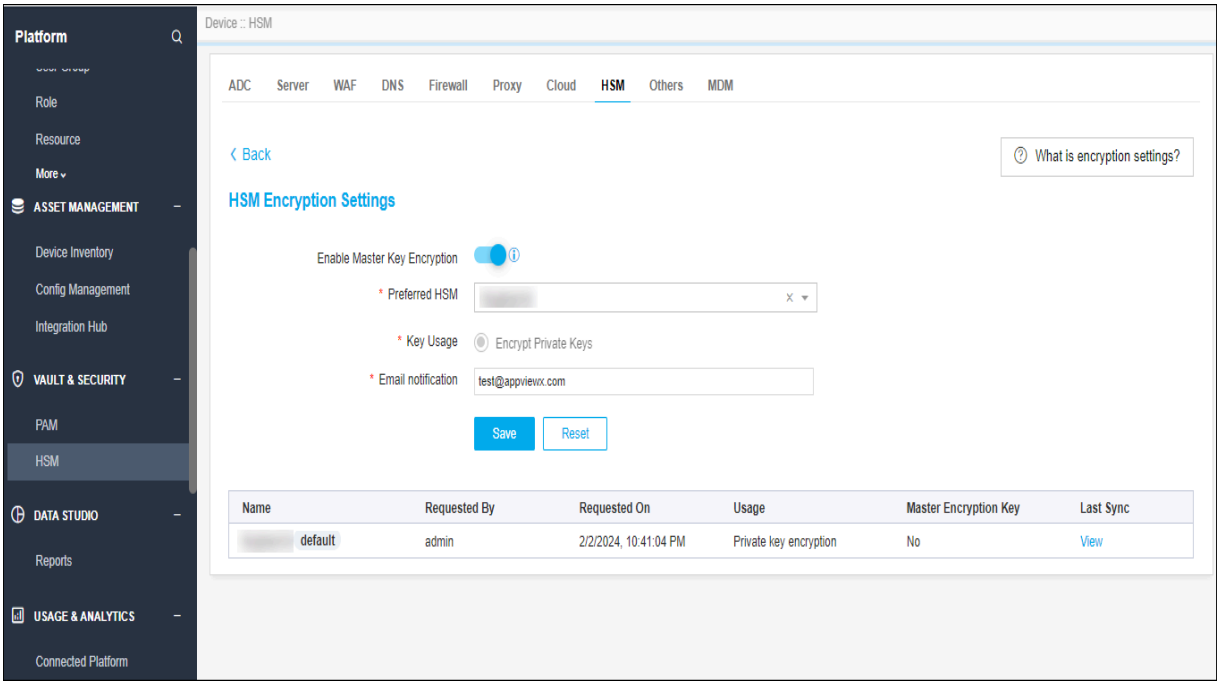
- Thales GPN
- Thales DPoD
- Thales TCT

## Verifying/Modifying HSM Configuration for Private Key Encryption

1. On the top-right corner of the HSM inventory page, click the master encryption settings icon.




2. If the already added HSM has implementation type **private key generation** or **Both** and Set as “default” in 2020.3.0 Appviewx version then that HSM will be available in the Master encryption settings page as default HSM.



3. To receive emails for default HSM health status, configure the email address in the settings page.

> **Note:** For email use case SMTP settings should be available in **Platform** > **System Administration** > **SMTP**.

4. To change the new default HSM, add the new HSM with HSM usage as **Master key encryption** or **both**.
5. Once the HSM is moved to **Available** status it will be present in the **Preferred HSM** dropdown field of the master key encryption settings page.
6. Choose the new HSM and click **Save**.

# Chapter 4: Troubleshooting for Setup Limitations

## Error while installing the AppViewX plugins

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

## Pod Out of Memory

During the mongo restore step of the application upgrade process, the pod may go into an out of memory state (exit code 137 as in the screenshot below). In this case resume the upgrade by rerunning the `upgrade.sh` command.