



Managed Kubernetes - AppViewX Stack Install and Upgrade on AKS

Version: 2022.1.0

Copyright AppViewX, Inc.

Copyright © 2022 AppViewX, Inc. All Rights Reserved.

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

Trademarks

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

External Reference Links

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

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

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

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

Contact Information

AppViewX, Inc.

222 Broadway, FL 19

New York, NY 10038

Email: info@appviewx.com

Web: www.appviewx.com

Contents

Preface.....	4
Revision History.....	4
About this Guide	4
Audience.....	4
Text Conventions.....	4
Chapter 1. AKS Components.....	5
Chapter 2. Deployment Architecture.....	6
Chapter 3. Prerequisites.....	9
Bastion Host Setup.....	9
Azure CLI.....	9
Kubectl.....	9
Helm.....	9
AKS Cluster.....	10
Azure Storage.....	11
Azure Storage Account.....	11
Azure Storage Container.....	11
Azure Service Principal.....	12
Chapter 4. Install AppViewX in Managed Kubernetes.....	14
Chapter 5. Upgrade AppViewX in Managed Kubernetes.....	24
Chapter 6. Uninstall and Cleanup.....	26
Chapter 7. More Information.....	28
Documentation Feedback.....	28
Requesting Technical Support.....	28
Self-Help Online Tools and Resources.....	28

Preface

Revision History

Revision	Description	Date
v1.0	Managed Kubernetes - Appviewx Stack Install and Upgrade on AKS	December 2022

About this Guide

The guide provides information about the AppViewX stack; procedure for installing and accessing the application.

Audience

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

- Platform engineers
- Implementation specialist
- Kubernetes administrators

Text Conventions

The following text conventions are used in this document:

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

Chapter 1: AKS Components

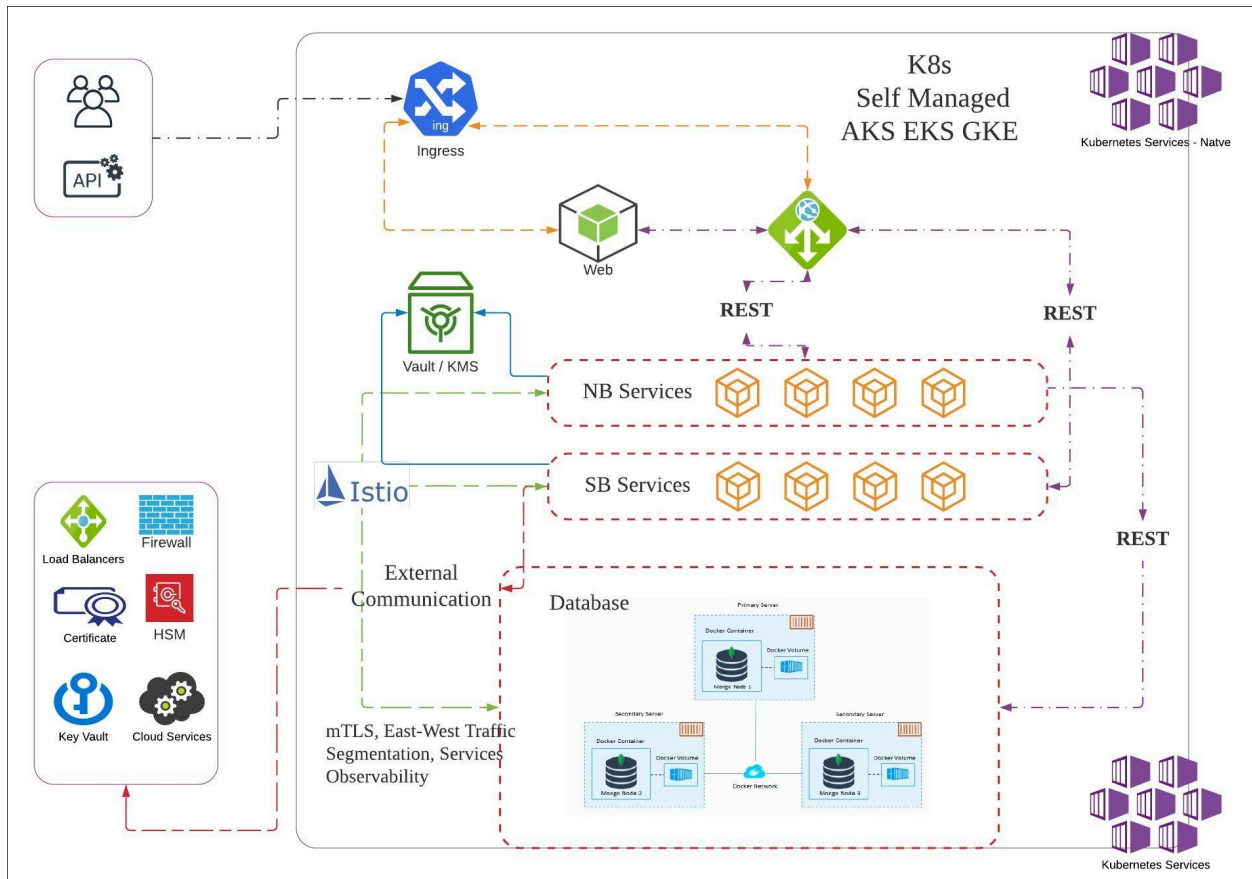
The following AKS components are utilized by AppViewX:

- Azure Kubernetes service
- Storage account for storing mongodb and vault backups
- Service principal for accessing the ACR registry

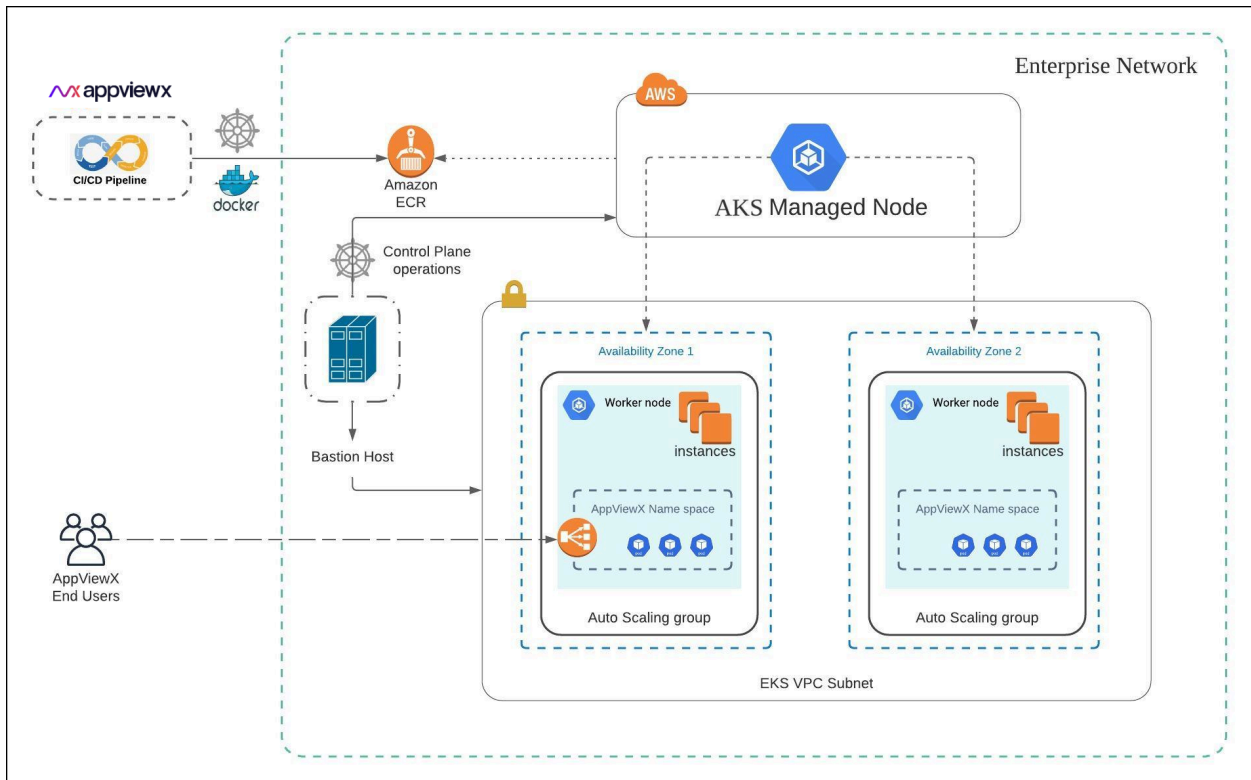
Chapter 2: Deployment Architecture

- Kubernetes Architecture
- AKS Deployment Model
- Cloud Connector

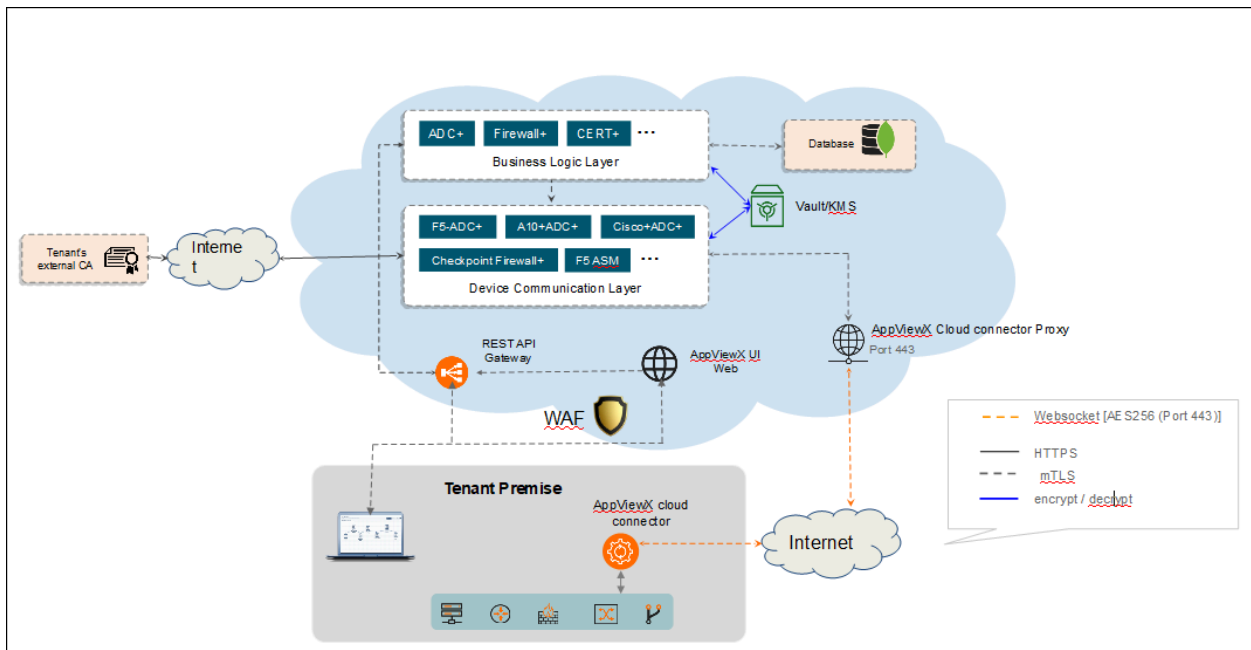
Kubernetes Architecture



AKS Deployment Model



Cloud Connector



For more details on cloud connectors refer to [AppViewX Cloud Connector User Guide](#)

Chapter 3: Prerequisites

- [Bastion Host Setup](#)
- [AKS Cluster](#)
- [Azure Storage](#)
- [Azure Service Principal](#)

Bastion Host Setup

The following packages must be installed on the bastion host or the host/tool from where the installation is triggered

Azure CLI

To set up the Azure CLI refer to [Install the Azure CLI on Linux](#) on the Microsoft documentation website.

Execute the following command:

```
curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash
```

Kubectl

To set up Kubectl refer to [Install and Set Up kubectl on Linux](#) on the Microsoft documentation website.

Execute the following commands

- `sudo curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"`
- `sudo chmod +x kubectl`
- `sudo mv ./kubectl /usr/bin/#`

Verify installation by executing the command

```
kubectl version
```

Helm

Helm is required only if the deployment is triggered from any other machine instead of the DevOps pipeline. To set up Helm refer to [Installing Helm](#) on the Helm documentation website.

Execute the following command:

- ```
curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3
```
- ```
chmod 700 get_helm.sh
```
- ```
./get_helm.sh#
```

Verify installation by executing the command

```
helm version
```

## AKS Cluster

To create an AKS cluster refer to Microsoft's online manual - [Azure Kubernetes Service \(AKS\)](#). Although Microsoft manuals are always up-to-date, the recommended choice to make before creating the cluster is as follows:

- Kubernetes version: 1.22
- The network model:
  - Azure Kubenet (supported by AppViewX).
  - Azure CNI (Recommended for optimal performance).
- Managed identity: System assigned managed identity.
- Enable Kubernetes RBAC.
- **Agent nodepool**: Three nodes of Machine type **D2sv4** with Auto Scaling disabled. Add taint to agent nodepool as **CriticalAddonsOnly=true:NoSchedule** to disable scheduling of application pods to the agent nodepool.



**Note:** The taint **CriticalAddonsOnly=true:NoSchedule** prevents the application pods from being scheduled on system node pools.

- **User nodepool**:
  - **appnodepool**: Three nodes of type **Da8sv4** with Auto Scaling disabled
  - **mongonodepool**: Three nodes of type **Da8sv4** with Auto Scaling disabled. Add label **mongo=true** and taint **designatedMongo=true:NoSchedule** to the nodepool (to be performed while creating the cluster).



**Note:** A minimum of 3 availability zone are needed during cluster creation to support the single AZ failover.

- Select multi zones for the Agent nodepool and the User Nodepool.



**Note:** The number of nodes mentioned here are applicable for managing up-to 25K certs. This number will vary if there are more certificates to manage.

## Azure Storage

### Azure Storage Account

A Storage account is required to store

- iControlJar
- MongoDB backup
- Vault backup

Always create a storage account with a valid name to indicate the storage account for a specific AKS cluster. A typical naming convention could be **<clusternamestorage>**.



**Note:** Storage account access by the AKS pods is granted by the storage account connection string.

For more information on the Azure storage account, refer to Microsoft's online manual - [Create a storage account](#).

### Azure Storage Container

The following containers must be created in the storage accounts that have already been setup.

1. mongodb-backup: Backup job stores mongodb backup into the container.
2. vault-backup: Backup job stores vault backup into the container.

For more information on the Azure storage containers, refer to Microsoft's online manual - [Quickstart: Upload, download, and list blobs with the Azure portal](#).

## Azure Service Principal

Azure Service Principal is used to create the image pull secrets and download images from ACR. When creating the Service Principal

- Set the expiration to never
- Assign the ACR pull access role

For information on commands to create the Azure Service Principal refer to Microsoft's online manual - [Azure Container Registry authentication with service principals](#).

A summary of the steps is given below:

- To derive the service principal id and the password, execute the commands below in the Azure CLI
- Provide input to the parameter ACR\_NAME and run the script
- Store the output of the script and pass as annotation in the global utility config

```
#!/bin/bash

Modify for your environment.

ACR_NAME: The name of your Azure Container Registry
ACR_NAME=<container-registry-name>

SERVICE_PRINCIPAL_NAME=avx-acr-service-principal

Obtain the full registry ID for subsequent command args
ACR_REGISTRY_ID=$(az acr show --name $ACR_NAME --query "id" --output tsv)

Create the service principal with rights scoped to the registry.
Default permissions are for docker pull access. Modify the '--role'
argument value as desired:
acrpull: pull only
acrpush: push and pull
owner: push, pull, and assign roles

PASSWORD=$(az ad sp create-for-rbac --name $SERVICE_PRINCIPAL_NAME --scopes $ACR_REGISTRY_ID --role acrpull --query "password" --output tsv)
USER_NAME=$(az ad sp list --display-name $SERVICE_PRINCIPAL_NAME --query "[].appId" --output tsv)

Output the service principal's credentials; use these in your services and
applications to authenticate to the container registry.

echo "Service principal ID: $USER_NAME"
echo "Service principal password: $PASSWORD"
```



**Note:** After the script execution, capture the outputs as they are required in the global utility config.

# Chapter 4: Install AppViewX in Managed Kubernetes

- [Migration Strategy](#)
- [Installation Steps](#)
- [Post-Installation Steps](#)

## Migration Strategy

To migrate from AppViewX legacy versions (2022.1.0, 2021.1.0, and 2020.3.0) to Managed Kubernetes, take a backup of the mongodb and vault in the respective legacy versions.



**Note:** Refer to the specific version of the release documents from the [release portal](#) and perform the backups or contact the AppViewX support team.

After performing the backup, follow the installation steps detailed in the section below. At step 11 of the installation process, ensure to restore the data at this stage.

## Installation Steps

This section describes the steps to for installing the AppViewX Stack on AKS.

1. Download the installer from the release portal (link to be shared post release).
2. Create a directory **Managedk8s-installer** in the bastion host and extract the installer file **tar -xf installer.tar.gz** in the same directory.
3. Verify that the extracted installer must have the following files
  - appviewxctl (binary)
  - helm\_charts (directory of helm charts)
4. Generate the configuration files based on the cloud provider. If the cloud provider is **Azure**, execute the command below.

```
./appviewxctl config generate --provider azure
```

5. Verify that the execution of the above command creates the configuration files named **.appviewxctl.yaml** in the same location.
6. The file `.appviewxctl` will be populated with the fields necessary for installation, in particular cloud provider that was provided in previous command (**-- provider**).


7. Edit the .appviewxctl.yaml file and populate the values as described below:

**Table 1.**

| Parameters                                  | Description of Values                                                                                                                                                                                                                                     |
|---------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>chartPath</b>                            | The path to the helm_charts which is to be installed. It points to the helm_charts directory extracted in step 3.                                                                                                                                         |
| <b>configFile</b>                           | The path to the kube config file to be used by helm and kubectl.<br><br>If the bastion host is already configured and kube config is under <b>\$HOME/.kube</b> directory, then keep this field empty.                                                     |
| <b>install.enableAppBackupCron</b>          | Boolean value to enable/disable the backup cronjobs. (True/False).<br><br>This value is needed for self-managed mongodb only. For atlas backup this has to be scheduled in the atlas dashboard.                                                           |
| <b>install.enablePrivateImagePullSecret</b> | Boolean value to enable image pull secret.<br><br>Set values as <b>false</b> if the cluster already has access to the container registry.<br><br>Otherwise set it to <b>true</b> and fill all the details of the access keys described in below sections. |
| <b>install.enableThirdPartyInstall</b>      | Boolean value to whether third party monitoring components such as ELK needs to be installed. (True/False)                                                                                                                                                |
| <b>install.imageRegistry</b>                | The URL of the container registry where the images are to be pulled from by the pods.<br><br><i>Example:</i> appviewx.azureacr.io                                                                                                                         |
| <b>install.imageTag</b>                     | The tag of the image that will be used for installation.                                                                                                                                                                                                  |

| Parameters                         | Description of Values                                                                                                                                                                                                                                                                                                                                                                              |
|------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                    | <i>Example:</i> 2022.1.0_FP_750-alpine                                                                                                                                                                                                                                                                                                                                                             |
| <b>install.isSaaSEnabled</b>       | Boolean value for SaaS enablement. This value should be set to <b>true</b> for Managed K8s.                                                                                                                                                                                                                                                                                                        |
| <b>install.kafkaCloudConnector</b> | <p>It is a combination of three values.</p> <ul style="list-style-type: none"> <li>• enable</li> <li>• password</li> <li>• user</li> </ul> <p>Set <b>enable</b> to <b>true</b> and keep the user, password fields empty for Managed K8s.</p> <p><i>Example</i></p> <pre>kafkaCloudConnector:   enable: true   password: ""   user: ""</pre>                                                        |
| <b>install.mongo</b>               | It is a combination of fields specific to the type of mongodb used.                                                                                                                                                                                                                                                                                                                                |
| <b>dbIsolation</b>                 | <p>Boolean value to indicate whether the database isolation is to be enabled.</p> <p>In order for database isolation to work, the following prerequisite must be taken care of while creating the cluster node group.</p> <ul style="list-style-type: none"> <li>• Add label <b>mongo=true</b> and taint <b>designatedMongo=true:NoSchedule</b> to the nodepool to be used for mongodb.</li> </ul> |
| <b>mongoAtlas</b>                  | <p>The fields specific to mongo atlas are as follows:</p> <ul style="list-style-type: none"> <li>• <b>enable:</b> Boolean value to decide if mongo atlas to be used. If set to <i>false</i>, a self managed mongo cluster will be created. If set to <i>true</i></li> </ul>                                                                                                                        |

| Parameters                                     | Description of Values                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                | <p>mongo atlas will be used and details of which are to be provided in below mentioned fields.</p> <ul style="list-style-type: none"> <li>• <b>host:</b> URL of the mongodb atlas cluster.</li> <li>• <b>password:</b> password of the mongodb atlas cluster.</li> <li>• <b>user:</b> username in the mongodb atlas cluster.</li> </ul> <p><i>Example:</i></p> <pre> mongo:   dbIsolation: false   mongoAtlas:     enable: true     host: "managed-k8s.test.mongodb.net"     password: "samplepassword"     user: "user1"                     </pre>                                                                                                                                                          |
| <p><b>install.useDockerPrivateRegistry</b></p> | <p>Set this to <b>true</b> if the dockerhub private repository is to be used for pulling the necessary images needed. Otherwise set the value <b>false</b> and the container registry ACR, ECR, and GCR will be used based on the cloud provider.</p> <p>If this value is set to <i>true</i>, populate the below values, otherwise keep it empty.</p> <ul style="list-style-type: none"> <li>• <b>dockerhub.pass:</b> password to be used for authenticating in the dockerhub private repository.</li> <li>• <b>dockerhub.username:</b> username configured in the dockerhub private repository.</li> </ul> <p><i>Example:</i></p> <pre> useDockerPrivateRegistry: true dockerhub:                     </pre> |

| Parameters                    | Description of Values                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                               | <pre>pass: "testpassword" username: "appviewx"</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <p><b>install.size</b></p>    | <p>The size of the installation. Based on the usecases and number of certs to be managed there different sizes (contact AppViewX for Size recommendations). The sizes supported are (case sensitive values)</p> <ul style="list-style-type: none"> <li>• xsmall</li> <li>• small</li> <li>• medium</li> <li>• large</li> <li>• xlarge</li> <li>• custom</li> </ul> <p><i>Example:</i></p> <pre>size: small</pre> <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note:</b> The size provided must be taken into cluster creation and nodegroup sizes must be defined accordingly. Follow the same document link above for nodegroup sizes.</p> </div> |
| <p><b>install.plugins</b></p> | <p>The list of plugins that will be installed. Each plugin will have three fields</p> <ul style="list-style-type: none"> <li>• enable</li> <li>• imageTag</li> <li>• name</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

| Parameters | Description of Values                                                                                                                                                                                                                                                                                                                            |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|            | <p>Set enable to <b>true</b> if the plugin is to be installed. If the same image tag is to be used as defined in the global ImageTag keep it <b>latest</b> otherwise override with some other tag of your choice.</p> <p><i>Example:</i></p> <pre data-bbox="846 527 1414 674">- enable: true   imageTag: latest   name: avx-config-server</pre> |

The next fields are to be filled with values that must be collected during the cluster creation and setup process and filled as mentioned below.

| Parameters                            | Description of Values                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|---------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>install.privateImagePullSecret</b> | <p>In this section populate the details of the access keys needed to authenticate and pull the image from the registry. They are not needed if the Dockerhub is used as described above.</p> <ul style="list-style-type: none"> <li>• <b>registry:</b> The ACR registry URL</li> <li>• <b>servicePrincipalPassword:</b> The service Principal Password for accessing the registry.</li> <li>• <b>servicePrincipalUsername:</b> The service Principal Username for accessing the registry.</li> </ul> <p><i>Example:</i></p> <pre data-bbox="639 1430 1414 1619">registry: "appviewxsample.azurecr.io" servicePrincipalPassword: "qLPUSA4R1ALkA-GH6m4v70iAC_jajEo9T" servicePrincipalUsername: "20892076-ct8a-4700-a7c0-178u066q9a9c"</pre> |
| <b>install.storageAccess</b>          | The storage bucket details to be used for setting up backup capability.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

| Parameters | Description of Values                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|            | <ul style="list-style-type: none"> <li>• <b>bucketObject</b>: The storage bucket access string.</li> <li>• <b>serviceAccountAnnotation</b>: "none"</li> </ul> <p><i>Example:</i></p> <pre> bucketObject:   "DefaultEndpointsProtocol=https;AccountName=sampleappviewx;AccountKey=Qy0SKtry2MR4Ik0   OIG+po3p0Kgl7u4KEjYo10jHYdVIZXP2/v4IMomkZK6s58YLSLbzcutkyjHJINuCo2Y7w==;EndpointS   uffix=core.windows.net" serviceAccountAnnotation: "none" </pre> |

8. Once the values are filled in `.appviewxctl` as described in the step above, proceed with the installation. Before doing so, check if the the preconditions are met by executing the command

```
./appviewxctl preflight --config .appviewxctl.yaml
```

This will prompt if the necessary prerequisites are met.

9. The metrics server in the Azure clusters comes pre-installed with the cluster, hence they must be disabled from the `avx_pre_req` chart.

a. Navigate to [helm\\_charts/avx\\_pre\\_req](#).

b. Edit the `values.yaml` file by setting the following parameters.

```

avx-metrics-server:
 enable: false

```

The metrics server installation is disabled.

10. To proceed with installation, execute the command

```
./appviewxctl install --config .appviewxctl.yaml
```



**Note:** Let the installation proceed to completion until you see the following message:

```
[Install] Successfully installed Appviewx infra stack
```

This would imply the completion of infra component setup.

11. This step involves restoring the existing data from the previous AppViewX version's cluster in case there is a need to migrate from the older versions to the Managed K8s version. **Ignore this step if it's a fresh setup with no migration necessary.**

To restore mongodb and vault fetch the backup files and place them in the bastion in a directory such as `/home/user/backup` execute the `mongo_restore` and `vault_restore` scripts as follows:

```
./mongo_restore.sh <path to the mongo backup tar file>
./vault_restore.sh -p <path to the vault backup file>
```



**Note:** The above commands work for a self-managed mongodb setup. Setting up the mongodb atlas requires the installation of mongodb tools in the bastion host as follows:

For an rpm based OS:

```
echo -e "[mongodb-org-4.2] \nname=MongoDB
Repository\nbaseurl=https://repo.mongodb.org/yum/redhat/\$releasever/mongodb-org/4.2/x86_64/\ngpgcheck=1\nenabled=1\ngpgkey=https://
www.mongodb.org/static/pgp/server-4.2.asc" > /etc/yum.repos.d/mongodb-org-4.2.repo
yum install mongodb-org-shell-4.2.0
yum install mongodb-org-tools-4.2.0
```

For a debian based OS:

```
wget -qO - https://www.mongodb.org/static/pgp/server-6.0.asc | sudo apt-key add -
sudo apt-get install gnupg
wget -qO - https://www.mongodb.org/static/pgp/server-6.0.asc | sudo apt-key add -
echo "deb [arch=amd64,arm64] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0 multiverse" | sudo
tee /etc/apt/sources.list.d/mongodb-org-6.0.list
sudo apt-get update
sudo apt-get install -y mongodb-mongosh
sudo apt-get install -y mongodb-org-tools
```

Verify if the mongo restore commands have executed successfully using the command

```
mongorestore -- version
```

12. To proceed with the AppViewX application installation, execute the command:

```
./appviewxctl installapp --config .appviewxctl.yaml
```

Once installation is complete the following messages are displayed:

```
[Install] Appviewx infrastructure chart [avx-app] installed successfully
[Install] Successfully installed Appviewx application stack
[Install] Fetching login URL for app
[Install] Waiting for Public IP allotment for istio service
[Install] AppViewX Web URL: https://34.100.197.159/appviewx/
[Install] AppViewX Gateway URL: https://34.100.197.159/avxmgr/
[Install] Grafana URL: https://34.100.197.159/grafana/
[Install] Kibana URL: https://34.100.197.159/kibana/login
[Install] Run below commands to get mongo user credentials
export MONGO_USER=$(kubectl get secret -n avx mongo-key -o=jsonpath='{.data.mongo-init-user}' | base64 -d)
export MONGO_PASS=$(kubectl get secret -n avx mongo-key -o=jsonpath='{.data.mongo-init-pass}' | base64 -d)
[Install] Run below commands to get Elasticsearch and Kibana credentials
export ES_PASS=$(kubectl get secret -n avx elasticsearch-pw-elasticsearch -o=jsonpath='{.data.password}' | base64 -d)
export KIBANA_PASS=$(kubectl get secret -n avx elasticsearch-pw-kibana -o=jsonpath='{.data.password}' | base64 -d)
[Install] Application Installation completed successfully
```



**Note:** Follow the URLs and commands given in the output message to get the credentials and access the application.

13. If installation of the third party monitoring components was not enabled during the entire process, they can be installed later by the following steps:

- a. While installing the third party components ([helm\\_charts/avx\\_third\\_party/values.yaml](#)), the only that values are set to 'true' by default are - *prometheus*, *nodeexporter*, *kube-state metrics*. The other components are set as 'false' by default and must be to set to true if they are to be enabled, they are - *elk-elasticsearch*, *elk-filebeat*, *elk-kibana*, *elk-logstash*, *grafana*, *elasticsearch-insight*, *logstash-syslog*.
- b. Edit the `.appviewxctl.yaml` file and set `install.enableThirdPartyInstall` to 'true'
- c. Run the command `./appviewxctl installtpt --config .appviewxctl.yaml`.

## Post-Installation Steps

INGRESS\_HOSTS and INGRESS\_PORT variables should be added in `appviewx.properties` which is embedded in `avx-common-config` config map for getting it exposed inside the containers.

To add these properties manually in the config map, follow the steps below:

1. Take a backup of current config map using the command

```
kubectl get cm avx-common-config -n avx -o yaml > avx-common-config.yaml
```

2. Execute the command:

```
kubectl get svc -n istio-system istio-ingressgateway
```

Make a note of the **EXTERNAL-IP** of the Load-balancer. A sample output is displayed below.

| NAME                 | TYPE         | CLUSTER-IP   | EXTERNAL-IP   | PORT(S)                    | AGE |
|----------------------|--------------|--------------|---------------|----------------------------|-----|
| istio-ingressgateway | LoadBalancer | 10.76.14.241 | 35.200.143.48 | 80:31540/TCP,443:32056/TCP | 7h  |

In the sample output shared the EXTERNALIP is "35.200.143.48"

3. Edit the config map and add the parameters at EOF. Use the command below.

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

Add the above parameters after the DATACENTER\_LIST variable like below

```
\nDATACENTER_LIST=absecon\n\nINGRESS_HOSTS=<LB_URL>\n\nINGRESS_PORT=443
```



**Note:** Replace <LB\_URL> with the EXTERNAL-IP fetched in step 2 above.

4. After saving the config map, restart all the pods which are prefixed with **avx**. Use the command.

```
kubectl delete pod -n avx --force $(kubectl get pods -n avx | grep avx | '{print $1}')
```

# Chapter 5: Upgrade AppViewX in Managed Kubernetes

To upgrade AppViewX with a new image version, follow the steps below:

1. Ensure to take a backup of the MongoDB and Vault for rollback in case something goes wrong during upgrade.

For self-managed mongoddb:

```
kubectl create job --from=cronjob/mongo-backup -n avx mongo-backup-<unique-identifier>
```

```
kubectl create job --from=cronjob/vault-backup -n avx vault-backup-<unique-identifier>
```


Replace <unique-identifier> in above commands with some random string and run. Monitor the pods until completion and verify the backups are placed in the storage bucket.



**Note:** Atlas backup must be taken in the atlas dashboard. Refer to the atlas snapshots section in the page [Backup and Restore](#).

2. Navigate to the installer directory.
3. Edit the **.appviewxctl.yaml** file's upgrade section for the parameters mentioned below.

| Parameters                   | Description of Values                                                                                                         |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| <b>upgrade.imageRegistry</b> | The URL of the container registry where the images are to be pulled from by the pods.<br><i>Example:</i> appviewx.azureacr.io |
| <b>upgrade.imageTag</b>      | The tag of the image that will be used for installation.<br><i>Example:</i> 2022.1.0_FP_750-alpine                            |
| <b>upgrade.isSaasEnabled</b> | Boolean value for SaaS enablement. This value should be set to <b>true</b> for Managed K8s.                                   |
| <b>upgrade.plugins</b>       | The list of plugins that will be installed. Each plugin will have three fields                                                |

| Parameters | Description of Values                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|            | <ul style="list-style-type: none"> <li>• enable</li> <li>• imageTag</li> <li>• name</li> </ul> <p>Set enable to <b>true</b> if the plugin is to be upgraded. If the same image tag is to be used as defined in the global ImageTag keep it <b>latest</b> otherwise override with some other tag of your choice.</p> <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin: 10px 0;">  <b>Note:</b> The list of plugins to be enabled should match the ones in the install section.         </div> <p><i>Example:</i></p> <pre style="background-color: #f0f0f0; padding: 5px;">- enable: true imageTag: latest name: avx-config-server</pre> |

#### 4. After editing the file, execute the command

```
./appviewxctl upgrade --config .appviewxctl.yaml
```

### Rollback Steps

1. Restore the DB using the restore scripts (step 11 in the Installation Steps section) for self-managed DB or in atlas using snapshot restore in the dashboard.
2. Update the **.appviewxctl.yaml** upgrade section's values to the previous image tag and re-run the upgrade command.

## Chapter 6: Uninstall and Cleanup

The process of uninstalling requires one to navigate to the installer directory and execute the following command

```
./appviewxctl uninstall --config .appviewxctl.yaml
```

The following messages are displayed after the uninstall command is executed successfully.

```
1 ./appviewxctl uninstall --config .appviewxctl.yaml
2
3 [Init] Using log file at [/avx/appviewxctl-3196327299.log] to dump logs
4 [Init] Initialise persistent flag config
5 [Init] Using config file
6 [Uninstall] Uninstalling appviewx application
7 [Uninstall] Uninstalling Appviewx application helm chart
8 [Uninstall] Uninstalling application backup helm chart
9 [Uninstall] Uninstalling Infra application helm chart
10 [Uninstall] Uninstalling Third party application helm chart
11 [Uninstall] Uninstalling IstioOperator from the cluster
12 [Uninstall] Uninstalling PVCs from the avx namespace
13 [Uninstall] Uninstalling Pre-requisite helm chart
14 [Uninstall] Uninstalling Appviewx installed namespaces
15 [Uninstall] Successfully uninstalled appviewx application and all the related
```



**Note:** In the Managed K8s environments removal of PVCs do not occur at times as it may require patching PVCs first before deletion. This may cause certain error messages to display, indicating that PVC has changed. In case such an error occurs, re-run the above command to solve the issue and uninstall the application.

Sometimes the namespaces take a longer time to be removed. Hence, post installation, check if namespaces are in the terminating state (use the command: **kubectl get namespace**). If any namespace is in the terminating state, manually remove the namespaces by executing the commands below:

```
kubectl get namespace "istio-operator" -o json | tr -d "\n" | sed "s/^finalizers\": \[[^\]]+\]^finalizers\": []/" | kubectl replace
--raw /api/v1/namespaces/istio-operator/finalize -f - 2>/dev/null
```

```
kubectl get namespace "istio-system" -o json | tr -d "\n" | sed "s/^finalizers\": \[[^\]]+\]^finalizers\": []/" | kubectl replace
--raw /api/v1/namespaces/istio-system/finalize -f - 2>/dev/null
```

```
kubectl get namespace "avx" -o json | tr -d "\n" | sed "s/^\"finalizers\": \[[^\]]+\]^\"finalizers\": []/" | kubectl replace --raw /api/v1/namespaces/avx/finalize -f -
2>/dev/null
```

```
kubectl delete ns istio-operator --force 2>/dev/null
```

```
kubectl delete ns istio-system --force 2>/dev/null
```

```
kubectl delete ns avx --force 2>/dev/null
```

# Chapter 7: More Information

For the latest, most complete information about known and fixed issues with the AppViewX modules, see the latest revision of the release notes.

To access Software Release Notifications for AppViewX Releases, visit our Help center at <https://help.appviewx.com/home>. You need to log in to your AppViewX account. From the Help center, search by the specific release number or navigate to Release Portal and choose the release, for example, v20.3.0.

## Documentation Feedback

We request you to provide feedback, comments, and suggestions so that we can improve the documentation. You can send your comments to [tech-documentation@appviewx.com](mailto:tech-documentation@appviewx.com)

If you are preferred to send feedback through e-mail, be sure to include the following information with your comments:

- Document or topic name
- URL or page number
- Software release version (if applicable).

## Requesting Technical Support

Technical product support is available through AppViewX help support center, request to send an email to [help@appviewx.com](mailto:help@appviewx.com)

## Self-Help Online Tools and Resources

For quick and easy problem resolution, AppViewX is designed an online self-service portal called the help support center that provides you with the following features:

- Find help support center: <https://help.appviewx.com/home>
- Find product technical documentation: <https://help.appviewx.com/documentation>
- Find solutions and answer questions using our Knowledge Base: <https://internalkb.appviewx.com/knowledge-base>
- Download the latest versions of software: <https://release.appviewx.com>