



Install and Configure Avi Multi-Cluster Kubernetes Operator

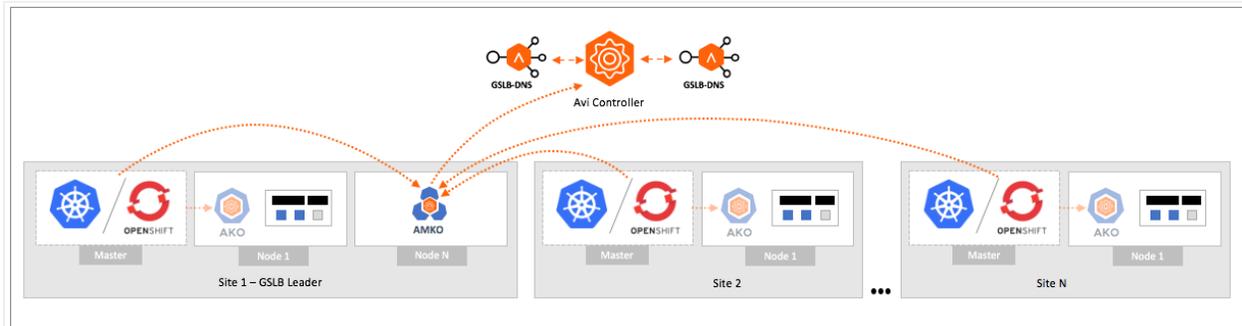
Avi Technical Reference (v21.1)

Install and Configure Avi Multi-Cluster Kubernetes Operator

[view online](#)

Overview

The Avi Multi-Cluster Kubernetes Operator (AMKO) facilitates application deployments across multiple OpenShift /Kubernetes clusters.



AMKO is aware of the following object types:

- Kubernetes:
 - Ingress
 - Service type load balancer
- OpenShift:
 - Routes
 - Service type load balancer

AMKO is a Kubernetes pod that runs in a namespace called `avi-system`.

Installation of AMKO is done via Helm which applies the following permissions via a Kubernetes service account for `GSLBConfig` and `GlobalDeploymentPolicy` objects: `* CREATE * GET * LIST * READ * UPDATE`

AMKO also requires permissions to read the ingress and service objects for all the member clusters. For this, a separate `kubeconfig` file is created with all the required permissions from all the clusters and is passed to AMKO via an OpenShift /Kubernetes `secret` object.

To know more, refer to the following articles:

- [Avi GSLB Overview](#)
- [Avi GSLB Site Configuration and Operations](#)
- [Avi Multi-Cluster Kubernetes Operator](#)

Requirements

This section explains the minimum version requirements to use AMKO in Kubernetes clusters and OpenShift clusters:

For Kubernetes Clusters

The AMKO version 1.4.2 support for Kubernetes/OpenShift is as below:

Component	Versions Supported
Kubernetes	Version 1.16 and higher
AKO	Version 1.4.2
Avi Controller	20.1.4-2p3+

For OpenShift Clusters

Components	Version
OpenShift	4.4+
AKO	1.4.2
Avi Controller	20.1.4-2p3+

Pre-Installation Checklist

Ensure that the following tasks are completed before commencing AMKO installation:

- At least one OpenShift/ Kubernetes cluster
- At least one Avi Controller
- AMKO assumes that it has connectivity to all the member clusters' OpenShift/Kubernetes API servers. Without this, AMKO will not be able to watch over the Kubernetes and OpenShift resources in the member clusters.
- Before deploying AMKO, one of the clusters have to be designated as the leader and rest of the clusters as followers. AMKO has to be deployed on all clusters (wherever federation is required). This is to ensure that the leader cluster's AMKO would federate the AMKO config objects like `GSLBConfig` and `GlobalDeploymentPolicy` objects to all follower clusters. Refer to the [Federation](#) article for more details on the specifics of federation and how to recover from a disaster recovery scenario.
- On all clusters, create the namespace `avi-system`.

```
kubectl create ns avi-system
```

- Create a `kubeconfig` file with the permissions to read the service and the ingress/route objects for all the member clusters. To know more refer [Creating a kubeconfig file for multi cluster access](#). Name this file `gslb-members` and generate a secret with the kubeconfig file in `cluster-amko` as shown below:

```
kubectl create secret generic gslb-config-secret --from-file gslb-members -n avi-system
```

This has to be done for all the member clusters wherever AMKO is going to be deployed.

The permissions provided in the `kubeconfig` file for all the clusters must have at least the permissions to `[get, list, watch]` on: * Kubernetes ingress and service type load balancers. * OpenShift routes and service type load balancers.

AMKO also needs permissions to `[create, get, list, watch, update, delete]` on: * `GSLBConfig` object * `GlobalDeploymentPolicy` object

Installing AMKO

Note: Only Helm version 3.0 is supported.

To install AMKO via Helm,

1. Create the `avi-system` namespace:

```
$ kubectl create ns avi-system
```

2. Add this repository to your helm client:

```
$ helm repo add amko https://projects.registry.vmware.com/chartrepo/ako
```

Note: The helm charts are present in VMWare's public harbor repository.

3. Search the available charts for AMKO:

```
$ helm search repo
```

NAME	CHART VERSION	APP VERSION	DESCRIPTION
ako/amko	1.4.2	1.4.2	A helm chart for Avi Multicluster Kubernetes Operator

4. Configure the parameters using [values.yaml](#) from this repository to provide values related to the Avi configuration. To get the `values.yaml` for a release, run the following command:

```
helm show values ako/amko --version 1.4.2 > values.yaml
```

5. To configure federation via `values.yaml`:

- Set `configs.federation.currentClusterIsLeader` to `true` for the leader cluster. For all follower clusters, set this to `false`.
- Set `configs.federation.currentCluster` to the current cluster context.
- Add the member clusters to `configs.federation.memberClusters`.

6. Install AMKO:

```
$ helm install ako/amko --generate-name --version 1.4.2 -f /path/to/values.yaml --set configs.gslbLeaderCon
<leader_controller_ip>
  --namespace=avi-system
</leader_controller_ip>
```

7. Verify the installation:

```
$ helm list -n avi-system
```

```
NAME      NAMESPACE      REVISION      UPDATED      STATUS      CHART      APP VERSION      amko-1598451370      avi-system      1      2020-08-26      14:16:21.889538175      +0000      UTC      deployed      amko-1.4.2      1.4.2
```

Configuring AMKO

AMKO is configured via two CRDs - `GSLBConfig` and `GlobalDeploymentPolicy`. The Helm based installation procedure will automatically create these two in the specified namespaces.

During AMKO installation, set the required parameters via `values.yaml`. These parameters are translated to the `GSLBConfig` and the `GlobalDeploymentPolicy` objects.

Parameter	Description	Default
<code>configs.controllerVersion</code>	Leader site's controller version	20.1.4
<code>configs.federation.image.repository</code>	Image repository for AMKO federator	<code>projects.registry.vmware.com/ako/amko-federator</code>
<code>configs.federation.image.pullPolicy</code>	Image pull policy for AMKO federator	<code>IfNotPresent</code>
<code>configs.federation.currentCluster</code>	Current cluster context (required)	Not Applicable
<code>configs.federation.currentClusterIsLeader</code>	Set to <code>true</code> if current cluster is the leader (required)	False
<code>configs.gslbLeaderController</code>	GSLB leader site URL	Not Applicable
<code>configs.federation.memberClusters</code>	Member clusters on which federation should be done	Not Applicable
<code>configs.gslbLeaderCredentials.username</code>	GSLB leader controller username	admin
<code>configs.gslbLeaderCredentials.password</code>	GSLB leader controller password	
<code>configs.federation.memberClusters.clusterContext</code>	Kubernetes member cluster context for GSLB	<code>cluster1-admin</code> and <code>cluster2-admin</code>
<code>configs.refreshInterval</code>	Interval in seconds after which the in-memory caches are refreshed	1800 seconds
<code>configs.logLevel</code>	Log level to be used by AMKO to print the type of logs. The supported values are <code>INFO</code> , <code>DEBUG</code> , <code>WARN</code> and <code>ERROR</code>	<code>INFO</code>

<code>configs.useCustomGlobalFqdn</code>	Select the GslbService FQDN mode for AMKO. If set to true, AMKO observes the HostRules to look for mapping between local and global FQDNs	False
<code>gdpConfig.appSelector.label{.key,.value}</code>	Selection criteria for applications, label key and value are provided	Nil
<code>configs.domainNames</code>	Domain names supported in the GSLB configuration	foo.com
<code>gdpConfig.appSelector.label{.key,.value}</code>	Selection criteria for applications, label key and value are provided	Not Applicable
<code>gdpConfig.namespaceSelector.label{.key,.value}</code>	Selection criteria for namespaces, label key and value are provided	Not Applicable
<code>gdpConfig.matchClusters</code>	List of clusters (names must match the names in <code>configs.memberClusters</code>) from where the objects will be selected	Not Applicable
<code>gdpConfig.trafficSplit</code>	List of weights for clusters (names must match the names in <code>configs.memberClusters</code>), each weight must range from 1 to 20	Not Applicable
<code>gdpConfig.ttl</code>	Time To Live, ranges from 1-86400 seconds	Nil
<code>gdpConfig.healthMonitorRefs</code>	List of health monitor references to be applied on all GSLB Services	Nil
<code>gdpConfig.sitePersistenceRef</code>	Reference for a federated application persistence profile created on the Avi Controller	Nil
<code>gdpConfig.poolAlgorithmSettings</code>	Pool algorithm settings to be used by the GslbServices for traffic distribution across pool members. For more information, click here .	GSLB_ALGORITHM_ROUND_ROBIN

With the AMKO installation based on the parameters configured above, Helm creates the objects `GSLBConfig` and `GlobalDeploymentPolicy`.

If `appSelector` and `namespaceSelector` were not specified via the Helm installation (`values.yaml`), the respective fields will be empty in the resulting `GlobalDeploymentPolicy` object.

This means that by default, no objects are selected.

You can edit this `GlobalDeploymentPolicy` object and add the required criteria for app or cluster selection.

To know more about the CRDs used to configure the GSLB services in the GSLB leader site and override specific `GslbService` properties, click [here](#).

Editing the Runtime Parameters of AMKO

The GDP object can be edited at runtime to change the application selection parameters, traffic split and the applicable clusters. AMKO will recognize these changes and will update the `GSLBServices` accordingly.

Changing only `logLevel` is permissible at runtime via an edit of the `GSLBConfig`. For all other changes to the `GSLBConfig`, the AMKO pod has to be restarted.

Choosing a Mode of GslbService FQDN

There can be different requirements to either use local FQDNs as the `GslbService` FQDNs, or use a different FQDN as the Global FQDN. Refer to [Deriving GSLB Service FQDNs](#) to choose a mode fit for the use-case and enable accordingly.

GSLB Service Properties

Certain GSLB Service properties can be set and modified at runtime. If these are set through the GDP object, they are applied to all the Gslb Services. To override specific properties, use a `GSLBHostRule` object for a `GslbService`.

Properties	Configured Via
TTL	GDP, GSLBHostRule
Site Persistence	GDP, GSLBHostRule
Custom Health Monitors	GDP, GSLBHostRule
Third Party Members	GSLBHostRule
Traffic Split	GDP, GSLBHostRule
Pool Algorithm Settings	GDP, GSLBHostRule

To set the properties, follow the steps provided in the [Deriving GSLB Service FQDNs](#) article.

Uninstall AMKO using Helm

To uninstall AMKO, use the command:

```
helm uninstall -n avi-system <amko-release-name>
```

To remove the already created GSLB services, remove the GDP object first. This will remove all the GSLB services selected via the GDP object.

```
kubectl delete gdp -n avi-system global-gdp
```

Also, delete the avi-system namespace:

```
kubectl delete ns avi-system
```

Installing AMKO Offline Using Helm

Pre-requisites for Installation

- The Docker image downloaded from the [Avi Portal](#)
- A private container registry to upload the AMKO Docker image
- Helm version 3.0 or higher installed

Installing AMKO

To install AMKO offline using Helm,

1. Create the GSLB members with the member clusters configuration and generate a secret with the kubeconfig file in cluster-amko as shown below:

```
kubectl create secret generic gslb-config-secret --from-file gslb-members -n avi-system
```

2. Extract the .tar file to get the AMKO installation directory with the helm and docker images.

```
tar -zxvf amko_cpr_sample.tar.gz
amko/
amko/install_docs.txt
amko/amko-1.4.2-docker.tar.gz
amko/amko-1.4.2-helm.tgz
amko/amko-federator-1.4.2-docker.tar.gz
```

3. Change the working directory to this path: `cd amko/`.

4. Load the docker image in one of your machines.

```
sudo docker load < amko-1.4.2-docker.tar.gz
```

5. Push the docker image to your private registry. For more information, click [here](#).
6. Extract the AMKO Helm package. This will create a sub-directory `amko/amko` which contains the Helm charts for AMKO (`amko/chart.yaml crds templates values.yaml`).
7. Update the helm `values.yaml` with the required AMKO configuration (Controller IP/credentials, docker registry information etc).
8. Create the namespace `avi-system` on the OpenShift/Kubernetes cluster.

```
kubectl create namespace avi-system
```

9. Install AMKO using the updated helm charts.

```
helm install ./amko --generate-name --namespace=avi-system
```

Uninstall AMKO via Helm

To uninstall AMKO using Helm,

```
helm uninstall -n avi-system <amko-release-name>
```

To remove the GSLB services created, remove the GDP object first. This will remove all the GSLB services selected via the GDP object.

```
kubectl delete gdp -n avi-system global-gdp
```

Delete the avi-system namespace:

```
kubectl delete ns avi-system
```

AMKO Default Values

A GSLBService created by AMKO has the following default values:

GSLBService Fields	Default Value
Pool load balancing algorithm	Round robin
TTL	Provided by the DNS service (the default value is 30s)
Health Monitor(s)	Refer to the GS Health Monitors section

GSLB Service Health Monitors AMKO creates health monitors to be used for site recovery. The object specific health monitors are as follows:

- **Service Type Load Balancer:** For GSLB services serving service type load balancer, the health monitor is of `System-GSLB-TCP` or `System-GSLB-UDP` depending on the service type.
- **Insecure Ingresses and Insecure Routes:** A custom health monitor of type `System-GSLB-HTTP` is created for each path present in the ingress/route definition and is added to the GSLB service.
- **Secure Ingresses and Secure Routes (excluding passthrough routes):** A custom health monitor of type `System-GSLB-HTTPS` is created for each path present in the ingress/route definition and added to the GSLB service.
- **Secure Passthrough Routes:** A custom health monitor of type `System-GSLB-TCP` is created and shared across all such GSLB Services serving passthrough routes.
- **Custom Health Monitors:** One or more federated health monitor(s) can be created on Avi, and the ref(s) for them can be specified in the `GSLBHostRule` or `GDP` object. In this case, the GSLB Services will be updated to use the referred health monitors instead of the usual path-based health monitors.

Document Revision History

Date	Change Summary
July 29, 2021	Updated the AMKO Installation Guide for Version 1.4.2
May 04, 2021	Updated the AMKO Installation Guide for Version 1.4.1
October 1, 2020	Published the AMKO Installation Guide for Version 1.2.1
July 21, 2020	Published the AMKO Installation Guide (Tech Preview)