



# Troubleshooting Avi Kubernetes Operator

Avi Technical Reference (v18.2)

Copyright © 2020

# Troubleshooting Avi Kubernetes Operator

[view online](#)

## Overview

AKO is a Kubernetes operator which works as an ingress controller and performs Avi-specific functions in a Kubernetes environment with the Avi Controller. It runs as a pod in the cluster and translates the required Kubernetes objects to Avi objects and automates the implementation of ingresses/routes/services on the Service Engines (SE) via the Avi Controller.

This article is a list of troubleshooting steps to use in case there are issues when using AKO.

### 1. AKO Pod Does Not Run

To check why the pod is not running, do the following:

```
kubectl get pods -n avi-system
NAME                READY   STATUS             RESTARTS   AGE
ako-f776577b-5zpxh  0/1     ImagePullBackOff   0           15s
```

Ensure that: \* Your Docker registry is optimally configured.

\* The image is configured locally.

### 2. AKO Does Not Respond to the Ingress Object Creations

Look into the AKO container logs and see if you find a reason on why the sync is disabled like this:

```
2020-06-26T10:27:26.032+0530    INFO    lib/lib.go:56    Setting AKOUser: ako-my-cluster for Avi Objects
2020-06-26T10:27:26.337+0530    ERROR   cache/controller_obj_cache.go:1814  Required param networkName not specified, s
2020-06-26T10:27:26.337+0530    WARN    cache/controller_obj_cache.go:1770  Invalid input detected, syncing will be dis
```

### 3. Ingress Object Does Not Sync in Avi

1. The ingress class is set as something other than `avi`. The `defaultIngressController` parameter is set to `True`.
2. For TLS ingress, the `Secret` object does not exist. Ensure that the `Secret` object is pre-created.
3. Check the connectivity between your AKO Pod and the Avi Controller.

### 4. Virtual Service Returns The Message **CONNECTION REFUSED** After Sometime

This is generally due to a duplicate IP in use in the network.

### 5. Virtual Service Settings Changed Directly on the Avi Vantage Controller is Overwritten

It is not recommended to change the properties of a virtual service by AKO, outside of AKO. If AKO has an ingress update that is related to this shared virtual service, then AKO will overwrite the configuration.

### 6. Static Routes are Populated, but the Pools are Down

Check if you have a dual network interface card (NIC) Kubernetes worker node setup.

In case of a dual NIC setup, AKO would populate the static routes using the default gateway network.

However, the default gateway network might not be the port group network that you want to use as the data network. Hence, the service engines may not be able to reach the pod CIDRs using the default gateway network.

If it is not possible to make your data networks routable via the default gateway, `disableStaticRoute sync` in AKO and edit your static routes with the correct network.

## Log Collection

For every log collection, collect the following information too:

1. What kubernetes distribution are you using? For example, RKE, PKS etc.
2. What is the CNI you are using with versions? For example, Calico v3.15
3. What is the Avi Controller version you are using? For example, Avi Vantage version 18.2.8

### Collecting AKO Logs

To collect the logs, use the script available [here](#) and collect all relevant information for the AKO pod.

The script does the following:

1. Collects the log file of AKO pod
2. Collects the `configmap` in a yml file
3. Zips the folder and returns

The following three cases are considered for log collection:

1. A running AKO pod logging into a Persistent Volume Claim, in this case the logs are collected from the PVC that the pod uses.
  2. A running AKO pod logging into console, in this case the logs are collected from the pod directly.
  3. A dead AKO pod that uses a Persistent Volume Claim, in this case a backup pod is created with the same PVC attached to the AKO pod and the logs are collected from it.

### Configuring PVC for the AKO Pod

It is recommended to use a Persistent Volume Claim for the AKO pod.

To create a persistent volume(PV) and a Persistent Volume Claim(PVC), refer to the [Configure a Pod to Use a Persistent Volume for Storage](#) article.

This is an example of hostpath persistent volume. Use the PV based on the storage class of your kubernetes environment.

1. To create persistent volume,

```
#persistent-volume.yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: ako-pv
  namespace : avi-system
  labels:
    type: local
spec:
  storageClassName: manual
  capacity:
    storage: 10Gi
```

```
accessModes:
  - ReadWriteOnce
hostPath:
  path: <any-host-path-dir> # make sure that the directory exists
```

A persistent volume claim can be created using the following:

```
#persistent-volume-claim.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: ako-pvc
  namespace : avi-system
spec:
  storageClassName: manual
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 3Gi
```

2. Add PVC name into the `ako/helm/ako/values.yaml` before the creation of the AKO pod as shown below:

```
persistentVolumeClaim: ako-pvc
mountPath: /log
logFile: avi.log
```

## Using the Script for AKO

### Use case 1

With PVC, (Mandatory) `--akoNamespace (-ako)` : The namespace in which the AKO pod is present.

```
python3 log_collections.py -ako avi-system
```

### Use case 2

Without PVC (Optional) `--since (-s)` : time duration from present time for logs.

```
python3 log_collections.py -ako avi-system -s 24h
```

### Sample Run:

At each stage of execution, the commands being executed are logged on the screen. The results are stored in a zip file with the format below:

```
ako-<helmchart name>-<current time>
```

**Sample Output with PVC:**

```

2020-06-25 13:20:37,141 - ***** AKO *****
2020-06-25 13:20:37,141 - For AKO : helm list -n avi-system
2020-06-25 13:20:38,974 - kubectl get pod -n avi-system -l app.kubernetes.io/instance=my-ako-release
2020-06-25 13:20:41,850 - kubectl describe pod ako-56887bd5b7-c2t6n -n avi-system
2020-06-25 13:20:44,019 - helm get all my-ako-release -n avi-system
2020-06-25 13:20:46,360 - PVC name is my-pvc
2020-06-25 13:20:46,361 - PVC mount point found - /log
2020-06-25 13:20:46,361 - Log file name is avi.log
2020-06-25 13:20:46,362 - Creating directory ako-my-ako-release-2020-06-25-132046
2020-06-25 13:20:46,373 - kubectl cp avi-system/ako-56887bd5b7-c2t6n:log/avi.log ako-my-ako-release-2020-06-25-132046/a
2020-06-25 13:21:02,098 - kubectl get cm -n avi-system -o yaml > ako-my-ako-release-2020-06-25-132046/config-map.yaml
2020-06-25 13:21:03,495 - Zipping directory ako-my-ako-release-2020-06-25-132046
2020-06-25 13:21:03,525 - Clean up: rm -r ako-my-ako-release-2020-06-25-132046

Success, Logs zipped into ako-my-ako-release-2020-06-25-132046.zip
    
```

**Document Revision History**

Date	Change Summary
June 30, 2020	Published the Troubleshooting Guide for AKO version 0.9.1