AVI Networks®

Frequently Asked Questions for AKO

Avi Technical Reference (v20.1)

Frequently Asked Questions for AKO

view online

Overview

This document answers some of the frequently asked questions for AKO:

## How do I clean up all my configs? The key `deleteConfig` in the data section of AKO `configmap` can be used to clean up the setup.
Edit the AKO `configmap` and set `deleteConfig`: *true* to delete the ako created objects in Avi.
After the flag is set in the `configmap`, a condition with type `ako.vmware.com/ObjectDeletionInProgress` is added in the status of AKO statefulset with reason as `objDeletionStarted` and status *True*. For example:

```
status:


    conditions:
    - lastTransitionTime: "2020-12-15T10:10:45Z"
      message: Started deleting objects
      reason: Started
      status: "True"
      type: ako.vmware.com/ObjectDeletionInProgress
```

After all relevant objects gets deleted from Avi, the reason is changed to `objDeletionDone`.

```
status:
    conditions:
    - lastTransitionTime: "2020-12-15T10:10:48Z"
      message: Successfully deleted all objects
      reason: Done
      status: "False"
      type: ako.vmware.com/ObjectDeletionInProgress
```

To re-create the objects in Avi, the `configmap` has to be edited to set `deleteConfig`: *false*.
After this, the condition in ako statefulset of type `ako.vmware.com/ObjectDeletionInProgress` is deleted automatically.

## How is the shared virtual service lifecycle controlled?

In hostname based sharding, when an ingress object is created with multiple host names, AKO generates an md5 hash using the hostname and the Shard VS number. This uniquely maps an FQDN to a given Shared VS and avoids DNS conflicts. During initial clean bootup, if the Shared virtual service does not exist in Avi - AKO creates the same and then patches the ingress FQDN to it either in the form of a pool (for insecure routes) or in the form of an SNI child virtual service (in case of secure routes).

The Shared virtual services are not deleted if all the FQDNs mapped to it are removed from Kubernetes. However, if the you want to AKO to delete unused shared virtual services , a pod restart is required that would evaluate the virtual service and delete it appropriately.

## How are virtual services sharded? If you create ingress with an insecure host/path combination then AKO creates a corresponding Avi pool object and patches the pool on one of the existing shard virtual services. The shard virtual service has a DataScript associated with it that reads the host/path of the incoming requests and appropriately selects a pool by matching it with the priority label specified for each pool member (corresponding to a host/path combination).

For secure ingresses, an SNI virtual service is created which although is a dedicated virtual service, does not have any IP addresses associated with it. The SNI virtual service is a child to a parent virtual service and is created based on the secret object specified in the ingress file against the host/path that is meant to be accessed securely.

## How do I decide the size of the shard virtual service? In the current AKO model, the Shard VS size is an enum. It allows 3 pre-fixed sets of values: * LARGE (8 virtual services) * MEDIUM (4 virtual services) * SMALL (1 virtual service)

The decision of selecting one of these sizes for Shard virtual service depends on the size of the Kubernetes cluster's ingress requirements. It is recommended to always go with the highest possible Shard virtual service number that is(LARGE) to account for expansion in the future.

## Can I change the Shard VS number? To Shard to virtual services, AKO uses a sharding mechanism that is driven either by the namespace on which the ingress object is created or the hostname of each rule within an ingress object. The latter is marked as default because it ensures that a unique hostname is always sharded consistently to the same virtual service.

Since the sharding logic is determined by the number of Shard virtual services, changing the Shard VS number has the potential hazard of messing up an existing cluster's already synced objects. Hence it's recommended to not change the Shard VS numbers, once they are fixed.

## How do I alter the Shard VS number? Altering the shard VS number is considered as disruptive. This is because dynamic re-adjustment of shard numbers may re-balance the ingress to virtual service mapping. Hence, if you want to alter the shard VS number, first delete the older `configmap` and trigger a complete cleanup of the virtual services in the Controller. Then, edit the configmap and restart of AKO.

## What is the use of static routes? Static routes are created with cluster name as the label. While deploying AKO the admin or the operator decides a Service Engine Group for a given Kubernetes cluster. The same labels are tagged on the routes of this AKO cluster. These routes are pushed to the Service Engine's created on the Service Engine Group. The static routes map each pod CIDR with the Kubernetes node's IP address. However, for static routes to work, the Service Engines must be L2 adjacent to your Kubernetes nodes.

## What happens if I have the same SNI host across multiple namespaces? The ingress API does not prohibit the user from creating the same SNI hostname across multiple namespaces. In the hostname sharding mode, AKO will create 1 SNI virtual service and gather all paths associated with it across namespaces to create corresponding switching rules. However, the user needs to denote each ingress with the TLS secret for a given hostname to qualify the host for the SNI virtual service.

Consider the below example:

```
Ingress 1 (default namespace) --> SNI hostname --> foo.com path: /foo, Secret: foo

Ingress 1 (foo namespace) --> SNI hostname --> foo.com path: /bar, Secret: foo
```

In this case, only one SNI virtual service will be created with `sslkeyandcertificate` as foo

However if the following happens, the behaviour of the SNI virtual service would be indeterministic since the secrets for the same SNI are different:

```
Ingress 1 (default namespace) --> SNI hostname --> foo.com path: /foo, Secret: foo<br>

Ingress 1 (foo namespace) --> SNI hostname --> foo.com path: /bar, Secret: bar<br>
```

This is not supported.

## What out-of-band operations can I do on the objects created by AKO?

AKO runs a refresh cycle that currently just refreshes the cloud object parameters. However, if some out-of-band operations are performed on objects created by AKO via directly interacting with the Avi APIs, AKO may not always be able to mitigate an error caused due to this.

AKO has the best effort, retry layer implementation that would try to detect a problem (For example, an SNI VS deleted from the Avi UI), but it is not guaranteed to work for all such manual operations.

Upon reboot of AKO, a full reconciliation loop is run and most of the out-of-band changes are overwritten with AKO's view of the intended model. This does not happen in every full sync cycle.

## What is the expected behaviour for the same host/path combination across different secure/insecure ingresses? The ingress API allows users to add duplicate hostpaths bound to separate backend services. For example,

```
Ingress1 (default namespace) --> foo.com path: /foo, Service: svc1

Ingress2 (default namespace) --> foo.com path: /foo, Service: svc2
```

Also, ingress allows you to have a mix of secure and insecure hostpath bound to the same backend services like so:

```
Ingress1 (default namespace) --> SNI hostname --> foo.com path: /foo, Secret: secret1

Ingress2 (default namespace) --> foo.com path: /foo, Service: svc2
```

AKO does not explicitly handle these conditions and would continue syncing these objects on the Avi controller, but this may lead to traffic issues. AKO does the best effort of detecting some of these conditions by printing them in logs. A sample log statement looks like this:

```
key: Ingress/default/ingress2, msg: Duplicate entries found for hostpath default/ingress2:
foo.com/foo in ingresses: ["default/ingress1"]
```

## What happens to static routes if the Kubernetes nodes are rebooted/shutdown? AKO programs a static route for every node IP and the pod CIDR associated with it. Even though node state changes to `NotReady` in Kubernetes this configuration is stored in the node object and does not change when the node rebooted/shutdown.

Hence, AKO will not remove the static routes until the Kubernetes node is completely removed from the cluster.

## Can I point my ingress objects to a service of type Loadbalancer? No. The ingress objects should point to the service of type clusterIP.
Loadbalancer services either point to an ingress Controller pod if one is using an in cluster ingress controller or they can directly point to application PODs that need layer 4 load-balancing.

In a configuration where the ingress objects are pointing to services of the type load balancer, AKO's behaviour would be indeterministic.

## What happens when AKO fails to connect to the Avi Controller while booting up? AKO would stop processing the Kubernetes objects and no update would be made to the Avi Controller. After the connection to the Avi Controller is restored, AKO pod has to be rebooted. This can be done by deleting the exiting POD and ako deployment would bring up a new pod, which would start processing Kubernetes objects after verifying connectivity to the Avi Controller.

## What happens if we create ingress objects in an OpenShift environment? AKO does not process ingress objects in an OpenShift environment. If any route corresponding to the ingress object is found, AKO would process that route.

## What are the virtual services for passthrough routes? A set of shared virtual services are created for passthrough routes only in the OpenShift environment to listen on port 443 to handle secure traffic using L4 DataScript. These virtual services have names of the format *'cluster-name'-Shared-Passthrough-'shard-number'*. The Number of shards can be configured using the flag `passthroughShardSize` while installation using helm.

## What happens if *insecureEdgeTerminationPolicy* is set to redirect for a passthrough route? For passthrough routes, the supported values for `insecureEdgeTerminationPolicy` are `None` and `Redirect`. To handle insecure traffic for passthrough routes a set of shared virtual services are created with names of the format `'cluster-name'-Shared-Passthrough-'shard-number'-insecure`. These virtual services listen on port 80. If for any passthrough route, the

`insecureEdgeTerminationPolicy` is found to be `Redirect`, then an HTTP Policy is configured in the insecure passthrough shared virtual service to send appropriate response to an incoming insecure traffic.

## How to debug Invalid input detected errors? AKO reboots and retries some of the invalid input errors. Look out for the following Below cases to in the logs:

- If an invalid cloud name is given in values.yaml or if `ipam_provider_ref` is not set in the vCenter and No Access clouds.
- If the same Service Engine group is used for multiple clusters for vCenter and No Access clouds in Cluster IP mode. This happens as AKO expects unique SE group per cluster if routes are configured by AKO for POD reachability. Look for the `labels does not match with cluster name` message in the logs which points to two clusters using the same Service Engine Group.

## Some of the pool servers in the NodePort mode of AKO are down. How is this fixed? The default behaviour for AKO is to populate all the Node IP as pool server. If master node is not schedulable then, it will be marked down. `nodePortSelector` can be used to specify the labels for the node. In that case, all the node with that label will be picked for the pool server. If the master node is not schedulable then, remove the `nodePortSelector` label for the master node.

## Can we create a secure route with edge/reencrypt termination without key or certificate? For secure routes having termination type edge/reencrypt, key and certificate must be specified in the spec of the route. AKO would not handle routes of these types without key and certificate.

## What happens if a route is created with multiple backends having the same service name? AKO rejects those routes, as each backend should be unique ith it's own weight. Multiple backends having same service would make the weight calculation indeterministic.

Document Revision History

Date	Change Summary
December 18, 2020	Published the Frequently Asked Questions for AKO