



# Setting up Routing Rules using CRDs

Avi Technical Reference (v21.1)

Copyright © 2021

# Setting up Routing Rules using CRDs

[view online](#)

## Overview

Custom Resource Definitions (CRDs) are used to extend the Kubernetes APIs server with additional schemas. To know more, click [here](#).

AMKO uses the [AMKOCluster](#) to configure a federation to member clusters.

AMKO uses the following custom resources to configure the GSLB services in the GSLB leader site:

\* [GSLBConfig](#) \* [GlobalDeploymentPolicy](#) \* [GSLBHostRule](#)

If AMKO is installed via Helm, it by default created one instance of each type in the `avi-system` namespace. To view these objects:

If AMKO is installed via `helm`, it by default creates one instance of each type in the `avi-system` namespace. To see these objects:

```
$ kubectl get amkocluster amkocluster-federation -n avi-system
NAME                               AGE
amkocluster-federation             45m
```

**Note:** Only one instance of each `GDP` and `GSLBConfig` is supported and AMKO will ignore other instances.

## GSLBConfig (gs)

The `GSLBConfig` adds GSLB configuration. The parameters of this object are discussed below:

```
apiVersion: "avi.k8s.io/v1alpha1"
kind: "GSLBConfig"
metadata:
  name: "amko-gc"
  namespace: "avi-system"
spec:
  gslbLeader:
    credentials: gslb-avi-secret
    controllerVersion: 20.1.1
    controllerIP: 10.10.10.10
  memberClusters:
    - clusterContext: cluster1-admin
    - clusterContext: cluster2-admin
  refreshInterval: 1800
  logLevel: "INFO"
  useCustomGlobalFqdn: false
```

Here,

\* `apiVersion`: The api version for this object is `avilb.k8s.io/v1alpha1` \* `kind`: The object kind is `GSLBConfig` \* `name`: Can be anything, but it has to be specified in the GDP object \* `namespace`: By default, this object must be created in `avi-system` \* `gslbLeader.credentials`: A secret object has to be created for (helm install does that automatically) the **GSLB Leader cluster**. The username and password have to be provided as part of this secret object. \* `gslbLeader.controllerVersion`: The version of the **GSLB leader cluster**. \* `gslbLeader.controllerIP`: The **GSLB leader IP address** or the **hostname** along with the **port number**, if any. \* `memberClusters`: The **Kubernetes/OpenShift cluster contexts** which are part of this **GSLB cluster**. \* `refreshInterval`: This is an **internal cache refresh time interval**, on which syncs up with the **AVI objects** and checks if a **sync is required**. \* `logLevel`: Define the **log level** that the **amko pod** prints. The supported values are `INFO`, `DEBUG`, `WARN` and `ERROR` \* `useCustomGlobalFqdn`: If set to `true`, **AMKO** will look for **AKO HostRules** to derive the `GslbService` name using the **local to global fqdn mapping**. If set to `false` (default case), **AMKO** ignores **AKO HostRules** and uses the default way of deriving `GslbService` names by just looking at the **local fqdn** in the `ingress/route/service` type **LB**.

**Notes** \* Only one `GSLBConfig` object is allowed. \* If using `helm install`, a `GSLBConfig` object is created by picking up values from the `values.yaml` file.

## GlobalDeploymentPolicy

The `GlobalDeploymentPolicy` CRD enables selecting **Kubernetes/OpenShift objects** based on certain rules. The selection policy applies to all the clusters which are mentioned in the `GDP` object.

An example for `GlobalDeploymentPolicy` is as shown below:

```
apiVersion: "amko.k8s.io/v1alpha2"
kind: "GlobalDeploymentPolicy"
metadata:
  name: "global-gdp"
  namespace: "avi-system" // a cluster-wide GDP
spec:
  matchRules:
    appSelector:
      label:
        app: gslb
    namespaceSelector:
      label:
        ns: gslb
  matchClusters:
    - cluster: cluster1-admin // cluster names are kubernetes cluster contexts
    - cluster: cluster2-admin
  trafficSplit:
    - cluster: cluster1
      weight: 8
    - cluster: cluster2
      weight: 2
  ttl: 10
  healthMonitorRefs:
    - my-health-monitor1
  sitePersistenceRef: gap-1
```

```
poolAlgorithmSettings:
  lbAlgorithm: GSLB_ALGORITHM_ROUND_ROBIN
```

Here,

- namespace: namespace of this object must be avi-system.
- matchRules: This enables selecting objects using either application labels (configured as labels on Ingress/Route objects) or via namespace labels (configured as labels on the namespace objects). MatchRules are defined as:

```
matchRules:
  appSelector: // application selection criteria
    label:
      app: gslb // kubernetes/openshift label key-value
  namespaceSelector: // namespace selection criteria
    label:
      ns: gslb // kubernetes/openshift label key-value
```

A combination of appSelector and namespaceSelector will decide which objects will be selected for GSLB service consideration.

- appSelector: Selection criteria only for applications:
  - Label: Is used to match the ingress/service type load balancer labels (key:value pair)
- namespaceSelector: Selection criteria only for namespaces:
  - Label: Is used to match the namespace labels (key:value pair).

AMKO supports the following combination for GDP match rules:

| appSelector | namespaceSelector | Result   |
|-------------|-------------------|--|
| Yes         | Yes               | Select all objects satisfying appSelector and from the namespaces satisfying the namespaceSelector |
| No          | Yes               | Select all objects from the selected namespaces (satisfying namespaceSelector)                     |
| Yes         | No                | Select all objects satisfying the appSelector criteria from all namespaces                         |
| No          | No                | No objects selected (default action)   |

For example,

- Select objects with label app:gslb from all the namespaces:

```
matchRules:
  appSelector:
    label:
      app: gslb
```

- Select objects with label app:gslb and from namespaces labelled ns:prod:

```
matchRules:
  appSelector:
```

```

label:
  app: gslb
namespaceSelector:
  label:
    ns: prod

```

- **matchClusters:** List of clusters on which the above matchRules will be applied on. The member object of this list are cluster contexts of the individual Kubernetes/ OpenShift clusters.
- **trafficSplit** is required if we want to route a percentage of traffic to objects in a given cluster. Weights for these clusters range from 1 to 20.
- **ttl:** Use this flag to set the Time-To-Live value. The value can range from 1-86400 seconds. This determines the frequency with which clients need to obtain fresh steering information for client requests. If none is specified in the GDP object, the value defaults to the one specified in the DNS application profile.
- **healthMonitorRefs:** Provide federated custom health monitors. If this option is used and refs are specified, the default path based health monitoring will be deleted for the GslbServices. If no custom health monitors are specified, AMKO sets the default health monitors for all GslbServices.
- **sitePersistenceRef:** Provide an Application Persistence Profile ref (pre-created in Avi Controller). This has to be a federated profile. Follow the steps provided [here](#) to create a federated Application Persistence Profile on the Avi Controller. If no reference is provided, Site Persistence is disabled.
- **poolAlgorithmSettings:** Provide the GslbService pool algorithm settings. Refer to [Pool Algorithm Settings](#) to know more. If this field is absent, the default is assumed as Round Robin algorithm.

## Notes

- Only one GDP object is supported.
- If using helm install, a GDP object is created by picking up values from values.yaml file. This GDP object can be edited to modify their selection of objects.
- trafficSplit, ttl, sitePersistence and healthMonitorRefs provided in the GDP object are applicable on all the GslbServices. These properties, however, can be overridden via GSLBHostRule created for a GslbService.
- Site Persistence, if specified, will only be enabled for the GslbServices which have secure ingresses or secure routes as the members and will be disabled for all other cases.

## GSLBHostRule

GSLBHostRule enables overriding certain properties of a specific GslbService object on the Avi Controller created by AMKO.

An example of a GSLBHostRule is as follows:

```

apiVersion: amko.vmware.com/v1alpha1
kind: GSLBHostRule
metadata:
  name: gslb-host-rule-1

```

```

namespace: avi-system
spec:
  fqdn: foo.avi.internal
  sitePersistence:
    enabled: true
    profileRef: "gap-1" # only enabled for secure ingresses/routes
  thirdPartyMembers:
  - site: non-avi-site
    vip: 10.10.10.10
  healthMonitorRefs:
  - hm1
  - hm2
  trafficSplit:
  - cluster: k8s
    weight: 15
  - cluster: oshift
    weight: 5
  ttl: 30
  poolAlgorithmSettings:
    lbAlgorithm: GSLB_ALGORITHM_ROUND_ROBIN

```

Here,

\* namespace: The namespace of this object must be `avi-system`. \* fqdn: FQDN of the `GslbService` \*

sitePersistence: Enable Site Persistence for client requests. Set the `enabled` flag as `true` and add a `profileRef` for a pre-created Application Persistence Profile created on the Avi Controller. Follow the steps provided [here](#) to create a federated Application Persistence Profile on the Avi Controller.

Note: Site persistence is disabled on `GslbServices` created for insecure Ingresses/Routes, irrespective of this field. If this field is not provided in `GSLBHostRule`, the site persistence property will be inherited from the GDP object. \*

thirdPartyMembers: To add one or more third party members to a GS from a non-avi site (third party site) for the purpose of maintenance, specify a list of those members. For each member, provide the site name in `site` and IP address in `vip`. Refer [Associating Third-Party Services with Third-Party Sites](#) to know more.

Note: To add third party members, set the `enable` flag in `sitePersistence` to `False` for this object. If site persistence is enabled for a GSLB Service, third party members can not be added.

The site must be added to the GSLB leader as a third party site before adding more members here. \* healthMonitorRefs:

If a `GslbService` requires some custom health monitoring, the user can create a federated custom health monitor in the Avi Controller and provide the ref(s) here. To add a custom health monitor, follow the steps given in [Configuring GSLB Health Monitors using Avi UI](#) Configuring GSLB Health Monitors using Avi UI. If no custom health monitor refs have been added, the custom health monitors from the GDP object will be inherited. \* trafficSplit: Specify traffic steering to member clusters

/sites. The traffic is then split proportionately between two different clusters. Weight for each cluster must be provided between 1 to 20. If not added, GDP object's traffic split applies on this `GslbService`. \* ttl: Override the default ttl value specified on the GDP object using this field. \* poolAlgorithmSettings: Provide the `GslbService` pool algorithm settings. Refer to [Pool Algorithm Settings](#) to know more. If this field is absent, the default is assumed as Round Robin algorithm.

## Pool Algorithm Settings The pool algorithm settings for `GslbService(s)` can be specified via the GDP or a `GSLBHostRule` objects. The `GslbService` uses the algorithm settings to distribute the traffic accordingly. To set the required settings, use the following fields:

```

poolAlgorithmSettings:
  lbAlgorithm:

```

```
hashMask:  
geoFallback:  
  lbAlgorithm:  
    hashMask:
```

`lbAlgorithm` is used to specify the name of the algorithm. The supported algorithms are:

- `GSLB_ALGORITHM_CONSISTENT_HASH` (needs the hash mask in the `hashMask` field).
- `GSLB_ALGORITHM_GEO` (needs the fallback algorithm settings to be specified in `geoFallback` fields)
- `GSLB_ALGORITHM_ROUND_ROBIN` (by default)
- `GSLB_ALGORITHM_TOPOLOGY`

If `GSLB_ALGORITHM_GEO` is set as the main algorithm, specify the `geoFallback` settings. `geoFallback.lbAlgorithm` can have either of the two values:

- `GSLB_ALGORITHM_CONSISTENT_HASH` (needs the hash mask in `geoFallback.hashMask`)
- `GSLB_ALGORITHM_ROUND_ROBIN`

For more details on the algorithm that best fits your needs and the configuration on the Avi Controller, refer to [Load Balancing Algorithms for GSLB Pool Members](#).

## Caveats

- Site Persistence cannot be enabled for the `GslbServices` which have insecure ingresses or routes as the members.

| Date          | Change Summary   |
|---------------|--|
| July 29, 2021 | Updated the article with <code>AMKOClerster</code> to configure Federation (version 1.4.2) |