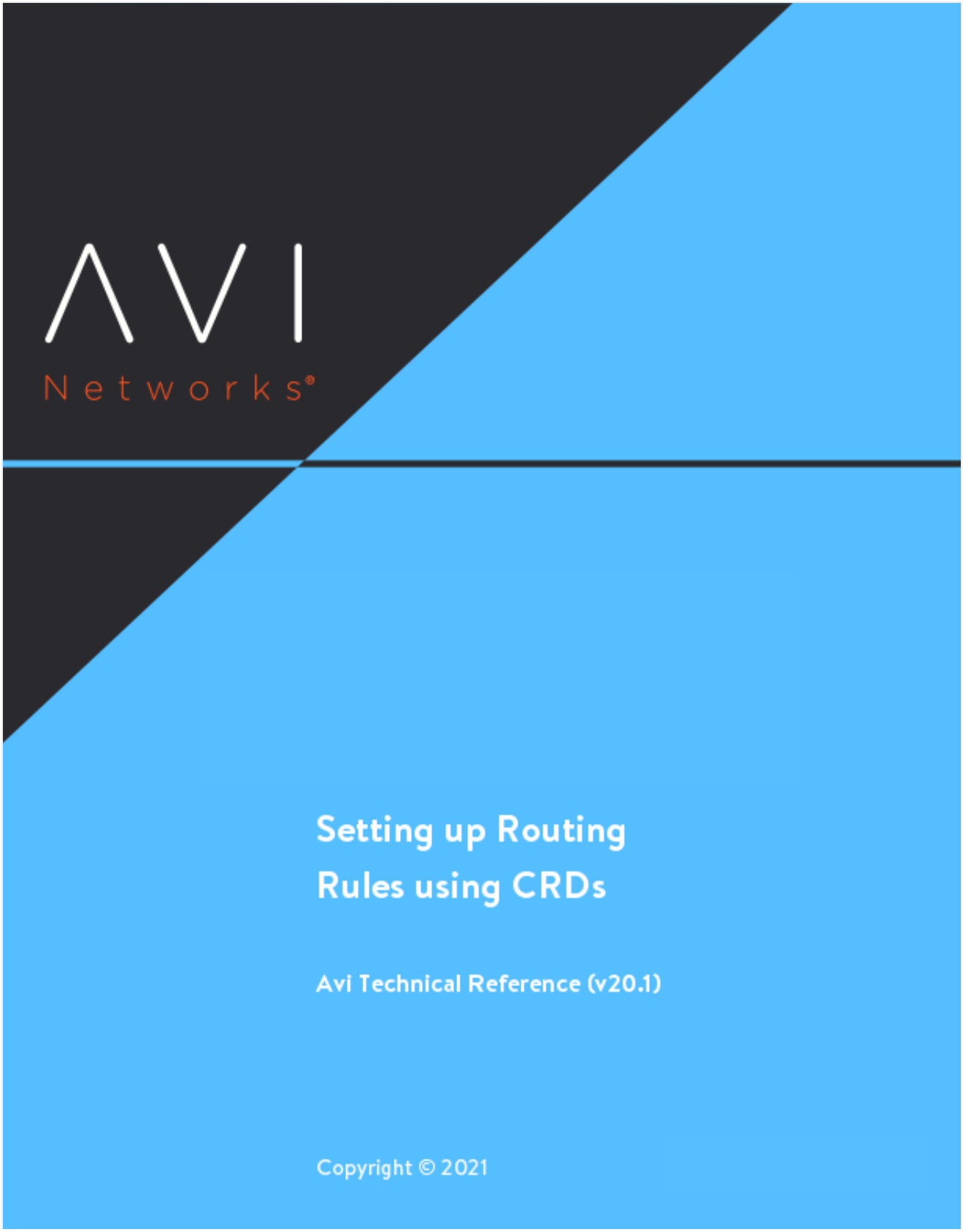

Setting up Routing Rules using CRDs

Avi Technical Reference (v20.1)

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.

The Avi Kubernetes Operator (AKO) supports some CRD objects (installed through helm). The CRDs are relevant to:

Operators:
Users of this category:
* Are aware of the Avi-related semantics * Have access to the Avi controller * Manage the lifecycle of AKO

Developers:
Users of this category:
* Are owners of microservices deployed in Kubernetes * Are assumed to know basic routing principles but may not know the specifics of Avi attributes

Advantages of CRDs

Some load balancers allow configuration options via annotations.

The following are the advantages of using CRDs:

- Versioning: CRDs allow AKO to version fields appropriately due to the dependency on the Avi Controller Versions. In general, this helps preserve unique states across various deployment versions.
- Syntactical Validations: CRDs can be used to verify syntax at the time of creation of the CRD object. This saves API cost and facilitates quicker feedback using a combination of field constraints and effective status messages.
- Role segregation: CRDs can benefit from the Role-based access control (RBAC) policies of Kubernetes and allow stricter access to a group of users.

Types of CRDs Supported in AKO

AKO categorizes the CRDs into:

- Layer 7: These CRD objects are used to express layer 7 traffic routing rules.
- Layer 4: These CRD objects are used to express layer 4 traffic routing rules.
- Infrastructure: These CRD objects are used to control Avi's infrastructure components like Ingress class, SE group properties etc.

Note: As of AKO release 1.1.1, only the layer 7 CRDs are supported.

Layer 7 CRDs

The following are the Layer 7 CRDs currently available in AKO:
* HostRule * HTTPRule

HostRule

The HostRule CRD is used to express additional virtual host properties. The virtual host FQDN is matched from either the Kubernetes ingress or OpenShift route-based objects.

A sample HostRule CRD looks as shown below:

```
apiVersion: ako.vmware.com/v1alpha1
kind: HostRule
metadata:
  name: my-host-rule
  namespace: red
spec:
  virtualhost:
    fqdn: foo.com # mandatory
    enableVirtualhost: true
    tls: # optional
      sslKeyCertificate:
        name: avi-ssl-key-cert
        type: ref
      sslProfile: avi-ssl-profile
      termination: edge
    httpPolicy:
      policySets:
      - avi-secure-policy-ref
      overwrite: false
    datascript:
    - avi-datascript-redirect-app1
    wafPolicy: avi-waf-policy
    applicationProfile: avi-app-ref
    analyticsProfile: avi-analytics-ref
    errorPageProfile: avi-errorpage-ref
```

Usage of HostRule

HostRule CRD can be created in a given namespace where the operator requires better control. This section explains the details and associated rules of using each field of the HostRule CRD.

Enable/Disable Virtual Host

HostRule CRD can be used to enable/disable corresponding virtual services created by AKO on Avi. This removes any virtual host related configuration from the data plane (Avi Service Engines) in addition to disabling traffic on the virtual host/FQDN.

This configuration is enabled by default. To disable virtual host, use

`enableVirtualHost: <em>False</em>`

Note: Enable/Disable virtual hosts can be applied only for secure FQDNs and cannot be applied for insecure routes.

Express *httppolicyset* Object Refs

HostRule CRD can be used to express `httppolicyset` references.

Note: The httppolicyset objects should be pre-created in the Avi Controller.

```
httpPolicy:
       policySets:
       - "avi-secure-policy-ref"
       overwrite: false
```

The `httppolicyset` currently is only applicable for secure FQDNs and cannot be applied for insecure routes. The order of evaluation of the `httpolicyset` rules is in the same order they appear in the CRD definition. The list of `httpolicyset` rules are always interpreted as an AND operation.

AKO currently uses the `httppolicyset` objects on the SNI virtual services to route traffic based on host/path matches. These rules are always at a lower index than the `httppolicyset` objects specified in the CRD object. By default, the value of `overwrite` is set to *False*. To overwrite all `httppolicyset` objects on a SNI virtual service with the ones specified in the HostRule CRD, set the `overwrite` flag to *True*.

Express WAF Policy Object Refs

HostRule CRD can be used to express WAF policy references.

Note: Create the WAF policy object in the Avi Controller prior to the CRD creation as follows:

`wafPolicy: "avi-waf-policy"`

This property can be applied only for secure FQDNs and cannot be applied for insecure routes. WAF policies are useful when deep layer 7 packet filtering is required.

Express Custom Application Profiles

HostRule CRD can be used to express application profile references.

Note: Create the application profile reference in the Avi Controller prior to the CRD creation. The application profile should be of Type `APPLICATION_PROFILE_TYPE_HTTP`.

`applicationProfile: "my-app-ref"`

This property can be applied only for secure FQDNs and cannot be applied for insecure routes. The application profiles can be used for various HTTP/HTTP2 protocol settings.

Express SSL Key and Certificates

For the Avi Kubernetes Operator to control the TLS termination from a privileged namespace, the HostRule CRD can be created in such a namespace.

```
tls:
     sslKeyCertificate:
```



```
        name:"avi-ssl-key-cert"
        type: ref
      termination: edge
```

Here, Name refers to an Avi object of the type *Ref*.

Currently, *edge* is the only type of termination that is supported.

Express Custom Analytics Profiles

HostRule CRD is used to express analytics profile references.
Ensure that the analytics profile reference is created in the Avi Controller prior to this CRD creation.
The analytics profiles can be used for various Network/HTTP/Healthscore analytics settings, log processing etc.
To express analytics profile references, use `analyticsProfile: avi-analytics-ref`.

Note: This property can be applied only for secure FQDNs and cannot be applied for insecure routes.

Express Custom Error Page Profiles

HostRule CRD can be used to express error page profile references.
Ensure that the error page profile reference is created in the Avi Controller prior to this CRD creation.
The error page profiles can be used to send a custom error page to the client generated by the proxy.
To express error page profile references, use `errorPageProfile: avi-errorpage-ref`.

Note: This property can be applied only for secure FQDNs and cannot be applied for insecure routes.

Express DataScripts

HostRule CRD can be used to express error DataScript references.
Ensure that the DataScript references are created in the Avi Controller prior to this CRD creation.
The DataScripts can be used to apply custom scripts to data traffic. The order of evaluation of the DataScripts is in the same order they appear in the CRD definition.

To express DataScript references
`datascript: - avi-datascript-redirect-app1`

Express TLS Configuration

For the Kubernetes operator to control the TLS termination from a privileged namespace, the HostRule CRD can be created in such a namespace.

```
tls:
      sslKeyCertificate:
        name: avi-ssl-key-cert
        type: ref
      sslProfile: avi-ssl-profile
      termination: edge
```

Here,
* name: Refers to an Avi object with the Type as *ref*. * sslProfile: Used to determine the set of SSL versions and ciphers to accept for SSL/TLS terminated connections. If the sslProfile is not defined, AKO defaults to the sslProfile `System-Standard-PFS` defined in Avi.

Note: Currently, only Edge is supported as the type of termination.

Status Messages

The status messages are used to give instant feedback about the reference objects specified in the HostRule CRD.

Following are some of the sample status messages:

Accepted HostRule Object

```
NAME                HOST                STATUS     AGE
secure-waf-policy    foo.avi.internal      Accepted   3d3h
```

A HostRule is accepted only when all the reference objects specified inside it exist in the Avi Controller.

A Rejected HostRule Object

```
NAME                    HOST                STATUS     AGE
secure-waf-policy-alt  foo.avi.internal       Rejected   2d23h
```

The reason for rejection can be obtained from the status:

```
status:
error: duplicate fqdn foo.avi.internal found in default/secure-waf-policy-alt
status: Rejected
```

Caveats

Converting Insecure FQDNs to Secure

The HostRule CRD can be used to convert an insecure host FQDN to a secure one. This is done by specifying a TLS section in the CRD object. The `sslKeyCertificate`is provided for the FQDN, will override all `sslkeyandcertificates` generated for the FQDN. This is useful if:

- The operator wants to convert an insecure ingress FQDN to secure.
- The operator wants to override any existing secrets for a given host FQDN and define TLS termination semantics.

HostRule Deletion

If a HostRule is deleted, all the settings for the FQDNs are withdrawn from the Avi controller.

HostRule Admission

A HostRule CRD is only admitted if all the objects referenced in it, exist in the Avi Controller. If after admission the object references are deleted out-of-band, then AKO does not re-validate the associated HostRule CRD objects. The user needs to manually edit or delete the object for new changes to take effect.

Duplicate FQDN rules

Two HostRule CRDs cannot be used for the same FQDN information across namespaces. If AKO finds a duplicate FQDN in more than one HostRules, AKO honors the first HostRule that gets created and rejects the others. In case of AKO reboots, the CRD that gets honored might not be the same as the one honored earlier.

HTTP Rule

The path matching rules in the ingress or route objects define traffic routing rules to the microservices. The HTTPRule CRD can be used as a complimentary object to control additional layer 7 properties like algorithm, hash, and tls re-encrypt use cases.

A sample HTTPRule object is as shown below:

```
apiVersion: ako.vmware.com/v1alpha1
kind: HTTPRule
metadata:
   name: my-http-rule
   namespace: purple-l7
spec:
  fqdn: foo.avi.internal
  paths:
  - target: /foo
    healthMonitors:
    - my-health-monitor-1
    - my-health-monitor-2
    loadBalancerPolicy:
      algorithm: LB_ALGORITHM_CONSISTENT_HASH
      hash: LB_ALGORITHM_CONSISTENT_HASH_SOURCE_IP_ADDRESS
    tls: ## This is a re-encrypt to pool
      type: reencrypt # Mandatory [re-encrypt]
      sslProfile: avi-ssl-profile
      destinationCA:  |-
        -----BEGIN CERTIFICATE-----
        [...]
        -----END CERTIFICATE-----
```

Note: The HTTPRule only applies to paths in the Ingress/Route objects which are specified in the same namespace as the HTTPRule CRD.

Usage of the HTTPRule CRD

The HTTPRule CRD does not have any Avi specific semantics. Hence you are free to express your preferences using this CRD without any knowledge of the Avi objects. Each HTTPRule CRD must be bound to a FQDN (both secure or insecure) to subscribe to rules for a specific hostpath combinations.

A HTTPRule CRD would be `Rejected` if the corresponding HostRule CRD does not exist.

Express Load Balancer Algorithm

The load balancer policies are a predefined set of values to choose from.
Currently, the following values are supported for load balancer policy:

- `LB_ALGORITHM_CONSISTENT_HASH`
- `LB_ALGORITHM_CORE_AFFINITY`
- `LB_ALGORITHM_FASTEST_RESPONSE`
- `LB_ALGORITHM_FEWEST_SERVERS`
- `LB_ALGORITHM_FEWEST_TASKS`
- `LB_ALGORITHM_LEAST_CONNECTIONS`
- `LB_ALGORITHM_LEAST_LOAD`
- `LB_ALGORITHM_NEAREST_SERVER`
- `LB_ALGORITHM_RANDOM`
- `LB_ALGORITHM_ROUND_ROBIN`
- `LB_ALGORITHM_TOPOLOGY`

To configure the load balancer policy for a given ingress path,

```
target: /foo
     loadBalancerPolicy:
       algorithm: LB_ALGORITHM_CONSISTENT_HASH
```

This rule is applied all paths matching `/foo` and subsets of `/foo/xxx`.

To know more, refer to the Load Balancing Algorithm article.

Express Health Monitors

The HTTPRule CRD can be used to express health monitor references.
Ensure that the health monitor reference is created in the Avi Controller prior to this CRD creation.

To express health monitor references, use:

```
healthMonitors:
  - my-health-monitor-1
  - my-health-monitor-2
```

The health monitors can be used to verify server health. A server (Kubernetes pods in this case) is marked as *UP* only when all the health monitors return successful responses.

Reencrypt Traffic to the Services

While AKO can terminate TLS traffic, it also provides an option where the users can choose to re-encrypt the traffic between the Avi SE and the backend application server. The following option is provided for reencrypt:

```
tls: ## This is a re-encrypt to pool
      type: reencrypt # Mandatory [re-encrypt]
      sslProfile: avi-ssl-profile
```

If the `sslProfile` is not defined, AKO defaults to sslProfile `System-Standard` to exchange the tls parameters like TLS versions, ciphers etc.

The `sslProfile`, additionally, can be used to determine the set of SSL versions and ciphers to accept for SSL/TLS terminated connections. If the `sslProfile` is not defined, AKO defaults to sslProfile `System-Standard` defined in Avi.

As a further enhancement, you can specify a destination CA, that uses a PKI profile to validate the server certificates.

Status Messages

The status messages are used to give instant feedback on whether a HTTPRule CRD was accepted or rejected.

Example of a HTTP Rule

```
$ kubectl get httprule
NAME            HOSTRULE                    STATUS     AGE
my-http-rules   default/secure-waf-policy    Accepted   5h34m
```

Document Revision History

Date	Change Summary
December 18, 2020	Updated the CRDs supported in AKO version 1.3.1
July 22, 2020	Published the article for Custom Resource Definitions