

asm.alauda.io

apiattributes	canarydeliveries	canaryevents
canarytemplates	casemonitors	connections
domains	envoyfilterbindings	envoyfilters
errorchecks	gatewaydeploys	gatewaygroups
gatewayprojectbindings	globalratelimiters	ingressgateways
isolatepods	jwtolicies	loadbalancers
manualgrayevents	manualgrayreleases	microservices
outlierdetections	servicemeshgroups	servicemeshes
sidecarbypasses	sidecarhotupgradeplans	wasmmodules

wasmmoduletemplates

whitelists

Description

ApiAttribute is the Schema for the apiattributes API

Type

object

Specification

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗

Property	Type	Description
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	ApiAttributeSpec defines the desired state of ApiAttribute
<code>status</code>	<code>object</code>	ApiAttributeStatus defines the observed state of ApiAttribute

`.spec`

Description

ApiAttributeSpec defines the desired state of ApiAttribute

Type

`object`

Property	Type	Description
<code>exacts</code>	<code>array</code>	Exact match type.
<code>prefixs</code>	<code>array</code>	Prefix matching.
<code>regmatches</code>	<code>array</code>	Regular matching, deprecated.

`.spec.exacts`

Description

Exact match type.

Type

array

.spec.exacts[]

Type

object

Required

matchpath

method

operationid

Property	Type	Description
envoypatternpath	string	
matchpath	string	
method	string	
operationid	string	
sourcetype	string	

.spec.prefixs

Description

Prefix matching.

Type

array

.spec.prefixs[]

Type

object

Required

matchpath method operationid

Property	Type	Description
envoypatternpath	string	
matchpath	string	
method	string	
operationid	string	
sourcetype	string	

.spec.regmatches

Description

Regular matching, deprecated.

Type

array

.spec.regmatches[]

Type

object

Required

matchpath method operationid

Property	Type	Description
envoypatternpath	string	

Property	Type	Description
<code>matchpath</code>	<code>string</code>	
<code>method</code>	<code>string</code>	
<code>operationid</code>	<code>string</code>	
<code>sourcetype</code>	<code>string</code>	

.status

Description

ApiAttributeStatus defines the observed state of ApiAttribute

Type

`object`

API Endpoints

The following API endpoints are available:

- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/apiattributes`
 - `DELETE` : delete collection of ApiAttribute
 - `GET` : list objects of kind ApiAttribute
 - `POST` : create a new ApiAttribute
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/apiattributes/{name}`
 - `DELETE` : delete the specified ApiAttribute
 - `GET` : read the specified ApiAttribute
 - `PATCH` : partially update the specified ApiAttribute
 - `PUT` : replace the specified ApiAttribute

- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/apiattributes/{name}/status`
 - `GET` : read status of the specified ApiAttribute
 - `PATCH` : partially update status of the specified ApiAttribute
 - `PUT` : replace status of the specified ApiAttribute

`/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/apiattributes`

HTTP method

`DELETE`

Description

delete collection of ApiAttribute

HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code>  schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

list objects of kind ApiAttribute

HTTP responses

HTTP code	Response body
200 - OK	<code>ApiAttributeList</code>  schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new ApiAttribute

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>ApiAttribute</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	ApiAttribute schema
201 - Created	ApiAttribute schema
202 - Accepted	ApiAttribute schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/apiattributes/{name}

HTTP method

[DELETE](#)

Description

delete the specified ApiAttribute

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	Status  schema
202 - Accepted	Status  schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified ApiAttribute

HTTP responses

HTTP code	Response body
200 - OK	<code>ApiAttribute</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified ApiAttribute

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields.

Parameter	Type	Description
		This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>ApiAttribute</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified `ApiAttribute`

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object,

Parameter	Type	Description
		and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	ApiAttribute schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	ApiAttribute schema
201 - Created	ApiAttribute schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/apiattributes/{name}/status

HTTP method

GET

Description

read status of the specified ApiAttribute

HTTP responses

HTTP code	Response body
200 - OK	<code>ApiAttribute</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified `ApiAttribute`

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>ApiAttribute</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified ApiAttribute

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+. - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>ApiAttribute</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ApiAttribute</code> schema
201 - Created	<code>ApiAttribute</code> schema
401 - Unauthorized	Empty

Description

CanaryDelivery is the Schema for the Canarydeliveries API

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗

Property	Type	Description
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	CanaryDeliverySpec defines the desired state of CanaryDelivery
<code>status</code>	<code>object</code>	CanaryDeliveryStatus defines the observed state of CanaryDelivery

.spec

Description

CanaryDeliverySpec defines the desired state of CanaryDelivery

Type

`object`

Required

`targetRef`

Property	Type	Description
<code>analysis</code>	<code>object</code>	Analysis defines the validation process of a release
<code>approvegate</code>	<code>boolean</code>	
<code>autopromote</code>	<code>boolean</code>	

Property	Type	Description
<code>autoscalerRef</code>	<code>object</code>	AutoscalerRef references an autoscaling resource
<code>delivertype</code>	<code>string</code>	
<code>failFallBack</code>	<code>boolean</code>	if set true,we will rollback canary workload modify
<code>isavaliabile</code>	<code>boolean</code>	
<code>maxResponseTime</code>	<code>number</code>	the max response time,if larger than will failed,default is 500ms
<code>metrics</code>	<code>array</code>	Metric check list for this canary analysis
<code>minSuccessRate</code>	<code>number</code>	the min request succcess rate,if below,canary will failed,default is 95 percent
<code>promote</code>	<code>boolean</code>	
<code>rollback</code>	<code>boolean</code>	
<code>targetRef</code>	<code>object</code>	TargetRef references a target resource

.spec.analysis

Description

Analysis defines the validation process of a release

Type

object

Property	Type	Description
interval	string	Schedule interval for this canary analysis
iterations	integer	Number of checks to run for A/B Testing and Blue/Green
match	array	A/B testing HTTP header match conditions
maxWeight	integer	Max traffic percentage routed to canary
mirror	boolean	Enable traffic mirroring for Blue/Green
stepWeight	integer	Incremental traffic percentage step
stepWeightPromotion	integer	Incremental traffic percentage step for promotion phase
threshold	integer	Max number of failed checks before the canary is terminated

Description

A/B testing HTTP header match conditions

Type

array

.spec.analysis.match[]

Description

HttpMatchRequest specifies a set of criterion to be met in order for the rule to be applied to the HTTP request. For example, the following restricts the rule to match only requests where the URL path starts with /ratings/v2/ and the request contains a "cookie" with value "user=jason".

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: ratings-route
spec:
  hosts:
  - ratings
  http:
  - match:
    - headers:
      cookie:
        regex: "^(.*?;)?(user=jason)(;.*)?"
      uri:
        prefix: "/ratings/v2/"
    route:
    - destination:
        host: ratings
```

HTTPMatchRequest CANNOT be empty.

Type

object

Property	Type	Description
authority	object	HTTP Authority values are case-sensitive and formatted as follows: <code>exact: "value"</code> for exact string match <code>prefix: "value"</code> for prefix-based match <code>regex: "value"</code> for ECMAScript style regex-based match

Property	Type	Description
<code>gateways</code>	<code>array</code>	Names of gateways where the rule should be applied to. Gateway names at the top of the VirtualService (if any) are overridden. The gateway match is independent of <code>sourceLabels</code>.
<code>headers</code>	<code>object</code>	The header keys must be lowercase and use hyphen as the separator, e.g. <i>x-request-id</i>. Header values are case-sensitive and formatted as follows: • <code>exact: "value"</code> for exact string match • <code>prefix: "value"</code> for prefix-based match • <code>regex: "value"</code> for ECMAScript style regex-based match Note: The keys <code>uri</code>, <code>scheme</code>, <code>method</code>, and <code>authority</code> will be ignored.
<code>method</code>	<code>object</code>	HTTP Method values are case-sensitive and formatted as follows: • <code>exact: "value"</code> for exact string match • <code>prefix: "value"</code> for prefix-based match • <code>regex: "value"</code> for ECMAScript style regex-based match
<code>port</code>	<code>integer</code>	Specifies the ports on the host that is being addressed. Many services only expose a single port or label ports with the protocols they support, in these cases it is not required to explicitly select the port.

Property	Type	Description
<code>scheme</code>	<code>object</code>	URI Scheme values are case-sensitive and formatted as follows: <code>exact: "value"</code> for exact string match <code>prefix: "value"</code> for prefix-based match <code>regex: "value"</code> for ECMAScript style regex-based match
<code>sourceLabels</code>	<code>object</code>	One or more labels that constrain the applicability of a rule to workloads with the given labels. If the VirtualService has a list of gateways specified at the top, it should include the reserved gateway <code>mesh</code> in order for this field to be applicable.
<code>uri</code>	<code>object</code>	URI to match values are case-sensitive and formatted as follows: <code>exact: "value"</code> for exact string match <code>prefix: "value"</code> for prefix-based match <code>regex: "value"</code> for ECMAScript style regex-based match

`.spec.analysis.match[].authority`

Description

HTTP Authority values are case-sensitive and formatted as follows: - ``exact: "value"``` for exact string match - ``prefix: "value"``` for prefix-based match - ``regex: "value"``` for ECMAScript style regex-based match

Type

`object`

Property	Type	Description
<code>exact</code>	<code>string</code>	exact string match
<code>prefix</code>	<code>string</code>	prefix-based match
<code>regex</code>	<code>string</code>	ECMAScript style regex-based match
<code>suffix</code>	<code>string</code>	suffix-based match.

`.spec.analysis.match[].gateways`

Description

Names of gateways where the rule should be applied to. Gateway names at the top of the VirtualService (if any) are overridden. The gateway match is independent of sourceLabels.

Type

`array`

`.spec.analysis.match[].gateways[]`

Type

`string`

`.spec.analysis.match[].headers`

Description

The header keys must be lowercase and use hyphen as the separator, e.g. `_x-request-id_`. Header values are case-sensitive and formatted as follows: - ``exact: "value"`` for exact string match - ``prefix: "value"`` for prefix-based match - ``regex: "value"`` for ECMAScript style regex-based match **Note:** The keys ``uri``, ``scheme``, ``method``, and ``authority`` will be ignored.

Type

object

`.spec.analysis.match[].method`

Description

HTTP Method values are case-sensitive and formatted as follows: - ``exact: "value"`` for exact string match - ``prefix: "value"`` for prefix-based match - ``regex: "value"`` for ECMAScript style regex-based match

Type

object

Property	Type	Description
<code>exact</code>	<code>string</code>	exact string match
<code>prefix</code>	<code>string</code>	prefix-based match
<code>regex</code>	<code>string</code>	ECMAScript style regex-based match
<code>suffix</code>	<code>string</code>	suffix-based match.

`.spec.analysis.match[].scheme`

Description

URI Scheme values are case-sensitive and formatted as follows: - `exact: "value"` for exact string match - `prefix: "value"` for prefix-based match - `regex: "value"` for ECMAScript style regex-based match

Type

object

Property	Type	Description
<code>exact</code>	<code>string</code>	exact string match
<code>prefix</code>	<code>string</code>	prefix-based match
<code>regex</code>	<code>string</code>	ECMAScript style regex-based match
<code>suffix</code>	<code>string</code>	suffix-based match.

`.spec.analysis.match[].sourceLabels`

Description

One or more labels that constrain the applicability of a rule to workloads with the given labels. If the VirtualService has a list of gateways specified at the top, it should include the reserved gateway `mesh` in order for this field to be applicable.

Type

object

`.spec.analysis.match[].uri`

Description

URI to match values are case-sensitive and formatted as follows: - `exact: "value"` for exact string match - `prefix: "value"` for prefix-based match - `regex: "value"` for ECMAScript style regex-based match

Type

object

Property	Type	Description
exact	string	exact string match
prefix	string	prefix-based match
regex	string	ECMAScript style regex-based match
suffix	string	suffix-based match.

.spec.autoscalerRef

Description

AutoscalerRef references an autoscaling resource

Type

object

Required

name

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	API version of the referent
<code>kind</code>	<code>string</code>	Kind of the referent
<code>name</code>	<code>string</code>	Name of the referent
<code>namespace</code>	<code>string</code>	Namespace of the referent

`.spec.metrics`

Description

Metric check list for this canary analysis

Type

`array`

`.spec.metrics[]`

Description

CanaryMetric holds the reference to metrics used for canary analysis

Type

`object`

Required

`name`

Property	Type	Description
<code>interval</code>	<code>string</code>	Interval represents the windows size
<code>name</code>	<code>string</code>	Name of the metric
<code>query</code>	<code>string</code>	Prometheus query for this metric (deprectated in favor of TemplateRef)
<code>templateRef</code>	<code>object</code>	TemplateRef references a metric template object
<code>threshold</code>	<code>number</code>	Max value accepted for this metric
<code>thresholdRange</code>	<code>object</code>	Range value accepted for this metric

`.spec.metrics[].templateRef`

Description

TemplateRef references a metric template object

Type

`object`

Required

`name`

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	API version of the referent
<code>kind</code>	<code>string</code>	Kind of the referent
<code>name</code>	<code>string</code>	Name of the referent
<code>namespace</code>	<code>string</code>	Namespace of the referent

`.spec.metrics[].thresholdRange`

Description

Range value accepted for this metric

Type

`object`

Property	Type	Description
<code>max</code>	<code>number</code>	Maximum value
<code>min</code>	<code>number</code>	Minimum value

`.spec.targetRef`

Description

TargetRef references a target resource

Type

object

Required

name

Property	Type	Description
apiVersion	string	API version of the referent
kind	string	Kind of the referent
name	string	Name of the referent
namespace	string	Namespace of the referent

.status

Description

CanaryDeliveryStatus defines the observed state of CanaryDelivery

Type

object

Required

canaryWeight failedChecks iterations lastInitErrorCount phase

Property	Type	Description
canaryWeight	integer	

Property	Type	Description
<code>canaryready</code>	<code>boolean</code>	
<code>conditions</code>	<code>array</code>	
<code>failedChecks</code>	<code>integer</code>	
<code>iterations</code>	<code>integer</code>	
<code>lastAppliedSpec</code>	<code>string</code>	
<code>lastInitErrorCount</code>	<code>integer</code>	
<code>lastInitErrorDesc</code>	<code>string</code>	
<code>lastPromotedSpec</code>	<code>string</code>	
<code>phase</code>	<code>string</code>	CanaryPhase is a label for the condition of a canary at the current time
<code>trackedConfigs</code>	<code>object</code>	

.status.conditions

Type

`array`

.status.conditions[]

Description

CanaryCondition is a status condition for a Canary

Type

`object`

Required

status

type

Property	Type	Description
lastTransitionTime	string	LastTransitionTime of this condition
lastUpdateTime	string	LastUpdateTime of this condition
message	string	Message associated with this condition
reason	string	Reason for the current status of this condition
status	string	Status of this condition
type	string	Type of this condition

.status.trackedConfigs

Type

object

API Endpoints

The following API endpoints are available:

- /apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/canarydeliveries

- **DELETE** : delete collection of CanaryDelivery
- **GET** : list objects of kind CanaryDelivery
- **POST** : create a new CanaryDelivery
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/canarydeliveries/{name}`
 - **DELETE** : delete the specified CanaryDelivery
 - **GET** : read the specified CanaryDelivery
 - **PATCH** : partially update the specified CanaryDelivery
 - **PUT** : replace the specified CanaryDelivery
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/canarydeliveries/{name}/status`
 - **GET** : read status of the specified CanaryDelivery
 - **PATCH** : partially update status of the specified CanaryDelivery
 - **PUT** : replace status of the specified CanaryDelivery

`/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/canarydeliveries`

HTTP method

DELETE

Description

delete collection of CanaryDelivery

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind CanaryDelivery

HTTP responses

HTTP code	Response body
200 - OK	<code>CanaryDeliveryList</code> schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new CanaryDelivery

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate

Parameter	Type	Description
		fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	CanaryDelivery schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	CanaryDelivery schema
201 - Created	CanaryDelivery schema
202 - Accepted	CanaryDelivery schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/canarydeliveries/{name}

HTTP method

DELETE

Description

delete the specified CanaryDelivery

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will

Parameter	Type	Description
		result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
202 - Accepted	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

[GET](#)

Description

read the specified CanaryDelivery

HTTP responses

HTTP code	Response body
200 - OK	CanaryDelivery schema
401 - Unauthorized	Empty

HTTP method

[PATCH](#)

Description

partially update the specified CanaryDelivery

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>CanaryDelivery</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified `CanaryDelivery`

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>CanaryDelivery</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>CanaryDelivery</code> schema

HTTP code	Response body
201 - Created	<code>CanaryDelivery</code> schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/canarydeliveries/{name}/status

HTTP method

`GET`

Description

read status of the specified CanaryDelivery

HTTP responses

HTTP code	Response body
200 - OK	<code>CanaryDelivery</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified CanaryDelivery

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>CanaryDelivery</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified `CanaryDelivery`

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun

Parameter	Type	Description
		directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>CanaryDelivery</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>CanaryDelivery</code> schema
201 - Created	<code>CanaryDelivery</code> schema
401 - Unauthorized	Empty

Description

CanaryEvent is the Schema for the CanaryEvent API

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗

Property	Type	Description
metadata	ObjectMeta	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	CanaryDeliverySpec defines the desired state of CanaryDelivery

.spec

Description

CanaryDeliverySpec defines the desired state of CanaryDelivery

Type

object

Property	Type	Description
currentRecordId	integer	current max record id
records	object	CanaryRecord list for canary

.spec.records

Description

CanaryRecord list for canary

Type

object

Required

namespace

startTime

Property	Type	Description
currentWeight	integer	
endTime	string	
canarydeploy	string	
configtracks	array	
primarydeploy	string	
events	array	
id	integer	Id for this record
name	string	
namespace	string	
startTime	string	
status	string	

.spec.records.configtracks

Type

array

.spec.records.configtracks[]

Type

object

Required

configtype

name

Property	Type	Description
canaryyaml	string	
configtype	string	config type configmap or secret
name	string	ConfigTracks
primaryyaml	string	

.spec.records.events

Type

array

.spec.records.events[]

Description

CanaryEvent is used for events of canary

Type

object

Property	Type	Description
eventName	string	name for the event
eventType	string	event type like canary ab bluegreen

Property	Type	Description
message	string	the message flagger report
phase	string	Phase
timestamp	string	the event generate time
weight	integer	current weight

API Endpoints

The following API endpoints are available:

- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/canaryevents`
 - `DELETE` : delete collection of CanaryEvent
 - `GET` : list objects of kind CanaryEvent
 - `POST` : create a new CanaryEvent
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/canaryevents/{name}`
 - `DELETE` : delete the specified CanaryEvent
 - `GET` : read the specified CanaryEvent
 - `PATCH` : partially update the specified CanaryEvent
 - `PUT` : replace the specified CanaryEvent

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/canaryevents

HTTP method

[DELETE](#)

Description

delete collection of CanaryEvent

HTTP responses

HTTP code	Response body
200 - OK	Status schema
401 - Unauthorized	Empty

HTTP method

[GET](#)

Description

list objects of kind CanaryEvent

HTTP responses

HTTP code	Response body
200 - OK	CanaryEventList schema
401 - Unauthorized	Empty

HTTP method

[POST](#)

Description

create a new CanaryEvent

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>CanaryEvent</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>CanaryEvent</code> schema
201 - Created	<code>CanaryEvent</code> schema

HTTP code	Response body
202 - Accepted	CanaryEvent schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/canaryevents/{name}

HTTP method

[DELETE](#)

Description

delete the specified CanaryEvent

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	Status  schema
202 - Accepted	Status  schema
401 - Unauthorized	Empty

HTTP method

[GET](#)

Description

read the specified CanaryEvent

HTTP responses

HTTP code	Response body
200 - OK	<code>CanaryEvent</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified CanaryEvent

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server

Parameter	Type	Description
		will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>CanaryEvent</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified `CanaryEvent`

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the

Parameter	Type	Description
		request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	CanaryEvent schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	CanaryEvent schema
201 - Created	CanaryEvent schema
401 - Unauthorized	Empty

Description

CanaryTemplate is the Schema for the Canarytemplates API

Type

object

Specification

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

Property	Type	Description
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	CanaryTemplateSpec defines the desired state of CanaryTemplate

.spec

Description

CanaryTemplateSpec defines the desired state of CanaryTemplate

Type

`object`

Property	Type	Description
<code>analysis</code>	<code>object</code>	Analysis defines the validation process of a release
<code>autoscalerRef</code>	<code>object</code>	AutoscalerRef references an autoscaling resource
<code>canaryAnalysis</code>	<code>object</code>	Deprecated: replaced by Analysis
<code>failFallBack</code>	<code>boolean</code>	if set true,we will rollback canary workload modify

Property	Type	Description
<code>ingressRef</code>	<code>object</code>	Reference to NGINX ingress resource
<code>maxResponseTime</code>	<code>number</code>	
<code>metricsServer</code>	<code>string</code>	MetricsServer overwrites the -metrics-server flag for this particular canary
<code>minSuccessRate</code>	<code>number</code>	
<code>progressDeadlineSeconds</code>	<code>integer</code>	ProgressDeadlineSeconds represents the maximum time in seconds for a canary deployment to make progress before it is considered to be failed
<code>service</code>	<code>object</code>	Service defines how ClusterIP services, service mesh or ingress routing objects are generated
<code>skipAnalysis</code>	<code>boolean</code>	SkipAnalysis promotes the canary without analysing it
<code>targetRef</code>	<code>object</code>	TargetRef references a target resource

.spec.analysis

Description

Analysis defines the validation process of a release

Type

object

Required

interval

Property	Type	Description
interval	string	Schedule interval for this canary analysis
iterations	integer	Number of checks to run for A/B Testing and Blue/Green
match	array	A/B testing HTTP header match conditions
maxWeight	integer	Max traffic percentage routed to canary
metrics	array	Metric check list for this canary analysis
mirror	boolean	Enable traffic mirroring for Blue/Green
stepWeight	integer	Incremental traffic percentage step
stepWeightPromotion	integer	Incremental traffic percentage step for promotion phase

Property	Type	Description
<code>threshold</code>	<code>integer</code>	Max number of failed checks before the canary is terminated
<code>webhooks</code>	<code>array</code>	Webhook list for this canary analysis

`.spec.analysis.match`

Description

A/B testing HTTP header match conditions

Type

`array`

`.spec.analysis.match[]`

Description

HttpMatchRequest specifies a set of criterion to be met in order for the rule to be applied to the HTTP request. For example, the following restricts the rule to match only requests where the URL path starts with `/ratings/v2/` and the request contains a "cookie" with value "user=jason".
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
 name: ratings-route
spec:
 hosts: - ratings
 http: - match: - headers: cookie: regex: "^(.*?;)?(user=jason)(;.*)?"
uri: prefix: "/ratings/v2/"
route: - destination: host: ratings
HTTPMatchRequest CANNOT be empty.

Type

`object`

Property	Type	Description
<code>authority</code>	<code>object</code>	HTTP Authority values are case-sensitive and formatted as follows:

Property	Type	Description
		<code>exact: "value"</code> for exact string match <code>prefix: "value"</code> for prefix-based match <code>regex: "value"</code> for ECMAScript style regex-based match
<code>gateways</code>	<code>array</code>	Names of gateways where the rule should be applied to. Gateway names at the top of the VirtualService (if any) are overridden. The gateway match is independent of sourceLabels.
<code>headers</code>	<code>object</code>	The header keys must be lowercase and use hyphen as the separator, e.g. <i>x-request-id</i>. Header values are case-sensitive and formatted as follows: <code>exact: "value"</code> for exact string match <code>prefix: "value"</code> for prefix-based match <code>regex: "value"</code> for ECMAScript style regex-based match Note: The keys <code>uri</code>, <code>scheme</code>, <code>method</code>, and <code>authority</code> will be ignored.
<code>method</code>	<code>object</code>	HTTP Method values are case-sensitive and formatted as follows: <code>exact: "value"</code> for exact string match <code>prefix: "value"</code> for prefix-based match <code>regex: "value"</code> for ECMAScript style regex-based match

Property	Type	Description
<code>port</code>	<code>integer</code>	Specifies the ports on the host that is being addressed. Many services only expose a single port or label ports with the protocols they support, in these cases it is not required to explicitly select the port.
<code>scheme</code>	<code>object</code>	URI Scheme values are case-sensitive and formatted as follows: • <code>exact: "value"</code> for exact string match • <code>prefix: "value"</code> for prefix-based match • <code>regex: "value"</code> for ECMAScript style regex-based match
<code>sourceLabels</code>	<code>object</code>	One or more labels that constrain the applicability of a rule to workloads with the given labels. If the VirtualService has a list of gateways specified at the top, it should include the reserved gateway <code>mesh</code> in order for this field to be applicable.
<code>uri</code>	<code>object</code>	URI to match values are case-sensitive and formatted as follows: • <code>exact: "value"</code> for exact string match • <code>prefix: "value"</code> for prefix-based match • <code>regex: "value"</code> for ECMAScript style regex-based match

`.spec.analysis.match[].authority`

Description

HTTP Authority values are case-sensitive and formatted as follows: - ``exact: "value"``` for exact string match - ``prefix: "value"``` for prefix-based match - ``regex: "value"``` for ECMAScript style regex-based match

Type

`object`

Property	Type	Description
<code>exact</code>	<code>string</code>	exact string match
<code>prefix</code>	<code>string</code>	prefix-based match
<code>regex</code>	<code>string</code>	ECMAScript style regex-based match
<code>suffix</code>	<code>string</code>	suffix-based match.

`.spec.analysis.match[].gateways`

Description

Names of gateways where the rule should be applied to. Gateway names at the top of the VirtualService (if any) are overridden. The gateway match is independent of sourceLabels.

Type

`array`

`.spec.analysis.match[].gateways[]`

Type

string

.spec.analysis.match[].headers

Description

The header keys must be lowercase and use hyphen as the separator, e.g. `_x-request-id_`. Header values are case-sensitive and formatted as follows: - ``exact: "value"`` for exact string match - ``prefix: "value"`` for prefix-based match - ``regex: "value"`` for ECMAScript style regex-based match **Note:** The keys ``uri``, ``scheme``, ``method``, and ``authority`` will be ignored.

Type

object

.spec.analysis.match[].method

Description

HTTP Method values are case-sensitive and formatted as follows: - ``exact: "value"`` for exact string match - ``prefix: "value"`` for prefix-based match - ``regex: "value"`` for ECMAScript style regex-based match

Type

object

Property	Type	Description
exact	string	exact string match
prefix	string	prefix-based match
regex	string	ECMAScript style regex-based match

Property	Type	Description
<code>suffix</code>	<code>string</code>	suffix-based match.

`.spec.analysis.match[].scheme`

Description

URI Scheme values are case-sensitive and formatted as follows: - ``exact: "value"``` for exact string match - ``prefix: "value"``` for prefix-based match - ``regex: "value"``` for ECMAScript style regex-based match

Type

`object`

Property	Type	Description
<code>exact</code>	<code>string</code>	exact string match
<code>prefix</code>	<code>string</code>	prefix-based match
<code>regex</code>	<code>string</code>	ECMAScript style regex-based match
<code>suffix</code>	<code>string</code>	suffix-based match.

`.spec.analysis.match[].sourceLabels`

Description

One or more labels that constrain the applicability of a rule to workloads with the given labels. If the VirtualService has a list of gateways specified at the top, it should include the reserved gateway `mesh` in order for this field to be applicable.

Type

object

.spec.analysis.match[].uri

Description

URI to match values are case-sensitive and formatted as follows: - `exact: "value"` for exact string match - `prefix: "value"` for prefix-based match - `regex: "value"` for ECMAScript style regex-based match

Type

object

Property	Type	Description
exact	string	exact string match
prefix	string	prefix-based match
regex	string	ECMAScript style regex-based match
suffix	string	suffix-based match.

.spec.analysis.metrics

Description

Metric check list for this canary analysis

Type

array

.spec.analysis.metrics[]

Description

CanaryMetric holds the reference to metrics used for canary analysis

Type

object

Required

name

Property	Type	Description
interval	string	Interval represents the windows size
name	string	Name of the metric
query	string	Prometheus query for this metric (deprecated in favor of TemplateRef)
templateRef	object	TemplateRef references a metric template object
threshold	number	Max value accepted for this metric
thresholdRange	object	Range value accepted for this metric

`.spec.analysis.metrics[].templateRef`

Description

TemplateRef references a metric template object

Type

object

Required

name

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	API version of the referent
<code>kind</code>	<code>string</code>	Kind of the referent
<code>name</code>	<code>string</code>	Name of the referent
<code>namespace</code>	<code>string</code>	Namespace of the referent

`.spec.analysis.metrics[].thresholdRange`

Description

Range value accepted for this metric

Type

object

Property	Type	Description
<code>max</code>	<code>number</code>	Maximum value
<code>min</code>	<code>number</code>	Minimum value

.spec.analysis.webhooks

Description

Webhook list for this canary analysis

Type

`array`

.spec.analysis.webhooks[]

Description

CanaryWebhook holds the reference to external checks used for canary analysis

Type

`object`

Required

`name` `type` `url`

Property	Type	Description
<code>metadata</code>	<code>ObjectMeta</code>	Metadata (key-value pairs) for this webhook
<code>name</code>	<code>string</code>	Name of this webhook

Property	Type	Description
<code>timeout</code>	<code>string</code>	Request timeout for this webhook
<code>type</code>	<code>string</code>	Type of this webhook
<code>url</code>	<code>string</code>	URL address of this webhook

.spec.autoscalerRef

Description

AutoscalerRef references an autoscaling resource

Type

`object`

Required

`name`

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	API version of the referent
<code>kind</code>	<code>string</code>	Kind of the referent
<code>name</code>	<code>string</code>	Name of the referent

Property	Type	Description
<code>namespace</code>	<code>string</code>	Namespace of the referent

`.spec.canaryAnalysis`

Description

Deprecated: replaced by Analysis

Type

`object`

Required

`interval`

Property	Type	Description
<code>interval</code>	<code>string</code>	Schedule interval for this canary analysis
<code>iterations</code>	<code>integer</code>	Number of checks to run for A/B Testing and Blue/Green
<code>match</code>	<code>array</code>	A/B testing HTTP header match conditions
<code>maxWeight</code>	<code>integer</code>	Max traffic percentage routed to canary
<code>metrics</code>	<code>array</code>	Metric check list for this canary analysis

Property	Type	Description
<code>mirror</code>	<code>boolean</code>	Enable traffic mirroring for Blue/Green
<code>stepWeight</code>	<code>integer</code>	Incremental traffic percentage step
<code>stepWeightPromotion</code>	<code>integer</code>	Incremental traffic percentage step for promotion phase
<code>threshold</code>	<code>integer</code>	Max number of failed checks before the canary is terminated
<code>webhooks</code>	<code>array</code>	Webhook list for this canary analysis

`.spec.canaryAnalysis.match`

Description

A/B testing HTTP header match conditions

Type

`array`

`.spec.canaryAnalysis.match[]`

Description

HttpMatchRequest specifies a set of criterion to be met in order for the rule to be applied to the HTTP request. For example, the following restricts the rule to match only requests where the URL path starts with `/ratings/v2/` and the request contains a "cookie" with value "user=jason".
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
name:

ratings-route spec: hosts: - ratings http: - match: - headers: cookie: regex: "(.*?;)?
(user=jason)(;.*)?" uri: prefix: "/ratings/v2/" route: - destination: host: ratings
HTTPMatchRequest CANNOT be empty.

Type

object

Property	Type	Description
authority	object	HTTP Authority values are case-sensitive and formatted as follows: <code>exact: "value"</code> for exact string match <code>prefix: "value"</code> for prefix-based match <code>regex: "value"</code> for ECMAScript style regex-based match
gateways	array	Names of gateways where the rule should be applied to. Gateway names at the top of the VirtualService (if any) are overridden. The gateway match is independent of sourceLabels.
headers	object	The header keys must be lowercase and use hyphen as the separator, e.g. <i>x-request-id</i>. Header values are case-sensitive and formatted as follows: <code>exact: "value"</code> for exact string match <code>prefix: "value"</code> for prefix-based match <code>regex: "value"</code> for ECMAScript style regex-based match Note: The keys <code>uri</code> , <code>scheme</code> , <code>method</code> , and <code>authority</code> will be ignored.

Property	Type	Description
<code>method</code>	<code>object</code>	HTTP Method values are case-sensitive and formatted as follows: <code>exact: "value"</code> for exact string match <code>prefix: "value"</code> for prefix-based match <code>regex: "value"</code> for ECMAScript style regex-based match
<code>port</code>	<code>integer</code>	Specifies the ports on the host that is being addressed. Many services only expose a single port or label ports with the protocols they support, in these cases it is not required to explicitly select the port.
<code>scheme</code>	<code>object</code>	URI Scheme values are case-sensitive and formatted as follows: <code>exact: "value"</code> for exact string match <code>prefix: "value"</code> for prefix-based match <code>regex: "value"</code> for ECMAScript style regex-based match
<code>sourceLabels</code>	<code>object</code>	One or more labels that constrain the applicability of a rule to workloads with the given labels. If the VirtualService has a list of gateways specified at the top, it should include the reserved gateway <code>mesh</code> in order for this field to be applicable.

Property	Type	Description
uri	object	URI to match values are case-sensitive and formatted as follows: <code>exact: "value"</code> for exact string match <code>prefix: "value"</code> for prefix-based match <code>regex: "value"</code> for ECMAScript style regex-based match

.spec.canaryAnalysis.match[].authority

Description

HTTP Authority values are case-sensitive and formatted as follows: - ``exact: "value"`` for exact string match - ``prefix: "value"`` for prefix-based match - ``regex: "value"`` for ECMAScript style regex-based match

Type

object

Property	Type	Description
exact	string	exact string match
prefix	string	prefix-based match
regex	string	ECMAScript style regex-based match
suffix	string	suffix-based match.

`.spec.canaryAnalysis.match[].gateways`

Description

Names of gateways where the rule should be applied to. Gateway names at the top of the VirtualService (if any) are overridden. The gateway match is independent of sourceLabels.

Type

array

`.spec.canaryAnalysis.match[].gateways[]`

Type

string

`.spec.canaryAnalysis.match[].headers`

Description

The header keys must be lowercase and use hyphen as the separator, e.g. `_x-request-id_`. Header values are case-sensitive and formatted as follows: - ``exact: "value"`` for exact string match - ``prefix: "value"`` for prefix-based match - ``regex: "value"`` for ECMAScript style regex-based match **Note:** The keys ``uri``, ``scheme``, ``method``, and ``authority`` will be ignored.

Type

object

`.spec.canaryAnalysis.match[].method`

Description

HTTP Method values are case-sensitive and formatted as follows: - ``exact: "value"`` for exact string match - ``prefix: "value"`` for prefix-based match - ``regex: "value"`` for ECMAScript style regex-based match

Type

object

Property	Type	Description
<code>exact</code>	<code>string</code>	exact string match
<code>prefix</code>	<code>string</code>	prefix-based match
<code>regex</code>	<code>string</code>	ECMAScript style regex-based match
<code>suffix</code>	<code>string</code>	suffix-based match.

`.spec.canaryAnalysis.match[].scheme`

Description

URI Scheme values are case-sensitive and formatted as follows: - ``exact: "value"``` for exact string match - ``prefix: "value"``` for prefix-based match - ``regex: "value"``` for ECMAScript style regex-based match

Type

`object`

Property	Type	Description
<code>exact</code>	<code>string</code>	exact string match
<code>prefix</code>	<code>string</code>	prefix-based match

Property	Type	Description
<code>regex</code>	<code>string</code>	ECMAScript style regex-based match
<code>suffix</code>	<code>string</code>	suffix-based match.

`.spec.canaryAnalysis.match[].sourceLabels`

Description

One or more labels that constrain the applicability of a rule to workloads with the given labels. If the VirtualService has a list of gateways specified at the top, it should include the reserved gateway `mesh` in order for this field to be applicable.

Type

`object`

`.spec.canaryAnalysis.match[].uri`

Description

URI to match values are case-sensitive and formatted as follows: - `exact: "value"` for exact string match - `prefix: "value"` for prefix-based match - `regex: "value"` for ECMAScript style regex-based match

Type

`object`

Property	Type	Description
<code>exact</code>	<code>string</code>	exact string match

Property	Type	Description
<code>prefix</code>	<code>string</code>	prefix-based match
<code>regex</code>	<code>string</code>	ECMAScript style regex-based match
<code>suffix</code>	<code>string</code>	suffix-based match.

`.spec.canaryAnalysis.metrics`

Description

Metric check list for this canary analysis

Type

`array`

`.spec.canaryAnalysis.metrics[]`

Description

CanaryMetric holds the reference to metrics used for canary analysis

Type

`object`

Required

`name`

Property	Type	Description
<code>interval</code>	<code>string</code>	Interval represents the windows size

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the metric
<code>query</code>	<code>string</code>	Prometheus query for this metric (deprecated in favor of TemplateRef)
<code>templateRef</code>	<code>object</code>	TemplateRef references a metric template object
<code>threshold</code>	<code>number</code>	Max value accepted for this metric
<code>thresholdRange</code>	<code>object</code>	Range value accepted for this metric

.spec.canaryAnalysis.metrics[].templateRef

Description

TemplateRef references a metric template object

Type

`object`

Required

`name`

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	API version of the referent

Property	Type	Description
<code>kind</code>	<code>string</code>	Kind of the referent
<code>name</code>	<code>string</code>	Name of the referent
<code>namespace</code>	<code>string</code>	Namespace of the referent

`.spec.canaryAnalysis.metrics[].thresholdRange`

Description

Range value accepted for this metric

Type

`object`

Property	Type	Description
<code>max</code>	<code>number</code>	Maximum value
<code>min</code>	<code>number</code>	Minimum value

`.spec.canaryAnalysis.webhooks`

Description

Webhook list for this canary analysis

Type

`array`

.spec.canaryAnalysis.webhooks[]

Description

CanaryWebhook holds the reference to external checks used for canary analysis

Type

object

Required

name

type

url

Property	Type	Description
metadata	ObjectMeta ↗	Metadata (key-value pairs) for this webhook
name	string	Name of this webhook
timeout	string	Request timeout for this webhook
type	string	Type of this webhook
url	string	URL address of this webhook

.spec.ingressRef

Description

Reference to NGINX ingress resource

Type

object

Required

`name`

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	API version of the referent
<code>kind</code>	<code>string</code>	Kind of the referent
<code>name</code>	<code>string</code>	Name of the referent
<code>namespace</code>	<code>string</code>	Namespace of the referent

.spec.service

Description

Service defines how ClusterIP services, service mesh or ingress routing objects are generated

Type

`object`

Required

`port``portDiscovery`

Property	Type	Description
<code>backends</code>	<code>array</code>	Backends of the generated App Mesh virtual nodes

Property	Type	Description
<code>corsPolicy</code>	<code>object</code>	Cross-Origin Resource Sharing policy for the generated Istio virtual service
<code>gateways</code>	<code>array</code>	Gateways attached to the generated Istio virtual service Defaults to the internal mesh gateway
<code>headers</code>	<code>object</code>	Headers operations for the generated Istio virtual service
<code>hosts</code>	<code>array</code>	Hosts attached to the generated Istio virtual service Defaults to the service name
<code>match</code>	<code>array</code>	URI match conditions for the generated service
<code>meshName</code>	<code>string</code>	Mesh name of the generated App Mesh virtual nodes and virtual service
<code>name</code>	<code>string</code>	Name of the Kubernetes service generated by Flagger Defaults to <code>CanarySpec.TargetRef.Name</code>
<code>port</code>	<code>integer</code>	Port of the generated Kubernetes service

Property	Type	Description
<code>portDiscovery</code>	<code>boolean</code>	PortDiscovery adds all container ports to the generated Kubernetes service
<code>portName</code>	<code>string</code>	Port name of the generated Kubernetes service Defaults to http
<code>retries</code>	<code>object</code>	Retries policy for the generated virtual service
<code>rewrite</code>	<code>object</code>	Rewrite HTTP URIs for the generated service
<code>targetPort</code>		Target port number or name of the generated Kubernetes service Defaults to CanaryService.Port
<code>timeout</code>	<code>string</code>	Timeout of the HTTP or gRPC request
<code>trafficPolicy</code>	<code>object</code>	TrafficPolicy attached to the generated Istio destination rules

.spec.service.backends

Description

Backends of the generated App Mesh virtual nodes

Type

`array`

`.spec.service.backends[]`

Type

`string`

`.spec.service.corsPolicy`

Description

Cross-Origin Resource Sharing policy for the generated Istio virtual service

Type

`object`

Property	Type	Description
<code>allowCredentials</code>	<code>boolean</code>	Indicates whether the caller is allowed to send the actual request (not the preflight) using credentials. Translates to Access-Control-Allow-Credentials header.
<code>allowHeaders</code>	<code>array</code>	List of HTTP headers that can be used when requesting the resource. Serialized to Access-Control-Allow-Methods header.
<code>allowMethods</code>	<code>array</code>	List of HTTP methods allowed to access the resource. The content will be serialized into the Access-Control-Allow-Methods header.

Property	Type	Description
<code>allowOrigin</code>	<code>array</code>	The list of origins that are allowed to perform CORS requests. The content will be serialized into the Access-Control-Allow-Origin header. Wildcard * will allow all origins.
<code>exposeHeaders</code>	<code>array</code>	A white list of HTTP headers that the browsers are allowed to access. Serialized into Access-Control-Expose-Headers header.
<code>maxAge</code>	<code>string</code>	Specifies how long the the results of a preflight request can be cached. Translates to the Access-Control-Max-Age header.

`.spec.service.corsPolicy.allowHeaders`

Description

List of HTTP headers that can be used when requesting the resource. Serialized to Access-Control-Allow-Methods header.

Type

`array`

`.spec.service.corsPolicy.allowHeaders[]`

Type

`string`

`.spec.service.corsPolicy.allowMethods`

Description

List of HTTP methods allowed to access the resource. The content will be serialized into the Access-Control-Allow-Methods header.

Type

array

.spec.service.corsPolicy.allowMethods[]

Type

string

.spec.service.corsPolicy.allowOrigin

Description

The list of origins that are allowed to perform CORS requests. The content will be serialized into the Access-Control-Allow-Origin header. Wildcard * will allow all origins.

Type

array

.spec.service.corsPolicy.allowOrigin[]

Type

string

.spec.service.corsPolicy.exposeHeaders

Description

A white list of HTTP headers that the browsers are allowed to access. Serialized into Access-Control-Expose-Headers header.

Type

array

.spec.service.corsPolicy.exposeHeaders[]

Type

string

.spec.service.gateways

Description

Gateways attached to the generated Istio virtual service Defaults to the internal mesh gateway

Type

array

.spec.service.gateways[]

Type

string

.spec.service.headers

Description

Headers operations for the generated Istio virtual service

Type

object

Property	Type	Description
request	object	Header manipulation rules to apply before forwarding a request to the destination service

Property	Type	Description
<code>response</code>	<code>object</code>	Header manipulation rules to apply before returning a response to the caller

`.spec.service.headers.request`

Description

Header manipulation rules to apply before forwarding a request to the destination service

Type

`object`

Property	Type	Description
<code>add</code>	<code>object</code>	Append the given values to the headers specified by keys (will create a comma-separated list of values)
<code>remove</code>	<code>array</code>	Remove the specified headers
<code>set</code>	<code>object</code>	Overwrite the headers specified by key with the given values

`.spec.service.headers.request.add`

Description

Append the given values to the headers specified by keys (will create a comma-separated list of values)

Type

`object`

.spec.service.headers.request.remove

Description

Remove the specified headers

Type

`array`

.spec.service.headers.request.remove[]

Type

`string`

.spec.service.headers.request.set

Description

Overwrite the headers specified by key with the given values

Type

`object`

.spec.service.headers.response

Description

Header manipulation rules to apply before returning a response to the caller

Type

`object`

Property	Type	Description
<code>add</code>	<code>object</code>	Append the given values to the headers specified by keys (will create a comma-separated list of values)
<code>remove</code>	<code>array</code>	Remove the specified headers
<code>set</code>	<code>object</code>	Overwrite the headers specified by key with the given values

.spec.service.headers.response.add

Description

Append the given values to the headers specified by keys (will create a comma-separated list of values)

Type

`object`

.spec.service.headers.response.remove

Description

Remove the specified headers

Type

`array`

.spec.service.headers.response.remove[]

Type

`string`

.spec.service.headers.response.set

Description

Overwrite the headers specified by key with the given values

Type

object

.spec.service.hosts

Description

Hosts attached to the generated Istio virtual service Defaults to the service name

Type

array

.spec.service.hosts[]

Type

string

.spec.service.match

Description

URI match conditions for the generated service

Type

array

.spec.service.match[]

Description

HttpMatchRequest specifies a set of criterion to be met in order for the rule to be applied to the HTTP request. For example, the following restricts the rule to match only requests where the URL path starts with `/ratings/v2/` and the request contains a "cookie" with value

"user=jason". apiVersion: networking.istio.io/v1alpha3 kind: VirtualService metadata: name: ratings-route spec: hosts: - ratings http: - match: - headers: cookie: regex: "^(*?;)?(user=jason)(;.*)?" uri: prefix: "/ratings/v2/" route: - destination: host: ratings HTTPMatchRequest CANNOT be empty.

Type

object

Property	Type	Description
authority	object	HTTP Authority values are case-sensitive and formatted as follows: <code>exact: "value"</code> for exact string match <code>prefix: "value"</code> for prefix-based match <code>regex: "value"</code> for ECMAScript style regex-based match
gateways	array	Names of gateways where the rule should be applied to. Gateway names at the top of the VirtualService (if any) are overridden. The gateway match is independent of sourceLabels.
headers	object	The header keys must be lowercase and use hyphen as the separator, e.g. <i>x-request-id</i>. Header values are case-sensitive and formatted as follows: <code>exact: "value"</code> for exact string match <code>prefix: "value"</code> for prefix-based match <code>regex: "value"</code> for ECMAScript style regex-based match Note: The keys <code>uri</code> , <code>scheme</code> , <code>method</code> , and <code>authority</code> will be ignored.

Property	Type	Description
<code>method</code>	<code>object</code>	HTTP Method values are case-sensitive and formatted as follows: <code>exact: "value"</code> for exact string match <code>prefix: "value"</code> for prefix-based match <code>regex: "value"</code> for ECMAScript style regex-based match
<code>port</code>	<code>integer</code>	Specifies the ports on the host that is being addressed. Many services only expose a single port or label ports with the protocols they support, in these cases it is not required to explicitly select the port.
<code>scheme</code>	<code>object</code>	URI Scheme values are case-sensitive and formatted as follows: <code>exact: "value"</code> for exact string match <code>prefix: "value"</code> for prefix-based match <code>regex: "value"</code> for ECMAScript style regex-based match
<code>sourceLabels</code>	<code>object</code>	One or more labels that constrain the applicability of a rule to workloads with the given labels. If the VirtualService has a list of gateways specified at the top, it should include the reserved gateway <code>mesh</code> in order for this field to be applicable.

Property	Type	Description
uri	object	URI to match values are case-sensitive and formatted as follows: <code>exact: "value"</code> for exact string match <code>prefix: "value"</code> for prefix-based match <code>regex: "value"</code> for ECMAScript style regex-based match

.spec.service.match[].authority

Description

HTTP Authority values are case-sensitive and formatted as follows: - ``exact: "value"``` for exact string match - ``prefix: "value"``` for prefix-based match - ``regex: "value"``` for ECMAScript style regex-based match

Type

object

Property	Type	Description
exact	string	exact string match
prefix	string	prefix-based match
regex	string	ECMAScript style regex-based match
suffix	string	suffix-based match.

`.spec.service.match[].gateways`

Description

Names of gateways where the rule should be applied to. Gateway names at the top of the VirtualService (if any) are overridden. The gateway match is independent of sourceLabels.

Type

array

`.spec.service.match[].gateways[]`

Type

string

`.spec.service.match[].headers`

Description

The header keys must be lowercase and use hyphen as the separator, e.g. `_x-request-id_`. Header values are case-sensitive and formatted as follows: - ``exact: "value"`` for exact string match - ``prefix: "value"`` for prefix-based match - ``regex: "value"`` for ECMAScript style regex-based match **Note:** The keys ``uri``, ``scheme``, ``method``, and ``authority`` will be ignored.

Type

object

`.spec.service.match[].method`

Description

HTTP Method values are case-sensitive and formatted as follows: - ``exact: "value"`` for exact string match - ``prefix: "value"`` for prefix-based match - ``regex: "value"`` for ECMAScript style regex-based match

Type

object

Property	Type	Description
<code>exact</code>	<code>string</code>	exact string match
<code>prefix</code>	<code>string</code>	prefix-based match
<code>regex</code>	<code>string</code>	ECMAScript style regex-based match
<code>suffix</code>	<code>string</code>	suffix-based match.

`.spec.service.match[].scheme`

Description

URI Scheme values are case-sensitive and formatted as follows: - ``exact: "value"``` for exact string match - ``prefix: "value"``` for prefix-based match - ``regex: "value"``` for ECMAScript style regex-based match

Type

`object`

Property	Type	Description
<code>exact</code>	<code>string</code>	exact string match
<code>prefix</code>	<code>string</code>	prefix-based match

Property	Type	Description
<code>regex</code>	<code>string</code>	ECMAScript style regex-based match
<code>suffix</code>	<code>string</code>	suffix-based match.

`.spec.service.match[].sourceLabels`

Description

One or more labels that constrain the applicability of a rule to workloads with the given labels. If the VirtualService has a list of gateways specified at the top, it should include the reserved gateway `mesh` in order for this field to be applicable.

Type

`object`

`.spec.service.match[].uri`

Description

URI to match values are case-sensitive and formatted as follows: - `exact: "value"` for exact string match - `prefix: "value"` for prefix-based match - `regex: "value"` for ECMAScript style regex-based match

Type

`object`

Property	Type	Description
<code>exact</code>	<code>string</code>	exact string match

Property	Type	Description
<code>prefix</code>	<code>string</code>	prefix-based match
<code>regex</code>	<code>string</code>	ECMAScript style regex-based match
<code>suffix</code>	<code>string</code>	suffix-based match.

`.spec.service.retries`

Description

Retries policy for the generated virtual service

Type

`object`

Property	Type	Description
<code>attempts</code>	<code>integer</code>	REQUIRED. Number of retries for a given request. The interval between retries can be determined automatically (25ms+). Actual number of retries is limited by the <code>httpReqTimeout</code> .
<code>perTryTimeout</code>	<code>string</code>	Timeout per retry attempt for a given request. format: 1h/1m/1s. Minimum value is <code>>=1ms</code> .
<code>retryOn</code>	<code>string</code>	Specifies the conditions under which retry takes place. One or more conditions can be specified using a ',' delimited list. The supported policies can be found at https://www.envoyproxy.io/docs/envoy/latest/configuration/http/http_filters_retry_policy_filters and envoy-retry-on ↗ and envoy-retry-on ↗ .

Property	Type	Description
		https://www.envoyproxy.io/docs/envoy/latest/configuration/http/http_filters_retry_grpc_filter.html

.spec.service.rewrite

Description

Rewrite HTTP URIs for the generated service

Type

object

Property	Type	Description
<code>authority</code>	<code>string</code>	rewrite the Authority/Host header with this value.
<code>uri</code>	<code>string</code>	rewrite the path (or the prefix) portion of the URI with this value. If the original URI was matched based on prefix, the value provided in this field will replace the corresponding matched prefix.

.spec.service.trafficPolicy

Description

TrafficPolicy attached to the generated Istio destination rules

Type

object

Property	Type	Description
connectionPool	object	Settings controlling the volume of connections to an upstream service
loadBalancer	object	Settings controlling the load balancer algorithms.
outlierDetection	object	Settings controlling eviction of unhealthy hosts from the load balancing pool
portLevelSettings	array	Traffic policies specific to individual ports. Note that port level settings will override the destination-level settings. Traffic settings specified at the destination-level will not be inherited when overridden by port-level settings, i.e. default values will be applied to fields omitted in port-level traffic policies.
tls	object	TLS related settings for connections to the upstream service.

.spec.service.trafficPolicy.connectionPool

Description

Settings controlling the volume of connections to an upstream service

Type

object

Property	Type	Description
<code>http</code>	<code>object</code>	HTTP connection pool settings.
<code>tcp</code>	<code>object</code>	Settings common to both HTTP and TCP upstream connections.

`.spec.service.trafficPolicy.connectionPool.http`

Description

HTTP connection pool settings.

Type

`object`

Property	Type	Description
<code>h2UpgradePolicy</code>	<code>string</code>	Specify if http1.1 connection should be upgraded to http2 for the associated destination. DEFAULT - Use the global default. DO_NOT_UPGRADE - Do not upgrade the connection to http2. UPGRADE - Upgrade the connection to http2.
<code>http1MaxPendingRequests</code>	<code>integer</code>	Maximum number of pending HTTP requests to a destination. Default 2^32-1.

Property	Type	Description
<code>http2MaxRequests</code>	<code>integer</code>	Maximum number of requests to a backend. Default 2^32-1.
<code>idleTimeout</code>	<code>string</code>	The idle timeout for upstream connection pool connections. The idle timeout is defined as the period in which there are no active requests. If not set, the default is 1 hour. When the idle timeout is reached the connection will be closed. Note that request based timeouts mean that HTTP/2 PINGs will not keep the connection alive. Applies to both HTTP1.1 and HTTP2 connections.
<code>maxRequestsPerConnection</code>	<code>integer</code>	Maximum number of requests per connection to a backend. Setting this parameter to 1 disables keep alive. Default 0, meaning "unlimited", up to 2^29.
<code>maxRetries</code>	<code>integer</code>	Maximum number of retries that can be outstanding to all hosts in a cluster at a given time. Defaults to 2^32-1.

`.spec.service.trafficPolicy.connectionPool.tcp`

Description

Settings common to both HTTP and TCP upstream connections.

Type

`object`

Property	Type	Description
<code>connectTimeout</code>	<code>string</code>	TCP connection timeout.
<code>maxConnections</code>	<code>integer</code>	Maximum number of HTTP1 /TCP connections to a destination host.

`.spec.service.trafficPolicy.loadBalancer`

Description

Settings controlling the load balancer algorithms.

Type

`object`

Property	Type	Description
<code>consistentHash</code>	<code>object</code>	Consistent Hash-based load balancing can be used to provide soft session affinity based on HTTP headers, cookies or other properties. This load balancing policy is applicable only for HTTP connections. The affinity to a particular destination host will be lost when one or more hosts are added/removed from the destination service.
<code>simple</code>	<code>string</code>	It is required to specify exactly one of the fields: Simple or ConsistentHash

`.spec.service.trafficPolicy.loadBalancer.consistentHash`

Description

Consistent Hash-based load balancing can be used to provide soft session affinity based on HTTP headers, cookies or other properties. This load balancing policy is applicable only for HTTP connections. The affinity to a particular destination host will be lost when one or more hosts are added/removed from the destination service.

Type

object

Property	Type	Description
<code>httpCookie</code>	<code>object</code>	Hash based on HTTP cookie.
<code>httpHeaderName</code>	<code>string</code>	It is required to specify exactly one of the fields as hash key: HTTPHeaderName, HTTPCookie, or UseSourceIP. Hash based on a specific HTTP header.
<code>minimumRingSize</code>	<code>integer</code>	The minimum number of virtual nodes to use for the hash ring. Defaults to 1024. Larger ring sizes result in more granular load distributions. If the number of hosts in the load balancing pool is larger than the ring size, each host will be assigned a single virtual node.
<code>useSourceIp</code>	<code>boolean</code>	Hash based on the source IP address.

`.spec.service.trafficPolicy.loadBalancer.consistentHash.httpCookie`

Description

Hash based on HTTP cookie.

Type

object

Required

name

ttl

Property	Type	Description
name	string	REQUIRED. Name of the cookie.
path	string	Path to set for the cookie.
ttl	string	REQUIRED. Lifetime of the cookie.

.spec.service.trafficPolicy.outlierDetection**Description**

Settings controlling eviction of unhealthy hosts from the load balancing pool

Type

object

Property	Type	Description
baseEjectionTime	string	Minimum ejection duration. A host will remain ejected for a period equal to the product of minimum ejection duration and the number of times the host has been ejected. This technique allows the system to automatically increase the ejection period for unhealthy upstream servers. format:

Property	Type	Description
		1h/1m/1s/1ms. MUST BE >=1ms. Default is 30s.
<code>consecutive5xxErrors</code>	<code>integer</code>	Number of 5xx errors before a host is ejected from the connection pool. When the upstream host is accessed over an opaque TCP connection, connect timeouts, connection error/failure and request failure events qualify as a 5xx error. This feature defaults to 5 but can be disabled by setting the value to 0. Note that <code>consecutive_gateway_errors</code> and <code>consecutive_5xx_errors</code> can be used separately or together. Because the errors counted by <code>consecutive_gateway_errors</code> are also included in <code>consecutive_5xx_errors</code>, if the value of <code>consecutive_gateway_errors</code> is greater than or equal to the value of <code>consecutive_5xx_errors</code>, <code>consecutive_gateway_errors</code> will have no effect.
<code>consecutiveErrors</code>	<code>integer</code>	Number of errors before a host is ejected from the connection pool. Defaults to 5. When the upstream host is accessed over HTTP, a 5xx return code qualifies as an error. When the upstream host is accessed over an opaque TCP connection, connect timeouts and connection error/failure events qualify as an error.

Property	Type	Description
<code>consecutiveGatewayErrors</code>	<code>integer</code>	Number of gateway errors before a host is ejected from the connection pool. When the upstream host is accessed over HTTP, a 502, 503, or 504 return code qualifies as a gateway error. When the upstream host is accessed over an opaque TCP connection, connect timeouts and connection error/failure events qualify as a gateway error. This feature is disabled by default or when set to the value 0. Note that <code>consecutive_gateway_errors</code> and <code>consecutive_5xx_errors</code> can be used separately or together. Because the errors counted by <code>consecutive_gateway_errors</code> are also included in <code>consecutive_5xx_errors</code> , if the value of <code>consecutive_gateway_errors</code> is greater than or equal to the value of <code>consecutive_5xx_errors</code> , <code>consecutive_gateway_errors</code> will have no effect.
<code>interval</code>	<code>string</code>	Time interval between ejection sweep analysis. format: 1h/1m/1s/1ms. MUST BE >=1ms. Default is 10s.
<code>maxEjectionPercent</code>	<code>integer</code>	Maximum % of hosts in the load balancing pool for the upstream service that can be ejected. Defaults to 10%.

Property	Type	Description
<code>minHealthPercent</code>	<code>integer</code>	Outlier detection will be enabled as long as the associated load balancing pool has at least <code>min_health_percent</code> hosts in healthy mode. When the percentage of healthy hosts in the load balancing pool drops below this threshold, outlier detection will be disabled and the proxy will load balance across all hosts in the pool (healthy and unhealthy). The threshold can be disabled by setting it to 0%. The default is 0% as it's not typically applicable in k8s environments with few pods per service.

`.spec.service.trafficPolicy.portLevelSettings`

Description

Traffic policies specific to individual ports. Note that port level settings will override the destination-level settings. Traffic settings specified at the destination-level will not be inherited when overridden by port-level settings, i.e. default values will be applied to fields omitted in port-level traffic policies.

Type

`array`

`.spec.service.trafficPolicy.portLevelSettings[]`

Description

Traffic policies that apply to specific ports of the service

Type

`object`

Required

port

Property	Type	Description
connectionPool	object	Settings controlling the volume of connections to an upstream service
loadBalancer	object	Settings controlling the load balancer algorithms.
outlierDetection	object	Settings controlling eviction of unhealthy hosts from the load balancing pool
port	object	Specifies the port name or number of a port on the destination service on which this policy is being applied. Names must comply with DNS label syntax (rfc1035) and therefore cannot collide with numbers. If there are multiple ports on a service with the same protocol the names should be of the form -.
tls	object	TLS related settings for connections to the upstream service.

.spec.service.trafficPolicy.portLevelSettings[].connectionPool

Description

Settings controlling the volume of connections to an upstream service

Type

`object`

Property	Type	Description
<code>http</code>	<code>object</code>	HTTP connection pool settings.
<code>tcp</code>	<code>object</code>	Settings common to both HTTP and TCP upstream connections.

`.spec.service.trafficPolicy.portLevelSettings[].connectionPool.http`

Description

HTTP connection pool settings.

Type

`object`

Property	Type	Description
<code>h2UpgradePolicy</code>	<code>string</code>	Specify if http1.1 connection should be upgraded to http2 for the associated destination. DEFAULT - Use the global default. DO_NOT_UPGRADE - Do not upgrade the connection to http2. UPGRADE - Upgrade the connection to http2.
<code>http1MaxPendingRequests</code>	<code>integer</code>	Maximum number of pending HTTP requests to a destination. Default 2^32-1.

Property	Type	Description
<code>http2MaxRequests</code>	<code>integer</code>	Maximum number of requests to a backend. Default 2^32-1.
<code>idleTimeout</code>	<code>string</code>	The idle timeout for upstream connection pool connections. The idle timeout is defined as the period in which there are no active requests. If not set, the default is 1 hour. When the idle timeout is reached the connection will be closed. Note that request based timeouts mean that HTTP/2 PINGs will not keep the connection alive. Applies to both HTTP1.1 and HTTP2 connections.
<code>maxRequestsPerConnection</code>	<code>integer</code>	Maximum number of requests per connection to a backend. Setting this parameter to 1 disables keep alive. Default 0, meaning "unlimited", up to 2^29.
<code>maxRetries</code>	<code>integer</code>	Maximum number of retries that can be outstanding to all hosts in a cluster at a given time. Defaults to 2^32-1.

`.spec.service.trafficPolicy.portLevelSettings[].connectionPool.tcp`

Description

Settings common to both HTTP and TCP upstream connections.

Type

`object`

Property	Type	Description
<code>connectTimeout</code>	<code>string</code>	TCP connection timeout.
<code>maxConnections</code>	<code>integer</code>	Maximum number of HTTP1 /TCP connections to a destination host.

`.spec.service.trafficPolicy.portLevelSettings[].loadBalancer`

Description

Settings controlling the load balancer algorithms.

Type

`object`

Property	Type	Description
<code>consistentHash</code>	<code>object</code>	Consistent Hash-based load balancing can be used to provide soft session affinity based on HTTP headers, cookies or other properties. This load balancing policy is applicable only for HTTP connections. The affinity to a particular destination host will be lost when one or more hosts are added/removed from the destination service.
<code>simple</code>	<code>string</code>	It is required to specify exactly one of the fields: Simple or ConsistentHash

`.spec.service.trafficPolicy.portLevelSettings[].loadBalancer.consistentHash`

Description

Consistent Hash-based load balancing can be used to provide soft session affinity based on HTTP headers, cookies or other properties. This load balancing policy is applicable only for HTTP connections. The affinity to a particular destination host will be lost when one or more hosts are added/removed from the destination service.

Type

`object`

Property	Type	Description
<code>httpCookie</code>	<code>object</code>	Hash based on HTTP cookie.
<code>httpHeaderName</code>	<code>string</code>	It is required to specify exactly one of the fields as hash key: HTTPHeaderName, HTTPCookie, or UseSourceIP. Hash based on a specific HTTP header.
<code>minimumRingSize</code>	<code>integer</code>	The minimum number of virtual nodes to use for the hash ring. Defaults to 1024. Larger ring sizes result in more granular load distributions. If the number of hosts in the load balancing pool is larger than the ring size, each host will be assigned a single virtual node.
<code>useSourceIp</code>	<code>boolean</code>	Hash based on the source IP address.

`.spec.service.trafficPolicy.portLevelSettings[].loadBalancer.consistentHash.httpCookie`

Description

Hash based on HTTP cookie.

Type

object

Required

name ttl

Property	Type	Description
name	string	REQUIRED. Name of the cookie.
path	string	Path to set for the cookie.
ttl	string	REQUIRED. Lifetime of the cookie.

.spec.service.trafficPolicy.portLevelSettings[].outlierDetection

Description

Settings controlling eviction of unhealthy hosts from the load balancing pool

Type

object

Property	Type	Description
baseEjectionTime	string	Minimum ejection duration. A host will remain ejected for a period equal to the product of minimum ejection duration and the number of times the host has been

Property	Type	Description
		ejected. This technique allows the system to automatically increase the ejection period for unhealthy upstream servers. format: 1h/1m/1s/1ms. MUST BE >=1ms. Default is 30s.
<code>consecutive5xxErrors</code>	<code>integer</code>	Number of 5xx errors before a host is ejected from the connection pool. When the upstream host is accessed over an opaque TCP connection, connect timeouts, connection error/failure and request failure events qualify as a 5xx error. This feature defaults to 5 but can be disabled by setting the value to 0. Note that <code>consecutive_gateway_errors</code> and <code>consecutive_5xx_errors</code> can be used separately or together. Because the errors counted by <code>consecutive_gateway_errors</code> are also included in <code>consecutive_5xx_errors</code> , if the value of <code>consecutive_gateway_errors</code> is greater than or equal to the value of <code>consecutive_5xx_errors</code> , <code>consecutive_gateway_errors</code> will have no effect.
<code>consecutiveErrors</code>	<code>integer</code>	Number of errors before a host is ejected from the connection pool. Defaults to 5. When the upstream host is accessed over HTTP, a 5xx return code qualifies as an error. When the upstream host is accessed over an opaque TCP connection, connect

Property	Type	Description
		timeouts and connection error/failure events qualify as an error.
<code>consecutiveGatewayErrors</code>	<code>integer</code>	Number of gateway errors before a host is ejected from the connection pool. When the upstream host is accessed over HTTP, a 502, 503, or 504 return code qualifies as a gateway error. When the upstream host is accessed over an opaque TCP connection, connect timeouts and connection error/failure events qualify as a gateway error. This feature is disabled by default or when set to the value 0. Note that <code>consecutive_gateway_errors</code> and <code>consecutive_5xx_errors</code> can be used separately or together. Because the errors counted by <code>consecutive_gateway_errors</code> are also included in <code>consecutive_5xx_errors</code>, if the value of <code>consecutive_gateway_errors</code> is greater than or equal to the value of <code>consecutive_5xx_errors</code>, <code>consecutive_gateway_errors</code> will have no effect.
<code>interval</code>	<code>string</code>	Time interval between ejection sweep analysis. format: 1h/1m/1s/1ms. MUST BE >=1ms. Default is 10s.
<code>maxEjectionPercent</code>	<code>integer</code>	Maximum % of hosts in the load balancing pool for the upstream service that can be

Property	Type	Description
		ejected. Defaults to 10%.
<code>minHealthPercent</code>	<code>integer</code>	Outlier detection will be enabled as long as the associated load balancing pool has at least <code>min_health_percent</code> hosts in healthy mode. When the percentage of healthy hosts in the load balancing pool drops below this threshold, outlier detection will be disabled and the proxy will load balance across all hosts in the pool (healthy and unhealthy). The threshold can be disabled by setting it to 0%. The default is 0% as it's not typically applicable in k8s environments with few pods per service.

`.spec.service.trafficPolicy.portLevelSettings[].port`

Description

Specifies the port name or number of a port on the destination service on which this policy is being applied. Names must comply with DNS label syntax (rfc1035) and therefore cannot collide with numbers. If there are multiple ports on a service with the same protocol the names should be of the form `<protocol-name>-<DNS label>`.

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Valid port name

Property	Type	Description
<code>number</code>	<code>integer</code>	Valid port number

`.spec.service.trafficPolicy.portLevelSettings[].tls`

Description

TLS related settings for connections to the upstream service.

Type

`object`

Required

`mode`

Property	Type	Description
<code>caCertificates</code>	<code>string</code>	OPTIONAL: The path to the file containing certificate authority certificates to use in verifying a presented server certificate. If omitted, the proxy will not verify the server's certificate. Should be empty if mode is <code>ISTIO_MUTUAL</code> .
<code>clientCertificate</code>	<code>string</code>	REQUIRED if mode is <code>MUTUAL</code> . The path to the file holding the client-side TLS certificate to use. Should be empty if mode is <code>ISTIO_MUTUAL</code> .
<code>mode</code>	<code>string</code>	REQUIRED: Indicates whether connections to this port should be secured using TLS. The value of this field determines how TLS is enforced.

Property	Type	Description
<code>privateKey</code>	<code>string</code>	REQUIRED if mode is <code>MUTUAL</code> . The path to the file holding the client's private key. Should be empty if mode is <code>ISTIO_MUTUAL</code> .
<code>sni</code>	<code>string</code>	SNI string to present to the server during TLS handshake. Should be empty if mode is <code>ISTIO_MUTUAL</code> .
<code>subjectAltNames</code>	<code>array</code>	A list of alternate names to verify the subject identity in the certificate. If specified, the proxy will verify that the server certificate's subject alt name matches one of the specified values. Should be empty if mode is <code>ISTIO_MUTUAL</code> .

`.spec.service.trafficPolicy.portLevelSettings[].tls.subjectAltNames`

Description

A list of alternate names to verify the subject identity in the certificate. If specified, the proxy will verify that the server certificate's subject alt name matches one of the specified values. Should be empty if mode is `ISTIO_MUTUAL` .

Type

`array`

`.spec.service.trafficPolicy.portLevelSettings[].tls.subjectAltNames[]`

Type

string

.spec.service.trafficPolicy.tls

Description

TLS related settings for connections to the upstream service.

Type

object

Required

mode

Property	Type	Description
caCertificates	string	OPTIONAL: The path to the file containing certificate authority certificates to use in verifying a presented server certificate. If omitted, the proxy will not verify the server's certificate. Should be empty if mode is <code>ISTIO_MUTUAL</code> .
clientCertificate	string	REQUIRED if mode is <code>MUTUAL</code> . The path to the file holding the client-side TLS certificate to use. Should be empty if mode is <code>ISTIO_MUTUAL</code> .
mode	string	REQUIRED: Indicates whether connections to this port should be secured using TLS. The value of this field determines how TLS is enforced.
privateKey	string	REQUIRED if mode is <code>MUTUAL</code> . The path to the file holding the client's private key. Should be empty if

Property	Type	Description
		mode is <code>ISTIO_MUTUAL</code> .
<code>sni</code>	<code>string</code>	SNI string to present to the server during TLS handshake. Should be empty if mode is <code>ISTIO_MUTUAL</code> .
<code>subjectAltNames</code>	<code>array</code>	A list of alternate names to verify the subject identity in the certificate. If specified, the proxy will verify that the server certificate's subject alt name matches one of the specified values. Should be empty if mode is <code>ISTIO_MUTUAL</code> .

`.spec.service.trafficPolicy.tls.subjectAltNames`

Description

A list of alternate names to verify the subject identity in the certificate. If specified, the proxy will verify that the server certificate's subject alt name matches one of the specified values. Should be empty if mode is `ISTIO_MUTUAL` .

Type

`array`

`.spec.service.trafficPolicy.tls.subjectAltNames[]`

Type

`string`

`.spec.targetRef`

Description

TargetRef references a target resource

Type

object

Required

name

Property	Type	Description
apiVersion	string	API version of the referent
kind	string	Kind of the referent
name	string	Name of the referent
namespace	string	Namespace of the referent

API Endpoints

The following API endpoints are available:

- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/canarytemplates`
 - **DELETE** : delete collection of CanaryTemplate
 - **GET** : list objects of kind CanaryTemplate
 - **POST** : create a new CanaryTemplate
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/canarytemplates/{name}`
 - **DELETE** : delete the specified CanaryTemplate

- **GET** : read the specified CanaryTemplate
- **PATCH** : partially update the specified CanaryTemplate
- **PUT** : replace the specified CanaryTemplate

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/canarytemplates

HTTP method

DELETE

Description

delete collection of CanaryTemplate

HTTP responses

HTTP code	Response body
200 - OK	Status  schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind CanaryTemplate

HTTP responses

HTTP code	Response body
200 - OK	CanaryTemplateList schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new CanaryTemplate

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>CanaryTemplate</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>CanaryTemplate</code> schema
201 - Created	<code>CanaryTemplate</code> schema
202 - Accepted	<code>CanaryTemplate</code> schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/canarytemplates/{name}

HTTP method

`DELETE`

Description

delete the specified CanaryTemplate

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code> schema
202 - Accepted	<code>Status</code> schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified CanaryTemplate

HTTP responses

HTTP code	Response body
200 - OK	<code>CanaryTemplate</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified CanaryTemplate

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields.

Parameter	Type	Description
		This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>CanaryTemplate</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified `CanaryTemplate`

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object,

Parameter	Type	Description
		and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	CanaryTemplate schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	CanaryTemplate schema
201 - Created	CanaryTemplate schema
401 - Unauthorized	Empty

Type

`object`

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources ↗
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds ↗

Property	Type	Description
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	
<code>status</code>	<code>object</code>	

.spec

Type

`object`

Property	Type	Description
<code>interval</code>	<code>string</code>	
<code>jobLabel</code>	<code>string</code>	INSERT ADDITIONAL SPEC FIELDS - desired state of cluster Important: Run "make" to regenerate code after modifying this file
<code>monitorResults</code>	<code>object</code>	SampleLimit defines per-scrape limit on number of scraped samples that will be accepted.
<code>namespaceSelector</code>	<code>object</code>	NamespaceSelector to select which namespaces the Endpoints objects are discovered from.
<code>pingurls</code>	<code>array</code>	A list of urls allowed as part of this ServiceMonitor.

Property	Type	Description
<code>selector</code>	<code>object</code>	Selector transfers labels on the Kubernetes Pod onto the target.
<code>crds</code>	<code>array</code>	A list of crds allowed as part of this CaseMonitor.

`.spec.monitorResults`

Description

SampleLimit defines per-scrape limit on number of scraped samples that will be accepted.

Type

`object`

Property	Type	Description
<code>latestUpdated</code>	<code>string</code>	
<code>targets</code>	<code>array</code>	

`.spec.monitorResults.targets`

Type

`array`

`.spec.monitorResults.targets[]`

Type

`object`

Property	Type	Description
lastErrors	array	
name	string	
running	string	
targetType	string	

.spec.monitorResults.targets[].lastErrors

Type

array

.spec.monitorResults.targets[].lastErrors[]

Type

string

.spec.namespaceSelector

Description

NamespaceSelector to select which namespaces the Endpoints objects are discovered from.

Type

object

Property	Type	Description
matchNames	array	List of namespace names.

.spec.namespaceSelector.matchNames

Description

List of namespace names.

Type

array

.spec.namespaceSelector.matchNames[]

Type

string

.spec.pingurls

Description

A list of urls allowed as part of this ServiceMonitor.

Type

array

.spec.pingurls[]

Type

object

Property	Type	Description
host	string	HTTP path to scrape for metrics.
interval	string	Interval at which metrics should be scraped

Property	Type	Description
<code>name</code>	<code>string</code>	HTTP path to scrape for metrics.
<code>params</code>	<code>object</code>	Optional HTTP URL parameters
<code>path</code>	<code>string</code>	HTTP path to scrape for metrics.
<code>port</code>	<code>string</code>	Name of the service port this url refers to. Mutually exclusive with targetPort.
<code>scheme</code>	<code>string</code>	HTTP scheme to use for scraping.
<code>scrapeTimeout</code>	<code>string</code>	Timeout after which the scrape is ended
<code>tlsConfig</code>	<code>object</code>	TLS configuration to use when scraping the url
<code>url</code>	<code>string</code>	url for whole url
<code>customErrors</code>	<code>array</code>	List of custom errors.

.spec.pingurls[].params

Description

Optional HTTP URL parameters

Type

object

.spec.pingurls[].tlsConfig

Description

TLS configuration to use when scraping the url

Type

object

Property	Type	Description
<code>insecureSkipVerify</code>	<code>boolean</code>	Disable target certificate validation.

.spec.pingurls[].customErrors

Description

List of custom errors.

Type

array

.spec.pingurls[].customErrors[]

Type

string

.spec.selector

Description

Selector transfers labels on the Kubernetes Pod onto the target.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.selector.matchExpressions

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.selector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.selector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.selector.matchExpressions[].values[]

Type

string

.spec.selector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.crd

Description

A list of crds allowed as part of this CaseMonitor.

Type

array

.spec.crd[]

Type

object

Required

name group kind version operator

Property	Type	Description
name	string	name of job.
group	string	group for apigroups GroupVersionKind.
kind	string	kind for GroupVersionKind.

Property	Type	Description
<code>version</code>	<code>string</code>	version for GroupVersionKind.
<code>crdname</code>	<code>string</code>	name of crd.
<code>namespace</code>	<code>string</code>	namespace of crd.
<code>operator</code>	<code>string</code>	operator of the crd, like Exists, HttpPing.
<code>basicAuth</code>	<code>object</code>	BasicAuth allow an http to authenticate over basic authentication
<code>url</code>	<code>string</code>	property name of url
<code>customErrors</code>	<code>array</code>	List of custom errors.

`.spec.crd[].basicAuth`

Description

BasicAuth allow an http to authenticate over basic authentication

Type

`object`

Property	Type	Description
<code>password</code>	<code>string</code>	property name of password.
<code>username</code>	<code>string</code>	property name of username.

`.spec.creds[].customErrors`

Description

List of custom errors.

Type

`array`

`.spec.creds[].customErrors[]`

Type

`string`

`.status`

Type

`object`

API Endpoints

The following API endpoints are available:

- `/apis/asm.alauda.io/v1beta2/namespaces/{namespace}/casemonitors`
 - `DELETE` : delete collection of CaseMonitor

- **GET** : list objects of kind CaseMonitor
- **POST** : create a new CaseMonitor
- `/apis/asm.alauda.io/v1beta2/namespaces/{namespace}/casemonitors/{name}`
 - **DELETE** : delete the specified CaseMonitor
 - **GET** : read the specified CaseMonitor
 - **PATCH** : partially update the specified CaseMonitor
 - **PUT** : replace the specified CaseMonitor
- `/apis/asm.alauda.io/v1beta2/namespaces/{namespace}/casemonitors/{name}/status`
 - **GET** : read status of the specified CaseMonitor
 - **PATCH** : partially update status of the specified CaseMonitor
 - **PUT** : replace status of the specified CaseMonitor

`/apis/asm.alauda.io/v1beta2/namespaces/{namespace}/casemonitors`

HTTP method

DELETE

Description

delete collection of CaseMonitor

HTTP responses

HTTP code	Response body
200 - OK	Status schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind CaseMonitor

HTTP responses

HTTP code	Response body
200 - OK	<code>CaseMonitorList</code> schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new CaseMonitor

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server

Parameter	Type	Description
		will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>CaseMonitor</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>CaseMonitor</code> schema
201 - Created	<code>CaseMonitor</code> schema
202 - Accepted	<code>CaseMonitor</code> schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1beta2/namespaces/{namespace}/casemonitors/{name}

HTTP method

DELETE

Description

delete the specified CaseMonitor

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the

Parameter	Type	Description
		request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	Status  schema
202 - Accepted	Status  schema
401 - Unauthorized	Empty

HTTP method

[GET](#)

Description

read the specified CaseMonitor

HTTP responses

HTTP code	Response body
200 - OK	CaseMonitor  schema
401 - Unauthorized	Empty

HTTP method

[PATCH](#)

Description

partially update the specified CaseMonitor

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun

Parameter	Type	Description
		directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>CaseMonitor</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified CaseMonitor

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>CaseMonitor</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>CaseMonitor</code> schema
201 - Created	<code>CaseMonitor</code> schema

HTTP code	Response body
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1beta2/namespaces/{namespace}/casemonitors/{name}/status

HTTP method

GET

Description

read status of the specified CaseMonitor

HTTP responses

HTTP code	Response body
200 - OK	<code>CaseMonitor</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified CaseMonitor

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing

Parameter	Type	Description
		unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>CaseMonitor</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified CaseMonitor

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>CaseMonitor</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>CaseMonitor</code> schema
201 - Created	<code>CaseMonitor</code> schema

HTTP code	Response body
401 - Unauthorized	Empty

Description

ConnectionPool is the Schema for the connectionpools API

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

Property	Type	Description
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	ConnectionPoolSpec defines the desired state of ConnectionPool
<code>status</code>	<code>object</code>	ConnectionPoolStatus defines the observed state of ConnectionPool

.spec

Description

ConnectionPoolSpec defines the desired state of ConnectionPool

Type

`object`

Property	Type	Description
<code>host</code>	<code>string</code>	INSERT ADDITIONAL SPEC FIELDS - desired state of cluster Important: Run "make" to regenerate code after modifying this file
<code>http</code>	<code>object</code>	Settings applicable to HTTP1.1/HTTP2/GRPC connections.

Property	Type	Description
<code>tcp</code>	<code>object</code>	Settings for common to both HTTP and TCP upstream connections.

`.spec.http`

Description

Settings applicable to HTTP1.1/HTTP2/GRPC connections.

Type

`object`

Property	Type	Description
<code>httpMaxPendingRequests</code>	<code>integer</code>	Maximum number of pending HTTP1 requests to a destination. Default 1024.
<code>httpMaxRequests</code>	<code>integer</code>	Maximum number of HTTP2 requests to a backend. Default 1024.
<code>maxConnections</code>	<code>integer</code>	Maximum number of HTTP1 /TCP connections to a destination host. Default $2^{32}-1$.
<code>maxRequestsPerConnection</code>	<code>integer</code>	parameter 1 disables keep alive. Default 0, meaning "unlimited",

Property	Type	Description
<code>maxRetries</code>	<code>integer</code>	cluster at a given time. Defaults to 3.

`.spec.tcp`

Description

Settings for common to both HTTP and TCP upstream connections.

Type

`object`

Property	Type	Description
<code>connectTimeout</code>	<code>string</code>	TCP connection timeout. format: 1h/1m/1s/1ms. MUST BE >=1ms. Default is 10s.
<code>maxConnections</code>	<code>integer</code>	Maximum number of HTTP1 /TCP connections to a destination host. Default 2^32-1.
<code>tcpKeepalive</code>	<code>object</code>	If set then set SO_KEEPALIVE on the socket to enable TCP Keepalives.

`.spec.tcp.tcpKeepalive`

Description

If set then set SO_KEEPALIVE on the socket to enable TCP Keepalives.

Type

`object`

Property	Type	Description
<code>interval</code>	<code>string</code>	The time duration between keep-alive probes. Default is to use the OS level configuration (unless overridden, Linux defaults to 75s.)
<code>probes</code>	<code>integer</code>	Maximum number of keepalive probes to send without response before deciding the connection is dead. Default is to use the OS level configuration (unless overridden, Linux defaults to 9.)
<code>time</code>	<code>string</code>	The time duration a connection needs to be idle before keep-alive probes start being sent. Default is to use the OS level configuration (unless overridden, Linux defaults to 7200s (ie 2 hours.)

.status

Description

ConnectionPoolStatus defines the observed state of ConnectionPool

Type

`object`

API Endpoints

The following API endpoints are available:

- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/connectionpools`
 - `DELETE` : delete collection of ConnectionPool

- **GET** : list objects of kind ConnectionPool
- **POST** : create a new ConnectionPool
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/connectionpools/{name}`
 - **DELETE** : delete the specified ConnectionPool
 - **GET** : read the specified ConnectionPool
 - **PATCH** : partially update the specified ConnectionPool
 - **PUT** : replace the specified ConnectionPool
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/connectionpools/{name}/status`
 - **GET** : read status of the specified ConnectionPool
 - **PATCH** : partially update status of the specified ConnectionPool
 - **PUT** : replace status of the specified ConnectionPool

`/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/connectionpools`

HTTP method

DELETE

Description

delete collection of ConnectionPool

HTTP responses

HTTP code	Response body
200 - OK	Status schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind `ConnectionPool`

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectionPoolList</code> schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new `ConnectionPool`

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server

Parameter	Type	Description
		will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>ConnectionPool</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectionPool</code> schema
201 - Created	<code>ConnectionPool</code> schema
202 - Accepted	<code>ConnectionPool</code> schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/connectionpools/{name}

HTTP method

DELETE

Description

delete the specified ConnectionPool

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the

Parameter	Type	Description
		request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
202 - Accepted	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

[GET](#)

Description

read the specified ConnectionPool

HTTP responses

HTTP code	Response body
200 - OK	ConnectionPool schema
401 - Unauthorized	Empty

HTTP method

[PATCH](#)

Description

partially update the specified ConnectionPool

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun

Parameter	Type	Description
		directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectionPool</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified ConnectionPool

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>ConnectionPool</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectionPool</code> schema
201 - Created	<code>ConnectionPool</code> schema

HTTP code	Response body
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/connectionpools/{name}/status

HTTP method

`GET`

Description

read status of the specified ConnectionPool

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectionPool</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified ConnectionPool

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing

Parameter	Type	Description
		unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectionPool</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified ConnectionPool

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>ConnectionPool</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectionPool</code> schema
201 - Created	<code>ConnectionPool</code> schema

HTTP code	Response body
401 - Unauthorized	Empty

Description

Domain is the Schema for the domains API

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources ↗
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds ↗

Property	Type	Description
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	
<code>status</code>	<code>object</code>	

`.spec`

Type

`object`

Property	Type	Description
<code>namespaces</code>	<code>array</code>	
<code>domain</code>	<code>string</code>	
<code>cluster</code>	<code>string</code>	

`.spec.namespaces`

Type

`array`

`.spec.namespaces[]`

Type

`object`

Property	Type	Description
namespace	string	

.status

Type

object

API Endpoints

The following API endpoints are available:

- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/domains`
 - `DELETE` : delete collection of Domain
 - `GET` : list objects of kind Domain
 - `POST` : create a new Domain
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/domains/{name}`
 - `DELETE` : delete the specified Domain
 - `GET` : read the specified Domain
 - `PATCH` : partially update the specified Domain
 - `PUT` : replace the specified Domain
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/domains/{name}/status`
 - `GET` : read status of the specified Domain
 - `PATCH` : partially update status of the specified Domain
 - `PUT` : replace status of the specified Domain

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/domains

HTTP method

DELETE

Description

delete collection of Domain

HTTP responses

HTTP code	Response body
200 - OK	Status schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind Domain

HTTP responses

HTTP code	Response body
200 - OK	DomainList schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new Domain

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>Domain</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>Domain</code> schema
201 - Created	<code>Domain</code> schema

HTTP code	Response body
202 - Accepted	Domain schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/domains/{name}

HTTP method

[DELETE](#)

Description

delete the specified Domain

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	Status  schema
202 - Accepted	Status  schema
401 - Unauthorized	Empty

HTTP method

[GET](#)

Description

read the specified Domain

HTTP responses

HTTP code	Response body
200 - OK	<code>Domain</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified Domain

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server

Parameter	Type	Description
		will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>Domain</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified Domain

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the

Parameter	Type	Description
		request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	Domain schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	Domain schema
201 - Created	Domain schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/domains/{name}/status

HTTP method

GET

Description

read status of the specified Domain

HTTP responses

HTTP code	Response body
200 - OK	Domain schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified Domain

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>Domain</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified Domain

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	Domain schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	Domain schema
201 - Created	Domain schema
401 - Unauthorized	Empty

Description

EnvoyFilterBinding is the Schema for the envoyfilterbindings API

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗

Property	Type	Description
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	EnvoyFilterBindingSpec defines the desired state of EnvoyFilterBinding
<code>status</code>	<code>object</code>	EnvoyFilterBindingStatus defines the observed state of EnvoyFilterBinding

.spec

Description

EnvoyFilterBindingSpec defines the desired state of EnvoyFilterBinding

Type

`object`

.status

Description

EnvoyFilterBindingStatus defines the observed state of EnvoyFilterBinding

Type

`object`

API Endpoints

The following API endpoints are available:

- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/envoyfilterbindings`
 - **DELETE** : delete collection of EnvoyFilterBinding
 - **GET** : list objects of kind EnvoyFilterBinding
 - **POST** : create a new EnvoyFilterBinding
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/envoyfilterbindings/{name}`
 - **DELETE** : delete the specified EnvoyFilterBinding
 - **GET** : read the specified EnvoyFilterBinding
 - **PATCH** : partially update the specified EnvoyFilterBinding
 - **PUT** : replace the specified EnvoyFilterBinding
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/envoyfilterbindings/{name}/status`
 - **GET** : read status of the specified EnvoyFilterBinding
 - **PATCH** : partially update status of the specified EnvoyFilterBinding
 - **PUT** : replace status of the specified EnvoyFilterBinding

`/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/envoyfilterbindings`

HTTP method

DELETE

Description

delete collection of EnvoyFilterBinding

HTTP responses

HTTP code	Response body
200 - OK	Status  schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind EnvoyFilterBinding

HTTP responses

HTTP code	Response body
200 - OK	<code>EnvoyFilterBindingList</code> schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new EnvoyFilterBinding

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields.

Parameter	Type	Description
		This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>EnvoyFilterBinding</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>EnvoyFilterBinding</code> schema
201 - Created	<code>EnvoyFilterBinding</code> schema
202 - Accepted	<code>EnvoyFilterBinding</code> schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/envoyfilterbindings/{name}

HTTP method

DELETE

Description

delete the specified EnvoyFilterBinding

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code>  schema
202 - Accepted	<code>Status</code>  schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified EnvoyFilterBinding

HTTP responses

HTTP code	Response body
200 - OK	<code>EnvoyFilterBinding</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified EnvoyFilterBinding

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>EnvoyFilterBinding</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified `EnvoyFilterBinding`

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>EnvoyFilterBinding</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>EnvoyFilterBinding</code> schema

HTTP code	Response body
201 - Created	<code>EnvoyFilterBinding</code> schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/envoyfilterbindings/{name}/status

HTTP method

`GET`

Description

read status of the specified EnvoyFilterBinding

HTTP responses

HTTP code	Response body
200 - OK	<code>EnvoyFilterBinding</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified EnvoyFilterBinding

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>EnvoyFilterBinding</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified EnvoyFilterBinding

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun

Parameter	Type	Description
		directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>EnvoyFilterBinding</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>EnvoyFilterBinding</code> schema
201 - Created	<code>EnvoyFilterBinding</code> schema
401 - Unauthorized	Empty

Description

EnvoyFilterTemplateSync is the Schema for the envoyfiltertemplatesyncs API

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗

Property	Type	Description
metadata	ObjectMeta	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	EnvoyFilterTemplateSyncSpec defines the desired state of EnvoyFilterTemplateSync
status	object	EnvoyFilterTemplateSyncStatus defines the observed state of EnvoyFilterTemplateSync

.spec

Description

EnvoyFilterTemplateSyncSpec defines the desired state of EnvoyFilterTemplateSync

Type

object

Required

templates

Property	Type	Description
templates	array	

.spec.templates

Type

array

.spec.templates[]

Type

object

Required

istioVersion

value

Property	Type	Description
hash	string	
istioVersion	string	
value	string	
version	string	

.status

Description

EnvoyFilterTemplateSyncStatus defines the observed state of EnvoyFilterTemplateSync

Type

object

API Endpoints

The following API endpoints are available:

- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/envoyfiltertemplatesyncs`
 - DELETE** : delete collection of EnvoyFilterTemplateSync
 - GET** : list objects of kind EnvoyFilterTemplateSync
 - POST** : create a new EnvoyFilterTemplateSync

- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/envoyfiltertemplatesyncs/{name}`
 - **DELETE** : delete the specified EnvoyFilterTemplateSync
 - **GET** : read the specified EnvoyFilterTemplateSync
 - **PATCH** : partially update the specified EnvoyFilterTemplateSync
 - **PUT** : replace the specified EnvoyFilterTemplateSync
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/envoyfiltertemplatesyncs/{name}/status`
 - **GET** : read status of the specified EnvoyFilterTemplateSync
 - **PATCH** : partially update status of the specified EnvoyFilterTemplateSync
 - **PUT** : replace status of the specified EnvoyFilterTemplateSync

`/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/envoyfiltertemplatesyncs`

HTTP method

DELETE

Description

delete collection of EnvoyFilterTemplateSync

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind EnvoyFilterTemplateSync

HTTP responses

HTTP code	Response body
200 - OK	<code>EnvoyFilterTemplateSyncList</code> schema
401 - Unauthorized	Empty

HTTP method

`POST`

Description

create a new EnvoyFilterTemplateSync

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+. - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>EnvoyFilterTemplateSync</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>EnvoyFilterTemplateSync</code> schema
201 - Created	<code>EnvoyFilterTemplateSync</code> schema
202 - Accepted	<code>EnvoyFilterTemplateSync</code> schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/envoyfiltertemplatesyncs/{name}

HTTP method

DELETE

Description

delete the specified EnvoyFilterTemplateSync

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
202 - Accepted	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified EnvoyFilterTemplateSync

HTTP responses

HTTP code	Response body
200 - OK	<code>EnvoyFilterTemplateSync</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified `EnvoyFilterTemplateSync`

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>EnvoyFilterTemplateSync</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified `EnvoyFilterTemplateSync`

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+. - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>EnvoyFilterTemplateSync</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>EnvoyFilterTemplateSync</code> schema
201 - Created	<code>EnvoyFilterTemplateSync</code> schema
401 - Unauthorized	Empty

`/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/envoyfiltertemplatesyncs/{name}/status`

HTTP method

GET

Description

read status of the specified EnvoyFilterTemplateSync

HTTP responses

HTTP code	Response body
200 - OK	<code>EnvoyFilterTemplateSync</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified EnvoyFilterTemplateSync

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields.

Parameter	Type	Description
		This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>EnvoyFilterTemplateSync</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified `EnvoyFilterTemplateSync`

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object,

Parameter	Type	Description
		and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	EnvoyFilterTemplateSync schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	EnvoyFilterTemplateSync schema
201 - Created	EnvoyFilterTemplateSync schema
401 - Unauthorized	Empty

Description

ErrorCheck is the Schema for the errorchecks API

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources ↗
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds ↗

Property	Type	Description
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	ErrorCheckSpec defines the desired state of ErrorCheck
<code>status</code>	<code>object</code>	ErrorCheckStatus defines the observed state of ErrorCheck

`.spec`

Description

ErrorCheckSpec defines the desired state of ErrorCheck

Type

`object`

Required

`errlist`

Property	Type	Description
<code>errlist</code>	<code>array</code>	

`.spec.errlist`

Type

`array`

`.spec.errlist[]`

Type

object

Property	Type	Description
errorcode	string	
errordesc	string	
errorlevel	string	
relatename	string	
resourename	string	

.status

Description

ErrorCheckStatus defines the observed state of ErrorCheck

Type

object

API Endpoints

The following API endpoints are available:

- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/errorchecks`
 - **DELETE** : delete collection of ErrorCheck
 - **GET** : list objects of kind ErrorCheck
 - **POST** : create a new ErrorCheck
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/errorchecks/{name}`
 - **DELETE** : delete the specified ErrorCheck

- **GET** : read the specified ErrorCheck
 - **PATCH** : partially update the specified ErrorCheck
 - **PUT** : replace the specified ErrorCheck
- **/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/errorchecks/{name}/status**
 - **GET** : read status of the specified ErrorCheck
 - **PATCH** : partially update status of the specified ErrorCheck
 - **PUT** : replace status of the specified ErrorCheck

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/errorchecks

HTTP method

DELETE

Description

delete collection of ErrorCheck

HTTP responses

HTTP code	Response body
200 - OK	Status schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind ErrorCheck

HTTP responses

HTTP code	Response body
200 - OK	<code>ErrorCheckList</code> schema
401 - Unauthorized	Empty

HTTP method

`POST`

Description

create a new ErrorCheck

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>ErrorCheck</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ErrorCheck</code> schema
201 - Created	<code>ErrorCheck</code> schema
202 - Accepted	<code>ErrorCheck</code> schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/errorchecks/{name}

HTTP method

`DELETE`

Description

delete the specified ErrorCheck

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
202 - Accepted	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

[GET](#)

Description

read the specified ErrorCheck

HTTP responses

HTTP code	Response body
200 - OK	ErrorCheck schema
401 - Unauthorized	Empty

HTTP method

[PATCH](#)

Description

partially update the specified ErrorCheck

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore:

Parameter	Type	Description
		This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>ErrorCheck</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified `ErrorCheck`

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+. - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>ErrorCheck</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ErrorCheck</code> schema
201 - Created	<code>ErrorCheck</code> schema
401 - Unauthorized	Empty

`/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/errorchecks/{name}/status`

HTTP method

GET

Description

read status of the specified `ErrorCheck`

HTTP responses

HTTP code	Response body
200 - OK	<code>ErrorCheck</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified `ErrorCheck`

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields.

Parameter	Type	Description
		This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>ErrorCheck</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified `ErrorCheck`

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object,

Parameter	Type	Description
		and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	ErrorCheck schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	ErrorCheck schema
201 - Created	ErrorCheck schema
401 - Unauthorized	Empty

[Alauda Service Mesh](#) > [API 参考](#) > [Kubernetes APIs](#) > [asm.alauda.io](#) > gatewaydeploy

Description

GatewayDeploy is the Schema for the gatewaydeploys API

Type

object

Specification

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗

Property	Type	Description
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	GatewayDeploySpec defines the desired state of GatewayDeploy
<code>status</code>	<code>object</code>	GatewayDeployStatus defines the observed state of GatewayDeploy

.spec

Description

GatewayDeploySpec defines the desired state of GatewayDeploy

Type

`object`

Required

`global` `k8s` `revision` `type`

Property	Type	Description
<code>accessLogging</code>	<code>object</code>	Access-log configuration.
<code>global</code>	<code>object</code>	Global configuration.

Property	Type	Description
<code>k8s</code>	<code>object</code>	Kubernetes configuration.
<code>revision</code>	<code>string</code>	Gateway Revision.
<code>type</code>	<code>string</code>	Type of gateway.

.spec.accessLogging

Description

Access-log configuration.

Type

`object`

Required

`enabled`

Property	Type	Description
<code>enabled</code>	<code>boolean</code>	

.spec.global

Description

Global configuration.

Type

`object`

Required

podStrategy

Property	Type	Description
autoscaleEnabled	boolean	Whether auto-scaling is enabled.
externalLoadBalancerIPAddress	string	External Load Balancer IP address
podStrategy	string	Anti-affinity strategy.
private	boolean	Visibility to other projects

.spec.k8s

Description

Kubernetes configuration.

Type

object

Property	Type	Description
hpaSpec	object	HPA configuration.
nodeSelector	object	Node Selector, key of the map is node's label key, value of the map is a type of []string that means In operation will be used for node selection.

Property	Type	Description
nodes	array	Deploy Nodes.
podAnnotations	object	Pod's annotations.
replicas	integer	Replicas.
resources	object	Resource limits.
service	object	Port details.
tolerations	array	Taint tolerance.

.spec.k8s.hpaSpec

Description

HPA configuration.

Type

object

Property	Type	Description
maxReplicas	integer	Maximum number of replicas.

Property	Type	Description
<code>minReplicas</code>	<code>integer</code>	Minimum number of replicas.

`.spec.k8s.nodeSelector`

Description

Node Selector, key of the map is node's label key, value of the map is a type of `[]string` that means ``In`` operation will be used for node selection.

Type

`object`

`.spec.k8s.nodes`

Description

Deploy Nodes.

Type

`array`

`.spec.k8s.nodes[]`

Type

`string`

`.spec.k8s.podAnnotations`

Description

Pod's annotations.

Type

`object`

.spec.k8s.resources

Description

Resource limits.

Type

object

Property	Type	Description
<code>limits</code>	object	Maximum resource limits.
<code>requests</code>	object	Minimum resource requirements.

.spec.k8s.resources.limits

Description

Maximum resource limits.

Type

object

Property	Type	Description
<code>cpu</code>	string	CPU.
<code>memory</code>	string	Memory.

.spec.k8s.resources.requests

Description

Minimum resource requirements.

Type

object

Property	Type	Description
cpu	string	CPU.
memory	string	Memory.

.spec.k8s.service

Description

Port details.

Type

object

Property	Type	Description
annotations	object	Service's annotations.
ports	object	Port details.
type	string	Type of service.

.spec.k8s.service.annotations

Description

Service's annotations.

Type

object

.spec.k8s.service.ports

Description

Port details.

Type

object

Property	Type	Description
http	object	HTTP port.
http-https	object	http/https port.
https	object	HTTPS port.
tcp	object	TCP port.

.spec.k8s.service.ports.http

Description

HTTP port.

Type

object

Property	Type	Description
<code>count</code>	<code>integer</code>	Number.
<code>data</code>	<code>array</code>	Port list.

`.spec.k8s.service.ports.http.data`

Description

Port list.

Type

`array`

`.spec.k8s.service.ports.http.data[]`

Type

`object`

Property	Type	Description
<code>items</code>	<code>array</code>	Info of port.
<code>name</code>	<code>string</code>	Name.

`.spec.k8s.service.ports.http.data[].items`

Description

Info of port.

Type

array

.spec.k8s.service.ports.http.data[].items[]

Type

object

Property	Type	Description
externalLoadBalancerPort	integer	External Load Balancer Port
nodePort	integer	Node port.
port	integer	Service port.
protocol	string	Type of protocol.
targetPort	integer	Container port.

.spec.k8s.service.ports.http-https

Description

http/https port.

Type

object

Property	Type	Description
count	integer	Number.
data	array	Port list.

.spec.k8s.service.ports.http-https.data

Description

Port list.

Type

array

.spec.k8s.service.ports.http-https.data[]

Type

object

Property	Type	Description
items	array	Info of port.
name	string	Name.

.spec.k8s.service.ports.http-https.data[].items

Description

Info of port.

Type

array

.spec.k8s.service.ports.http-https.data[].items[]

Type

object

Property	Type	Description
externalLoadBalancerPort	integer	External Load Balancer Port
nodePort	integer	Node port.
port	integer	Service port.
protocol	string	Type of protocol.
targetPort	integer	Container port.

.spec.k8s.service.ports.https

Description

HTTPS port.

Type

object

Property	Type	Description
count	integer	Number.
data	array	Port list.

`.spec.k8s.service.ports.https.data`

Description

Port list.

Type

array

`.spec.k8s.service.ports.https.data[]`

Type

object

Property	Type	Description
items	array	Info of port.
name	string	Name.

`.spec.k8s.service.ports.https.data[].items`

Description

Info of port.

Type

array

.spec.k8s.service.ports.https.data[].items[]

Type

object

Property	Type	Description
externalLoadBalancerPort	integer	External Load Balancer Port
nodePort	integer	Node port.
port	integer	Service port.
protocol	string	Type of protocol.
targetPort	integer	Container port.

.spec.k8s.service.ports.tcp

Description

TCP port.

Type

object

Property	Type	Description
count	integer	Number.
data	array	Port list.

.spec.k8s.service.ports.tcp.data

Description

Port list.

Type

array

.spec.k8s.service.ports.tcp.data[]

Type

object

Property	Type	Description
items	array	Info of port.
name	string	Name.

.spec.k8s.service.ports.tcp.data[].items

Description

Info of port.

Type

array

.spec.k8s.service.ports.tcp.data[].items[]

Type

object

Property	Type	Description
externalLoadBalancerPort	integer	External Load Balancer Port
nodePort	integer	Node port.
port	integer	Service port.
protocol	string	Type of protocol.
targetPort	integer	Container port.

.spec.k8s.tolerations

Description

Taint tolerance.

Type

array

.spec.k8s.tolerations[]

Type

object

Property	Type	Description
effect	string	Scheduling policy.
key	string	Key of taint.
operator	string	Tolerant matching strategy.
value	string	Value of taint.

.status

Description

GatewayDeployStatus defines the observed state of GatewayDeploy

Type

object

Property	Type	Description
availableReplicas	integer	Number of replicas available.

Property	Type	Description
message	string	A human readable message indicating details about why the gateway is in this phase.
phase	string	Status of the gateway workloads and service.
Pods	array	Pod status.
proxy	object	Istio Proxy status
replicas	integer	Expected number of replicas.
service	object	Service status

.status.pods

Description

Pod status.

Type

array

.status.pods[]

Type

object

Property	Type	Description
<code>hostIP</code>	<code>string</code>	Node ip.
<code>phase</code>	<code>string</code>	Status of container.
<code>podIP</code>	<code>string</code>	Pod ip.

.status.proxy

Description

Istio Proxy status

Type

`object`

Property	Type	Description
<code>conditions</code>	<code>array</code>	
<code>reason</code>	<code>string</code>	
<code>revision</code>	<code>object</code>	
<code>state</code>	<code>string</code>	

.status.proxy.conditions

Type

`array`

.status.proxy.conditions[]

Description

Condition contains details for one aspect of the current state of this API Resource. --- This struct is intended for direct use as an array at the field path .status.conditions. For example, type FooStatus struct{ // Represents the observations of a foo's current state. // Known .status.conditions.type are: "Available", "Progressing", and "Degraded" // +patchMergeKey=type // +patchStrategy=merge // +listType=map // +listMapKey=type Conditions []metav1.Condition `json:"conditions,omitempty" patchStrategy:"merge" patchMergeKey:"type" protobuf:"bytes,1,rep,name=conditions"` // other fields }

Type

object

Required

lastTransitionTime

message

reason

status

type

Property	Type	Description
lastTransitionTime	string	lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.
message	string	message is a human readable message indicating details about the transition. This may be an empty string.
observedGeneration	integer	observedGeneration represents the .metadata.generation that the condition was set based upon. For instance, if .metadata.generation is currently 12, but the .status.conditions[x].observedGeneration is 9, the

Property	Type	Description
		condition is out of date with respect to the current state of the instance.
reason	string	reason contains a programmatic identifier indicating the reason for the condition's last transition. Producers of specific condition types may define expected values and meanings for this field, and whether the values are considered a guaranteed API. The value should be a CamelCase string. This field may not be empty.
status	string	status of the condition, one of True, False, Unknown.
type	string	type of condition in CamelCase or in foo.example.com/CamelCase. --- Many .condition.type values are consistent across resources like Available, but because arbitrary conditions can be useful (see .node.status.conditions), the ability to deconflict is important. The regex it matches is (dns1123SubdomainFmt)?(qualifiedNameFmt)

.status.proxy.revision

Type

object

Required

currenttarget

Property	Type	Description
<code>current</code>	<code>string</code>	
<code>target</code>	<code>string</code>	

.status.service

Description

Service status

Type

`object`

Property	Type	Description
<code>loadBalancer</code>	<code>object</code>	LoadBalancer status
<code>ports</code>	<code>array</code>	Ports status

.status.service.loadBalancer

Description

LoadBalancer status

Type

`object`

Property	Type	Description
<code>ingress</code>	<code>array</code>	Ingress is a list containing ingress points for the load-balancer. Traffic intended for the service should be sent to these ingress points.

`.status.service.loadBalancer.ingress`

Description

Ingress is a list containing ingress points for the load-balancer. Traffic intended for the service should be sent to these ingress points.

Type

`array`

`.status.service.loadBalancer.ingress[]`

Description

LoadBalancerIngress represents the status of a load-balancer ingress point: traffic intended for the service should be sent to an ingress point.

Type

`object`

Property	Type	Description
<code>hostname</code>	<code>string</code>	Hostname is set for load-balancer ingress points that are DNS based (typically AWS load-balancers)
<code>ip</code>	<code>string</code>	IP is set for load-balancer ingress points that are IP based (typically GCE or OpenStack load-balancers)

Property	Type	Description
<code>ports</code>	<code>array</code>	Ports is a list of records of service ports If used, every port defined in the service should have an entry in it

`.status.service.loadBalancer.ingress[].ports`

Description

Ports is a list of records of service ports If used, every port defined in the service should have an entry in it

Type

`array`

`.status.service.loadBalancer.ingress[].ports[]`

Type

`object`

Required

`port``protocol`

Property	Type	Description
<code>error</code>	<code>string</code>	Error is to record the problem with the service port The format of the error shall comply with the following rules: - built-in error values shall be specified in this file and those shall use CamelCase names - cloud provider specific error values must have names that comply with the format foo.example.com/CamelCase. --- The regex it matches is (dns1123SubdomainFmt/)?(qualifiedNameFmt)

Property	Type	Description
<code>port</code>	<code>integer</code>	Port is the port number of the service port of which status is recorded here
<code>protocol</code>	<code>string</code>	Protocol is the protocol of the service port of which status is recorded here The supported values are: "TCP", "UDP", "SCTP"

`.status.service.ports`

Description

Ports status

Type

`array`

`.status.service.ports[]`

Description

ServicePort contains information on service's port.

Type

`object`

Required

`port`

Property	Type	Description
<code>appProtocol</code>	<code>string</code>	The application protocol for this port. This field follows standard Kubernetes label syntax. Un-prefixed names are reserved for IANA standard service names (as per RFC-

Property	Type	Description
		6335 and https://www.iana.org/assignments/service-names ↗). Non-standard protocols should use prefixed names such as mycompany.com/my-custom-protocol.
name	string	The name of this port within the service. This must be a DNS_LABEL. All ports within a ServiceSpec must have unique names. When considering the endpoints for a Service, this must match the 'name' field in the EndpointPort. Optional if only one ServicePort is defined on this service.
nodePort	integer	The port on each node on which this service is exposed when type is NodePort or LoadBalancer. Usually assigned by the system. If a value is specified, in-range, and not in use it will be used, otherwise the operation will fail. If not specified, a port will be allocated if this Service requires one. If this field is specified when creating a Service which does not need it, creation will fail. This field will be wiped when updating a Service to no longer need it (e.g. changing type from NodePort to ClusterIP). More info: https://kubernetes.io/docs/concepts/services-networking/service/#type-nodeport ↗
port	integer	The port that will be exposed by this service.
protocol	string	The IP protocol for this port. Supports "TCP", "UDP", and "SCTP". Default is TCP.

Property	Type	Description
<code>targetPort</code>	<code>int</code>	Number or name of the port to access on the pods targeted by the service. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME. If this is a string, it will be looked up as a named port in the target Pod's container ports. If this is not specified, the value of the 'port' field is used (an identity map). This field is ignored for services with clusterIP=None, and should be omitted or set equal to the 'port' field. More info: https://kubernetes.io/docs/concepts/services-networking/service/#defining-a-service

API Endpoints

The following API endpoints are available:

- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/gatewaydeploys`
 - `DELETE` : delete collection of GatewayDeploy
 - `GET` : list objects of kind GatewayDeploy
 - `POST` : create a new GatewayDeploy
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/gatewaydeploys/{name}`
 - `DELETE` : delete the specified GatewayDeploy
 - `GET` : read the specified GatewayDeploy
 - `PATCH` : partially update the specified GatewayDeploy
 - `PUT` : replace the specified GatewayDeploy
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/gatewaydeploys/{name}/status`
 - `GET` : read status of the specified GatewayDeploy

- **PATCH** : partially update status of the specified GatewayDeploy
- **PUT** : replace status of the specified GatewayDeploy

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/gatewaydeploys

HTTP method

DELETE

Description

delete collection of GatewayDeploy

HTTP responses

HTTP code	Response body
200 - OK	Status  schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind GatewayDeploy

HTTP responses

HTTP code	Response body
200 - OK	GatewayDeployList schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new GatewayDeploy

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>GatewayDeploy</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>GatewayDeploy</code> schema

HTTP code	Response body
201 - Created	<code>GatewayDeploy</code> schema
202 - Accepted	<code>GatewayDeploy</code> schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/gatewaydeploys/{name}

HTTP method

`DELETE`

Description

delete the specified GatewayDeploy

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code> schema
202 - Accepted	<code>Status</code> schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified GatewayDeploy

HTTP responses

HTTP code	Response body
200 - OK	<code>GatewayDeploy</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified GatewayDeploy

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the

Parameter	Type	Description
		request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>GatewayDeploy</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified GatewayDeploy

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The

Parameter	Type	Description
		request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	GatewayDeploy schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	GatewayDeploy schema
201 - Created	GatewayDeploy schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/gatewaydeploys/{name}/status

HTTP method

GET

Description

read status of the specified GatewayDeploy

HTTP responses

HTTP code	Response body
200 - OK	<code>GatewayDeploy</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified GatewayDeploy

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>GatewayDeploy</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified GatewayDeploy

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+. - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>GatewayDeploy</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>GatewayDeploy</code> schema
201 - Created	<code>GatewayDeploy</code> schema
401 - Unauthorized	Empty

Description

GatewayGroup is the Schema for the gatewaygroups API

Type

object

Specification

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

Property	Type	Description
metadata	ObjectMeta	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	GatewayGroupSpec defines the desired state of GatewayGroup
status	object	GatewayGroupStatus defines the observed state of GatewayGroup

.spec

Description

GatewayGroupSpec defines the desired state of GatewayGroup

Type

object

Required

- name
- publish

Property	Type	Description
apiPrefixPath	string	Access path to the API group.
apiPrefixRule	string	Access rules for the API group.

Property	Type	Description
<code>apiRoute</code>	<code>array</code>	API group.
<code>gateway</code>	<code>string</code>	
<code>host</code>	<code>string</code>	Access method (domain or non-domain).
<code>name</code>	<code>string</code>	Group name.
<code>publish</code>	<code>boolean</code>	Whether to publish.

.spec.apiRoute

Description

API group.

Type

`array`

.spec.apiRoute[]

Type

`object`

Required

`method`

`msname`

`namespace`

`operationid`

`path`

`port`

Property	Type	Description
<code>method</code>	<code>string</code>	Http Request method.
<code>msname</code>	<code>string</code>	Service name.
<code>namespace</code>	<code>string</code>	namespace.
<code>operationid</code>	<code>string</code>	operation id.
<code>path</code>	<code>string</code>	Request path.
<code>port</code>	<code>integer</code>	Service port.
<code>status</code>	<code>string</code>	Status of group.

.status

Description

GatewayGroupStatus defines the observed state of GatewayGroup

Type

`object`

API Endpoints

The following API endpoints are available:

- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/gatewaygroups`
 - **DELETE** : delete collection of GatewayGroup
 - **GET** : list objects of kind GatewayGroup
 - **POST** : create a new GatewayGroup
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/gatewaygroups/{name}`
 - **DELETE** : delete the specified GatewayGroup
 - **GET** : read the specified GatewayGroup
 - **PATCH** : partially update the specified GatewayGroup
 - **PUT** : replace the specified GatewayGroup
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/gatewaygroups/{name}/status`
 - **GET** : read status of the specified GatewayGroup
 - **PATCH** : partially update status of the specified GatewayGroup
 - **PUT** : replace status of the specified GatewayGroup

`/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/gatewaygroups`

HTTP method

DELETE

Description

delete collection of GatewayGroup

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind GatewayGroup

HTTP responses

HTTP code	Response body
200 - OK	<code>GatewayGroupList</code> schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new GatewayGroup

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a

Parameter	Type	Description
		warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	GatewayGroup schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	GatewayGroup schema
201 - Created	GatewayGroup schema
202 - Accepted	GatewayGroup schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/gatewaygroups/{name}

HTTP method

DELETE

Description

delete the specified GatewayGroup

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	Status <code>schema</code>
202 - Accepted	Status <code>schema</code>
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified GatewayGroup

HTTP responses

HTTP code	Response body
200 - OK	<code>GatewayGroup</code> <code>schema</code>
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified GatewayGroup

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>GatewayGroup</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified GatewayGroup

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>GatewayGroup</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>GatewayGroup</code> schema
201 - Created	<code>GatewayGroup</code> schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/gatewaygroups/{name}/status

HTTP method

`GET`

Description

read status of the specified GatewayGroup

HTTP responses

HTTP code	Response body
200 - OK	<code>GatewayGroup</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified GatewayGroup

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further

Parameter	Type	Description
		processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	GatewayGroup schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified GatewayGroup

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>GatewayGroup</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>GatewayGroup</code> schema
201 - Created	<code>GatewayGroup</code> schema

HTTP code	Response body
401 - Unauthorized	Empty

Description

GatewayProjectBinding is the Schema for the gatewayprojectbindings API

Type

object

Specification

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗

Property	Type	Description
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	GatewayProjectBindingSpec defines the desired state of GatewayProjectBinding
<code>status</code>	<code>object</code>	GatewayProjectBindingStatus defines the observed state of GatewayProjectBinding

.spec

Description

GatewayProjectBindingSpec defines the desired state of GatewayProjectBinding

Type

`object`

Required

`gatewayRef`

Property	Type	Description
<code>gatewayRef</code>	<code>object</code>	Foo is an example field of GatewayProjectBinding. Edit gatewayprojectbinding_types.go to remove/update

.spec.gatewayRef

Description

Foo is an example field of GatewayProjectBinding. Edit gatewayprojectbinding_types.go to remove/update

Type

object

Required

name

namespace

Property	Type	Description
name	string	
namespace	string	

.status

Description

GatewayProjectBindingStatus defines the observed state of GatewayProjectBinding

Type

object

API Endpoints

The following API endpoints are available:

- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/gatewayprojectbindings`
 - DELETE** : delete collection of GatewayProjectBinding
 - GET** : list objects of kind GatewayProjectBinding
 - POST** : create a new GatewayProjectBinding
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/gatewayprojectbindings/{name}`
 - DELETE** : delete the specified GatewayProjectBinding

- **GET** : read the specified GatewayProjectBinding
- **PATCH** : partially update the specified GatewayProjectBinding
- **PUT** : replace the specified GatewayProjectBinding
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/gatewayprojectbindings/{name}/status`
 - **GET** : read status of the specified GatewayProjectBinding
 - **PATCH** : partially update status of the specified GatewayProjectBinding
 - **PUT** : replace status of the specified GatewayProjectBinding

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/gatewayprojectbindings

HTTP method

DELETE

Description

delete collection of GatewayProjectBinding

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind GatewayProjectBinding

HTTP responses

HTTP code	Response body
200 - OK	<code>GatewayProjectBindingList</code> schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new GatewayProjectBinding

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>GatewayProjectBinding</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>GatewayProjectBinding</code> schema
201 - Created	<code>GatewayProjectBinding</code> schema
202 - Accepted	<code>GatewayProjectBinding</code> schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/gatewayprojectbindings/{name}

HTTP method

`DELETE`

Description

delete the specified GatewayProjectBinding

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
202 - Accepted	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

[GET](#)

Description

read the specified GatewayProjectBinding

HTTP responses

HTTP code	Response body
200 - OK	GatewayProjectBinding schema
401 - Unauthorized	Empty

HTTP method

[PATCH](#)

Description

partially update the specified GatewayProjectBinding

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore:

Parameter	Type	Description
		This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>GatewayProjectBinding</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified GatewayProjectBinding

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+. - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>GatewayProjectBinding</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>GatewayProjectBinding</code> schema
201 - Created	<code>GatewayProjectBinding</code> schema
401 - Unauthorized	Empty

`/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/gatewayprojectbindings/{name}/status`

HTTP method

GET

Description

read status of the specified GatewayProjectBinding

HTTP responses

HTTP code	Response body
200 - OK	<code>GatewayProjectBinding</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified GatewayProjectBinding

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields.

Parameter	Type	Description
		This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>GatewayProjectBinding</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified `GatewayProjectBinding`

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object,

Parameter	Type	Description
		and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	GatewayProjectBinding schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	GatewayProjectBinding schema
201 - Created	GatewayProjectBinding schema
401 - Unauthorized	Empty

Description

GlobalRateLimiter is the Schema for the globalratelimiters API

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

Property	Type	Description
metadata	ObjectMeta	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	GlobalRateLimiterSpec defines the desired state of GlobalRateLimiter
status	object	GlobalRateLimiterStatus defines the observed state of GlobalRateLimiter

.spec

Description

GlobalRateLimiterSpec defines the desired state of GlobalRateLimiter

Type

object

Required

- domain
- failureModeDeny
- rules
- serviceType

Property	Type	Description
domain	string	The domain to limit (globally unique).
failureModeDeny	boolean	Failure to deny access mode.

Property	Type	Description
<code>rules</code>	<code>array</code>	List of rate limit rule.
<code>serviceType</code>	<code>string</code>	Service type.

.spec.rules

Description

List of rate limit rule.

Type

`array`

.spec.rules[]

Type

`object`

Required

`descriptors` `name` `rateLimit`

Property	Type	Description
<code>descriptors</code>	<code>array</code>	List of rate limit description information.
<code>name</code>	<code>string</code>	Name.

Property	Type	Description
<code>rateLimit</code>	<code>object</code>	Current limit value configuration.
<code>shadowMode</code>	<code>boolean</code>	Shadow mode.
<code>updatedUtc</code>	<code>integer</code>	Timestamp of update time(s).

`.spec.rules[].descriptors`

Description

List of rate limit description information.

Type

`array`

`.spec.rules[].descriptors[]`

Type

`object`

Required

`kind` `match`

Property	Type	Description
<code>kind</code>	<code>string</code>	Kind of rate limits.

Property	Type	Description
<code>match</code>	<code>object</code>	Rate limit matching.

`.spec.rules[].descriptors[].match`

Description

Rate limit matching.

Type

`object`

Required

`method`

Property	Type	Description
<code>key</code>	<code>string</code>	Match item (required for request header matching).
<code>method</code>	<code>string</code>	Match method.
<code>value</code>	<code>string</code>	Match value.

`.spec.rules[].rateLimit`

Description

Current limit value configuration.

Type

`object`

Property	Type	Description
<code>fixedWindow</code>	<code>object</code>	Fixed window.

`.spec.rules[].rateLimit.fixedWindow`

Description

Fixed window.

Type

`object`

Required

`requestsPerUnit`

`unit`

Property	Type	Description
<code>requestsPerUnit</code>	<code>integer</code>	Number of requests per time unit.
<code>unit</code>	<code>string</code>	Time unit.

`.status`

Description

GlobalRateLimiterStatus defines the observed state of GlobalRateLimiter

Type

`object`

API Endpoints

The following API endpoints are available:

- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/globalratelimiters`
 - **DELETE** : delete collection of GlobalRateLimiter
 - **GET** : list objects of kind GlobalRateLimiter
 - **POST** : create a new GlobalRateLimiter
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/globalratelimiters/{name}`
 - **DELETE** : delete the specified GlobalRateLimiter
 - **GET** : read the specified GlobalRateLimiter
 - **PATCH** : partially update the specified GlobalRateLimiter
 - **PUT** : replace the specified GlobalRateLimiter
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/globalratelimiters/{name}/status`
 - **GET** : read status of the specified GlobalRateLimiter
 - **PATCH** : partially update status of the specified GlobalRateLimiter
 - **PUT** : replace status of the specified GlobalRateLimiter

`/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/globalratelimiters`

HTTP method

DELETE

Description

delete collection of GlobalRateLimiter

HTTP responses

HTTP code	Response body
200 - OK	Status schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind GlobalRateLimiter

HTTP responses

HTTP code	Response body
200 - OK	<code>GlobalRateLimiterList</code> schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new GlobalRateLimiter

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a

Parameter	Type	Description
		warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	GlobalRateLimiter schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	GlobalRateLimiter schema
201 - Created	GlobalRateLimiter schema
202 - Accepted	GlobalRateLimiter schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/globalratelimiters/{name}

HTTP method

DELETE

Description

delete the specified GlobalRateLimiter

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	Status <code>schema</code>
202 - Accepted	Status <code>schema</code>
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified GlobalRateLimiter

HTTP responses

HTTP code	Response body
200 - OK	<code>GlobalRateLimiter</code> <code>schema</code>
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified GlobalRateLimiter

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>GlobalRateLimiter</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified GlobalRateLimiter

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>GlobalRateLimiter</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>GlobalRateLimiter</code> schema
201 - Created	<code>GlobalRateLimiter</code> schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/globalratelimiters/{name}/status

HTTP method

`GET`

Description

read status of the specified GlobalRateLimiter

HTTP responses

HTTP code	Response body
200 - OK	<code>GlobalRateLimiter</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified GlobalRateLimiter

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further

Parameter	Type	Description
		processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	GlobalRateLimiter schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified GlobalRateLimiter

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>GlobalRateLimiter</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>GlobalRateLimiter</code> schema
201 - Created	<code>GlobalRateLimiter</code> schema

HTTP code	Response body
401 - Unauthorized	Empty

Description

IngressGateway is the Schema for the ingressgateways API

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

Property	Type	Description
metadata	ObjectMeta	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	IngressGatewaySpec defines the desired state of IngressGateway
status	object	IngressGatewayStatus defines the observed state of IngressGateway

.spec

Description

IngressGatewaySpec defines the desired state of IngressGateway

Type

object

Required

port

Property	Type	Description
corsPolicy	object	CORS configuration.
gateway	object	GatewayDeploy

Property	Type	Description
gatewayGroupsRef	array	The associated API group.
host	string	Access mode.
httpRoute	object	HTTP route config.
patches	array	List of patches to apply to root virtualservice.
port	object	Port info.
ports	object	Port details.
serviceRoutes	array	The associated service route.
tcpRoute	object	TCP route config
tcpRoutes	array	The associated TCP service route.
tlsDomain	string	TLSDomain which come out of TLS secret

.spec.corsPolicy

Description

CORS configuration.

Type

object

Required

configType enabled

Property	Type	Description
allowCredentials	boolean	Whether to allow the browser to read the contents of the response.
allowHeaders	array	The list of request headers that the server allows for cross-origin requests.
allowMethods	array	The list of cross-origin request methods allowed by the server.
allowOrigins	array	The server allows cross-origin requests for the list of domains.
configType	string	CORS type. The value range is simple or advance, which means "basic configuration" and "advanced configuration" respectively.

Property	Type	Description
<code>enabled</code>	<code>boolean</code>	Whether to enable CORS.
<code>exposeHeaders</code>	<code>array</code>	A list of headers that the server allows browsers with cross-origin requests to access.
<code>maxAge</code>	<code>string</code>	Cross-origin preflight request cache time.

`.spec.corsPolicy.allowHeaders`

Description

The list of request headers that the server allows for cross-origin requests.

Type

`array`

`.spec.corsPolicy.allowHeaders[]`

Type

`string`

`.spec.corsPolicy.allowMethods`

Description

The list of cross-origin request methods allowed by the server.

Type

`array`

`.spec.corsPolicy.allowMethods[]`

Type

`string`

`.spec.corsPolicy.allowOrigins`

Description

The server allows cross-origin requests for the list of domains.

Type

`array`

`.spec.corsPolicy.allowOrigins[]`

Type

`object`

Property	Type	Description
<code>type</code>	<code>string</code>	Match method.
<code>value</code>	<code>string</code>	Value.

`.spec.corsPolicy.exposeHeaders`

Description

A list of headers that the server allows browsers with cross-origin requests to access.

Type

`array`

.spec.corsPolicy.exposeHeaders[]

Type

string

.spec.gateway

Description

GatewayDeploy

Type

object

Required

name

namespace

Property	Type	Description
name	string	The name of gateway.
namespace	string	The namespace of gateway.

.spec.gatewayGroupsRef

Description

The associated API group.

Type

array

.spec.gatewayGroupsRef[]

Type

object

Required

apiPrefixPath

apiPrefixRule

name

Property	Type	Description
apiPrefixPath	string	Access path.
apiPrefixRule	string	Access rule.
corsPolicy	object	CORS configuration.
name	string	Group name.

.spec.gatewayGroupsRef[].corsPolicy

Description

CORS configuration.

Type

object

Required

configType

enabled

Property	Type	Description
allowCredentials	boolean	Whether to allow the browser to read the contents of the response.

Property	Type	Description
<code>allowHeaders</code>	<code>array</code>	The list of request headers that the server allows for cross-origin requests.
<code>allowMethods</code>	<code>array</code>	The list of cross-origin request methods allowed by the server.
<code>allowOrigins</code>	<code>array</code>	The server allows cross-origin requests for the list of domains.
<code>configType</code>	<code>string</code>	CORS type. The value range is simple or advance, which means "basic configuration" and "advanced configuration" respectively.
<code>enabled</code>	<code>boolean</code>	Whether to enable CORS.
<code>exposeHeaders</code>	<code>array</code>	A list of headers that the server allows browsers with cross-origin requests to access.
<code>maxAge</code>	<code>string</code>	Cross-origin preflight request cache time.

`.spec.gatewayGroupsRef[].corsPolicy.allowHeaders`

Description

The list of request headers that the server allows for cross-origin requests.

Type

array

.spec.gatewayGroupsRef[].corsPolicy.allowHeaders[]**Type**

string

.spec.gatewayGroupsRef[].corsPolicy.allowMethods**Description**

The list of cross-origin request methods allowed by the server.

Type

array

.spec.gatewayGroupsRef[].corsPolicy.allowMethods[]**Type**

string

.spec.gatewayGroupsRef[].corsPolicy.allowOrigins**Description**

The server allows cross-origin requests for the list of domains.

Type

array

.spec.gatewayGroupsRef[].corsPolicy.allowOrigins[]**Type**

object

Property	Type	Description
<code>type</code>	<code>string</code>	Match method.
<code>value</code>	<code>string</code>	Value.

`.spec.gatewayGroupsRef[].corsPolicy.exposeHeaders`

Description

A list of headers that the server allows browsers with cross-origin requests to access.

Type

`array`

`.spec.gatewayGroupsRef[].corsPolicy.exposeHeaders[]`

Type

`string`

`.spec.httpRoute`

Description

HTTP route config.

Type

`object`

Required

`matchType`

`uri`

Property	Type	Description
<code>corsPolicy</code>	<code>object</code>	CORS configuration.
<code>headers</code>	<code>array</code>	Match headers.
<code>matchType</code>	<code>string</code>	Match method.
<code>name</code>	<code>string</code>	Service name.
<code>namespace</code>	<code>string</code>	Service namespace.
<code>port</code>	<code>integer</code>	Service port.
<code>redirect</code>	<code>object</code>	Redirect configuration, new attribute in release3.14.
<code>rewrite</code>	<code>string</code>	Request rewrite.
<code>uri</code>	<code>string</code>	Request path.

.spec.httpRoute.corsPolicy

Description

CORS configuration.

Type

object

Required

configType

enabled

Property	Type	Description
allowCredentials	boolean	Whether to allow the browser to read the contents of the response.
allowHeaders	array	The list of request headers that the server allows for cross-origin requests.
allowMethods	array	The list of cross-origin request methods allowed by the server.
allowOrigins	array	The server allows cross-origin requests for the list of domains.
configType	string	CORS type. The value range is simple or advance, which means "basic configuration" and "advanced configuration" respectively.
enabled	boolean	Whether to enable CORS.

Property	Type	Description
<code>exposeHeaders</code>	<code>array</code>	A list of headers that the server allows browsers with cross-origin requests to access.
<code>maxAge</code>	<code>string</code>	Cross-origin preflight request cache time.

`.spec.httpRoute.corsPolicy.allowHeaders`

Description

The list of request headers that the server allows for cross-origin requests.

Type

`array`

`.spec.httpRoute.corsPolicy.allowHeaders[]`

Type

`string`

`.spec.httpRoute.corsPolicy.allowMethods`

Description

The list of cross-origin request methods allowed by the server.

Type

`array`

`.spec.httpRoute.corsPolicy.allowMethods[]`

Type

`string`

`.spec.httpRoute.corsPolicy.allowOrigins`

Description

The server allows cross-origin requests for the list of domains.

Type

array

`.spec.httpRoute.corsPolicy.allowOrigins[]`

Type

object

Property	Type	Description
<code>type</code>	<code>string</code>	Match method.
<code>value</code>	<code>string</code>	Value.

`.spec.httpRoute.corsPolicy.exposeHeaders`

Description

A list of headers that the server allows browsers with cross-origin requests to access.

Type

array

`.spec.httpRoute.corsPolicy.exposeHeaders[]`

Type

string

.spec.httpRoute.headers

Description

Match headers.

Type

array

.spec.httpRoute.headers[]

Type

object

Property	Type	Description
key	string	Key.
type	string	Match method.
value	string	Value.

.spec.httpRoute.redirect

Description

Redirect configuration, new attribute in release3.14.

Type

object

Property	Type	Description
<code>httpsRedirect</code>	<code>boolean</code>	
<code>location</code>	<code>string</code>	
<code>redirectCode</code>	<code>integer</code>	

`.spec.patches`

Description

List of patches to apply to root virtualservice.

Type

`array`

`.spec.patches[]`

Type

`object`

Property	Type	Description
<code>path</code>	<code>string</code>	Path of the form a.[key1:value1].b.[value2] Where [key1:value1] is a selector for a key-value pair to identify a list element and [value2] is a value selector to identify a list element in a leaf list. All path intermediate nodes must exist.

Property	Type	Description
<code>value</code>	<code>string</code>	Value to add, delete or replace. For add, the path should be a new leaf. For delete, value should be unset. For replace, path should reference an existing node. All values are strings but are converted into appropriate type based on schema.

.spec.port

Description

Port info.

Type

`object`

Required

`name`

Property	Type	Description
<code>name</code>	<code>string</code>	Port name, in the formatted of "protocol-port".

.spec.ports

Description

Port details.

Type

`object`

Required

`group`

Property	Type	Description
<code>detail</code>	<code>array</code>	Port details.
<code>group</code>	<code>string</code>	Group name.

.spec.ports.detail

Description

Port details.

Type

`array`

.spec.ports.detail[]

Type

`object`

Required

`port` `protocol` `targetPort`

Property	Type	Description
<code>nodePort</code>	<code>integer</code>	Node port.
<code>port</code>	<code>integer</code>	Service port.

Property	Type	Description
<code>protocol</code>	<code>string</code>	Protocal type.
<code>targetPort</code>	<code>integer</code>	Container port.

`.spec.serviceRoutes`

Description

The associated service route.

Type

`array`

`.spec.serviceRoutes[]`

Type

`object`

Required

`matchType`

`msName`

`msNamespace`

`msPort`

`publish`

`uri`

Property	Type	Description
<code>corsPolicy</code>	<code>object</code>	CORS configuration.
<code>headers</code>	<code>array</code>	Header.

Property	Type	Description
<code>matchType</code>	<code>string</code>	Match method.
<code>msName</code>	<code>string</code>	Service name.
<code>msNamespace</code>	<code>string</code>	Namespace where the service is located.
<code>msPort</code>	<code>integer</code>	Service port.
<code>publish</code>	<code>boolean</code>	Whether to publish.
<code>rewrite</code>	<code>string</code>	Request rewrite.
<code>uri</code>	<code>string</code>	Request path.

`.spec.serviceRoutes[].corsPolicy`

Description

CORS configuration.

Type

`object`

Required

`configType`

`enabled`

Property	Type	Description
<code>allowCredentials</code>	<code>boolean</code>	Whether to allow the browser to read the contents of the response.
<code>allowHeaders</code>	<code>array</code>	The list of request headers that the server allows for cross-origin requests.
<code>allowMethods</code>	<code>array</code>	The list of cross-origin request methods allowed by the server.
<code>allowOrigins</code>	<code>array</code>	The server allows cross-origin requests for the list of domains.
<code>configType</code>	<code>string</code>	CORS type. The value range is simple or advance, which means "basic configuration" and "advanced configuration" respectively.
<code>enabled</code>	<code>boolean</code>	Whether to enable CORS.
<code>exposeHeaders</code>	<code>array</code>	A list of headers that the server allows browsers with cross-origin requests to access.
<code>maxAge</code>	<code>string</code>	Cross-origin preflight request cache time.

.spec.serviceRoutes[].corsPolicy.allowHeaders

Description

The list of request headers that the server allows for cross-origin requests.

Type

array

.spec.serviceRoutes[].corsPolicy.allowHeaders[]

Type

string

.spec.serviceRoutes[].corsPolicy.allowMethods

Description

The list of cross-origin request methods allowed by the server.

Type

array

.spec.serviceRoutes[].corsPolicy.allowMethods[]

Type

string

.spec.serviceRoutes[].corsPolicy.allowOrigins

Description

The server allows cross-origin requests for the list of domains.

Type

array

`.spec.serviceRoutes[].corsPolicy.allowOrigins[]`

Type

`object`

Property	Type	Description
<code>type</code>	<code>string</code>	Match method.
<code>value</code>	<code>string</code>	Value.

`.spec.serviceRoutes[].corsPolicy.exposeHeaders`

Description

A list of headers that the server allows browsers with cross-origin requests to access.

Type

`array`

`.spec.serviceRoutes[].corsPolicy.exposeHeaders[]`

Type

`string`

`.spec.serviceRoutes[].headers`

Description

Header.

Type

`array`

`.spec.serviceRoutes[].headers[]`

Type

`object`

Property	Type	Description
<code>key</code>	<code>string</code>	Key.
<code>type</code>	<code>string</code>	Match method.
<code>value</code>	<code>string</code>	Value.

`.spec.tcpRoute`

Description

TCP route config

Type

`object`

Required

`name``namespace``port`

Property	Type	Description
<code>name</code>	<code>string</code>	Service name.
<code>namespace</code>	<code>string</code>	Namespace where the service is located.

Property	Type	Description
<code>port</code>	<code>integer</code>	Service port.

`.spec.tcpRoutes`

Description

The associated TCP service route.

Type

`array`

`.spec.tcpRoutes[]`

Type

`object`

Required

`gwPort`

`msName`

`msNamespace`

`msPort`

Property	Type	Description
<code>gwPort</code>	<code>integer</code>	Gateway port.
<code>msName</code>	<code>string</code>	Service name.
<code>msNamespace</code>	<code>string</code>	Namespace where the service is located.

Property	Type	Description
<code>msPort</code>	<code>integer</code>	Service port.

.status

Description

IngressGatewayStatus defines the observed state of IngressGateway

Type

`object`

Property	Type	Description
<code>upgrade</code>	<code>object</code>	The status of upgration

.status.upgrade

Description

The status of upgration

Type

`object`

Property	Type	Description
<code>failReason</code>	<code>string</code>	The reason of failed upgration
<code>phase</code>	<code>string</code>	Phase of upgration

API Endpoints

The following API endpoints are available:

- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/ingressgateways`
 - `DELETE` : delete collection of IngressGateway
 - `GET` : list objects of kind IngressGateway
 - `POST` : create a new IngressGateway
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/ingressgateways/{name}`
 - `DELETE` : delete the specified IngressGateway
 - `GET` : read the specified IngressGateway
 - `PATCH` : partially update the specified IngressGateway
 - `PUT` : replace the specified IngressGateway
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/ingressgateways/{name}/status`
 - `GET` : read status of the specified IngressGateway
 - `PATCH` : partially update status of the specified IngressGateway
 - `PUT` : replace status of the specified IngressGateway

`/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/ingressgateways`

HTTP method

`DELETE`

Description

delete collection of IngressGateway

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

[GET](#)

Description

list objects of kind IngressGateway

HTTP responses

HTTP code	Response body
200 - OK	IngressGatewayList schema
401 - Unauthorized	Empty

HTTP method

[POST](#)

Description

create a new IngressGateway

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last

Parameter	Type	Description
		duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	IngressGateway schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	IngressGateway schema
201 - Created	IngressGateway schema
202 - Accepted	IngressGateway schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/ingressgateways/{name}

HTTP method

DELETE

Description

delete the specified IngressGateway

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code> ↗ schema
202 - Accepted	<code>Status</code> ↗ schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified IngressGateway

HTTP responses

HTTP code	Response body
200 - OK	<code>IngressGateway</code> ↗ schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified IngressGateway

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>IngressGateway</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified IngressGateway

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>IngressGateway</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>IngressGateway</code> schema
201 - Created	<code>IngressGateway</code> schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/ingressgateways/{name}/status

HTTP method

`GET`

Description

read status of the specified IngressGateway

HTTP responses

HTTP code	Response body
200 - OK	<code>IngressGateway</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified IngressGateway

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further

Parameter	Type	Description
		processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>IngressGateway</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified IngressGateway

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>IngressGateway</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>IngressGateway</code> schema
201 - Created	<code>IngressGateway</code> schema

HTTP code	Response body
401 - Unauthorized	Empty

Description

IsolatePod is the Schema for the isolatepods API

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗

Property	Type	Description
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	IsolatePodSpec defines the desired state of IsolatePod

.spec

Description

IsolatePodSpec defines the desired state of IsolatePod

Type

`object`

Property	Type	Description
<code>startTime</code>	<code>string</code>	The point at which isolation begins.
<code>ttl</code>	<code>string</code>	The expiration time of the isolation.

API Endpoints

The following API endpoints are available:

- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/isolatepods`
 - `DELETE` : delete collection of IsolatePod
 - `GET` : list objects of kind IsolatePod

- **POST** : create a new IsolatePod
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/isolatepods/{name}`
- **DELETE** : delete the specified IsolatePod
- **GET** : read the specified IsolatePod
- **PATCH** : partially update the specified IsolatePod
- **PUT** : replace the specified IsolatePod

`/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/isolatepods`

HTTP method

DELETE

Description

delete collection of IsolatePod

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind IsolatePod

HTTP responses

HTTP code	Response body
200 - OK	IsolatePodList schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new IsolatePod

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>IsolatePod</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>IsolatePod</code> schema
201 - Created	<code>IsolatePod</code> schema
202 - Accepted	<code>IsolatePod</code> schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/isolatepods/{name}

HTTP method

`DELETE`

Description

delete the specified IsolatePod

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code> schema
202 - Accepted	<code>Status</code> schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified IsolatePod

HTTP responses

HTTP code	Response body
200 - OK	<code>IsolatePod</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified IsolatePod

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a

Parameter	Type	Description
		warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>IsolatePod</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified IsolatePod

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently

Parameter	Type	Description
		dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	IsolatePod schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	IsolatePod schema
201 - Created	IsolatePod schema
401 - Unauthorized	Empty

[Alauda Service Mesh](#) > [API 参考](#) > [Kubernetes APIs](#) > [asm.alauda.io](#) > [jwt policies](#)

Description

JwtPolicy is the Schema for the jwt policies API

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

Property	Type	Description
metadata	ObjectMeta	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	JwtPolicySpec defines the desired state of JwtPolicy
status	object	JwtPolicyStatus defines the observed state of JwtPolicy

.spec

Description

JwtPolicySpec defines the desired state of JwtPolicy

Type

object

Required

enabled

Property	Type	Description
enabled	string	
privateJwk	string	
publicJwk	string	
token	string	

.status

Description

JwtPolicyStatus defines the observed state of JwtPolicy

Type

object

API Endpoints

The following API endpoints are available:

- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/jwtolicies`
 - `DELETE` : delete collection of JwtPolicy
 - `GET` : list objects of kind JwtPolicy
 - `POST` : create a new JwtPolicy
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/jwtolicies/{name}`
 - `DELETE` : delete the specified JwtPolicy
 - `GET` : read the specified JwtPolicy
 - `PATCH` : partially update the specified JwtPolicy
 - `PUT` : replace the specified JwtPolicy
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/jwtolicies/{name}/status`
 - `GET` : read status of the specified JwtPolicy
 - `PATCH` : partially update status of the specified JwtPolicy
 - `PUT` : replace status of the specified JwtPolicy

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/jwtolicies

HTTP method

DELETE

Description

delete collection of JwtPolicy

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind JwtPolicy

HTTP responses

HTTP code	Response body
200 - OK	JwtPolicyList schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new JwtPolicy

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing

Parameter	Type	Description
		unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>JwtPolicy</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>JwtPolicy</code> schema
201 - Created	<code>JwtPolicy</code> schema
202 - Accepted	<code>JwtPolicy</code> schema
401 - Unauthorized	Empty

`/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/jwtolicies/{name}`

HTTP method

DELETE

Description

delete the specified JwtPolicy

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
202 - Accepted	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified JwtPolicy

HTTP responses

HTTP code	Response body
200 - OK	JwtPolicy schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified JwtPolicy

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>JwtPolicy</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified JwtPolicy

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>JwtPolicy</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>JwtPolicy</code> schema
201 - Created	<code>JwtPolicy</code> schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/jwtpolicies/{name}/status

HTTP method

`GET`

Description

read status of the specified JwtPolicy

HTTP responses

HTTP code	Response body
200 - OK	<code>JwtPolicy</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified JwtPolicy

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun

Parameter	Type	Description
		directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>JwtPolicy</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified JwtPolicy

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>JwtPolicy</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>JwtPolicy</code> schema
201 - Created	<code>JwtPolicy</code> schema

HTTP code	Response body
401 - Unauthorized	Empty

Description

LoadBalancer is the Schema for the loadbalancers API

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

Property	Type	Description
metadata	ObjectMeta ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	LoadBalancerSpec defines the desired state of LoadBalancer
status	object	LoadBalancerStatus defines the observed state of LoadBalancer

.spec

Description

LoadBalancerSpec defines the desired state of LoadBalancer

Type

object

Property	Type	Description
consistentHash	object	Consistent Hash-based load balancing can be used to provide soft session affinity based on HTTP headers, cookies or other properties. This load balancing policy is applicable only for HTTP connections. The affinity to a particular destination host will be lost when one or more hosts are added/removed from the destination service.
host	string	

Property	Type	Description
<code>localityLbSetting</code>	<code>object</code>	Optional: only one of distribute or failover can be set. Explicitly specify loadbalancing weight across different zones and geographical locations. Refer to Locality weighted load balancing ↗ If empty, the locality weight is set according to the endpoints number within it.
<code>simple</code>	<code>string</code>	Standard load balancing algorithms that require no tuning.

.spec.consistentHash

Description

Consistent Hash-based load balancing can be used to provide soft session affinity based on HTTP headers, cookies or other properties. This load balancing policy is applicable only for HTTP connections. The affinity to a particular destination host will be lost when one or more hosts are added/removed from the destination service.

Type

`object`

Property	Type	Description
<code>httpCookie</code>	<code>object</code>	Hash based on HTTP cookie.
<code>httpHeaderName</code>	<code>string</code>	Hash based on a specific HTTP header.

Property	Type	Description
<code>minimumRingSize</code>	<code>integer</code>	The minimum number of virtual nodes to use for the hash ring. Defaults to 1024. Larger ring sizes result in more granular load distributions. If the number of hosts in the load balancing pool is larger than the ring size, each host will be assigned a single virtual node.
<code>useSourceIp</code>	<code>boolean</code>	Hash based on the source IP address.

`.spec.consistentHash.httpCookie`

Description

Hash based on HTTP cookie.

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the cookie.
<code>path</code>	<code>string</code>	Path to set for the cookie.
<code>ttl</code>	<code>string</code>	Lifetime of the cookie.

`.spec.localityLbSetting`

Description

Optional: only one of distribute or failover can be set. Explicitly specify loadbalancing weight across different zones and geographical locations. Refer to [Locality weighted load balancing] (https://www.envoyproxy.io/docs/envoy/latest/intro/arch_overview/upstream/load_balancing/locality_weighted_load_balancing)
If empty, the locality weight is set according to the endpoints number within it.

Type

object

Required

enabled

Property	Type	Description
distribute	array	
enabled	boolean	
failover	array	

.spec.localityLbSetting.distribute

Type

array

.spec.localityLbSetting.distribute[]

Type

object

Property	Type	Description
from	string	Originating locality, '/' separated, e.g. 'region/zone/sub_zone'.

Property	Type	Description
<code>to</code>	<code>object</code>	Map of upstream localities to traffic distribution weights. The sum of all weights should be 100. Any locality not present will receive no traffic.

`.spec.localityLbSetting.distribute[].to`

Description

Map of upstream localities to traffic distribution weights. The sum of all weights should be 100. Any locality not present will receive no traffic.

Type

`object`

`.spec.localityLbSetting.failover`

Type

`array`

`.spec.localityLbSetting.failover[]`

Type

`object`

Property	Type	Description
<code>from</code>	<code>string</code>	Originating region.

Property	Type	Description
<code>to</code>	<code>string</code>	Destination region the traffic will fail over to when endpoints in the 'from' region becomes unhealthy.

.status

Description

LoadBalancerStatus defines the observed state of LoadBalancer

Type

`object`

API Endpoints

The following API endpoints are available:

- `/apis/asm.alauda.io/v1beta1/namespaces/{namespace}/loadbalancers`
 - `DELETE` : delete collection of LoadBalancer
 - `GET` : list objects of kind LoadBalancer
 - `POST` : create a new LoadBalancer
- `/apis/asm.alauda.io/v1beta1/namespaces/{namespace}/loadbalancers/{name}`
 - `DELETE` : delete the specified LoadBalancer
 - `GET` : read the specified LoadBalancer
 - `PATCH` : partially update the specified LoadBalancer
 - `PUT` : replace the specified LoadBalancer
- `/apis/asm.alauda.io/v1beta1/namespaces/{namespace}/loadbalancers/{name}/status`
 - `GET` : read status of the specified LoadBalancer

- **PATCH** : partially update status of the specified LoadBalancer
- **PUT** : replace status of the specified LoadBalancer

/apis/asm.alauda.io/v1beta1/namespaces/{namespace}/loadbalancers

HTTP method

DELETE

Description

delete collection of LoadBalancer

HTTP responses

HTTP code	Response body
200 - OK	Status  schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind LoadBalancer

HTTP responses

HTTP code	Response body
200 - OK	LoadBalancerList  schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new LoadBalancer

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>LoadBalancer</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>LoadBalancer</code> schema

HTTP code	Response body
201 - Created	LoadBalancer schema
202 - Accepted	LoadBalancer schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1beta1/namespaces/{namespace}/loadbalancers/{name}

HTTP method

[DELETE](#)

Description

delete the specified LoadBalancer

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	Status  schema
202 - Accepted	Status  schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified LoadBalancer

HTTP responses

HTTP code	Response body
200 - OK	<code>LoadBalancer</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified LoadBalancer

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the

Parameter	Type	Description
		request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>LoadBalancer</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified LoadBalancer

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The

Parameter	Type	Description
		request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	LoadBalancer schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	LoadBalancer schema
201 - Created	LoadBalancer schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1beta1/namespaces/{namespace}/loadbalancers/{name}/status

HTTP method

GET

Description

read status of the specified LoadBalancer

HTTP responses

HTTP code	Response body
200 - OK	<code>LoadBalancer</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified LoadBalancer

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>LoadBalancer</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified LoadBalancer

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+. - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>LoadBalancer</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>LoadBalancer</code> schema
201 - Created	<code>LoadBalancer</code> schema
401 - Unauthorized	Empty

Description

ManualGrayEvent is the Schema for the ManualGrayEvent API

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗

Property	Type	Description
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	ManualGrayEventSpec defines the desired state of ManualGrayEvent
<code>status</code>	<code>object</code>	ManualGrayEventStatus defines the observed state of ManualGrayRelease

.spec

Description

ManualGrayEventSpec defines the desired state of ManualGrayEvent

Type

`object`

Required

`deliverType` `phase`

Property	Type	Description
<code>deliverType</code>	<code>string</code>	deliver type, canary or ab or others
<code>events</code>	<code>array</code>	
<code>finishTime</code>	<code>string</code>	GrayWorkloadDeployTime of this condition

Property	Type	Description
<code>phase</code>	<code>string</code>	

.spec.events

Type

`array`

.spec.events[]

Description

GrayEvent is a status condition for a event

Type

`object`

Required

`step`

Property	Type	Description
<code>err</code>	<code>string</code>	err if the target not valid
<code>lastTransitionTime</code>	<code>string</code>	LastTransitionTime of this condition
<code>message</code>	<code>string</code>	Message associated with this condition
<code>phase</code>	<code>string</code>	phase of this condition

Property	Type	Description
<code>startTime</code>	<code>string</code>	StartTime of this condition
<code>step</code>	<code>string</code>	Type of this condition

.status

Description

ManualGrayEventStatus defines the observed state of ManualGrayRelease

Type

`object`

Property	Type	Description
<code>finishTime</code>	<code>string</code>	GrayWorkloadDeployTime of this condition
<code>phase</code>	<code>string</code>	

API Endpoints

The following API endpoints are available:

- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/manualgrayevents`
 - `DELETE` : delete collection of ManualGrayEvent
 - `GET` : list objects of kind ManualGrayEvent
 - `POST` : create a new ManualGrayEvent
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/manualgrayevents/{name}`

- **DELETE** : delete the specified ManualGrayEvent
- **GET** : read the specified ManualGrayEvent
- **PATCH** : partially update the specified ManualGrayEvent
- **PUT** : replace the specified ManualGrayEvent
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/manualgrayevents/{name}/status`
 - **GET** : read status of the specified ManualGrayEvent
 - **PATCH** : partially update status of the specified ManualGrayEvent
 - **PUT** : replace status of the specified ManualGrayEvent

`/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/manualgrayevents`

HTTP method

DELETE

Description

delete collection of ManualGrayEvent

HTTP responses

HTTP code	Response body
200 - OK	Status schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind ManualGrayEvent

HTTP responses

HTTP code	Response body
200 - OK	<code>ManualGrayEventList</code> schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new ManualGrayEvent

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>ManualGrayEvent</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ManualGrayEvent</code> schema
201 - Created	<code>ManualGrayEvent</code> schema
202 - Accepted	<code>ManualGrayEvent</code> schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/manualgrayevents/{name}

HTTP method

`DELETE`

Description

delete the specified ManualGrayEvent

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
202 - Accepted	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

[GET](#)

Description

read the specified ManualGrayEvent

HTTP responses

HTTP code	Response body
200 - OK	ManualGrayEvent schema
401 - Unauthorized	Empty

HTTP method

[PATCH](#)

Description

partially update the specified ManualGrayEvent

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore:

Parameter	Type	Description
		This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>ManualGrayEvent</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified ManualGrayEvent

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+. - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>ManualGrayEvent</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ManualGrayEvent</code> schema
201 - Created	<code>ManualGrayEvent</code> schema
401 - Unauthorized	Empty

`/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/manualgrayevents/{name}/status`

HTTP method

GET

Description

read status of the specified ManualGrayEvent

HTTP responses

HTTP code	Response body
200 - OK	<code>ManualGrayEvent</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified ManualGrayEvent

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields.

Parameter	Type	Description
		This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>ManualGrayEvent</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified `ManualGrayEvent`

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object,

Parameter	Type	Description
		and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	ManualGrayEvent schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	ManualGrayEvent schema
201 - Created	ManualGrayEvent schema
401 - Unauthorized	Empty

Description

ManualGrayRelease is the Schema for the ManualGrayRelease API

Type

object

Specification

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗

Property	Type	Description
metadata	ObjectMeta	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	ManualGrayReleaseSpec defines the desired state of ManualGrayRelease
status	object	ManualGrayReleaseStatus defines the observed state of ManualGrayRelease

.spec

Description

ManualGrayReleaseSpec defines the desired state of ManualGrayRelease

Type

object

Required

deliverType desiredStep from grayWorkloads rollBack targetRef

Property	Type	Description
deliverType	string	deliver type, canary or ab or others
desiredStep	string	whether the canary deliver is form table or yaml creat

Property	Type	Description
from	string	whether the canary deliver is form table or yaml creat
grayWorkloads	array	
rollBack	boolean	check for if should cangle gray release
targetRef	object	TargetRef references a target resource

.spec.grayWorkloads

Type

array

.spec.grayWorkloads[]

Type

object

Required

suffix

templateSpec

Property	Type	Description
replicas	integer	Name of the referent
suffix	string	canary resources suffix

Property	Type	Description
<code>templateSpec</code>	<code>string</code>	the workload template spec

.spec.targetRef

Description

TargetRef references a target resource

Type

`object`

Required

`name`

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	API version of the referent
<code>kind</code>	<code>string</code>	Kind of the referent
<code>name</code>	<code>string</code>	Name of the referent
<code>namespace</code>	<code>string</code>	Namespace of the referent

.status

Description

ManualGrayReleaseStatus defines the observed state of ManualGrayRelease

Type

object

Required

phase

Property	Type	Description
conditions	array	
grayWorkloadDeployTime	string	GrayWorkloadDeployTime of this condition
grayWorkloadReadyTime	string	GrayWorkloadReadyTime of this condition
phase	string	ManualGrayPhase is a label for the condition of a canary at the current time
targetRefApplyTime	string	TargetRefApplyTime of this condition
targetRefReadyTime	string	TargetRefReadyTime of this condition

.status.conditions

Type

array

.status.conditions[]

Description

ManualCanaryCondition is a status condition for a Canary

Type

object

Required

step

Property	Type	Description
err	string	err if the target not valid
lastTransitionTime	string	LastTransitionTime of this condition
message	string	Message associated with this condition
phase	string	phase of this condition
startTime	string	StartTime of this condition
step	string	Type of this condition

API Endpoints

The following API endpoints are available:

- /apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/manualgrayreleases
 - DELETE** : delete collection of ManualGrayRelease

- **GET** : list objects of kind ManualGrayRelease
- **POST** : create a new ManualGrayRelease
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/manualgrayreleases/{name}`
 - **DELETE** : delete the specified ManualGrayRelease
 - **GET** : read the specified ManualGrayRelease
 - **PATCH** : partially update the specified ManualGrayRelease
 - **PUT** : replace the specified ManualGrayRelease
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/manualgrayreleases/{name}/status`
 - **GET** : read status of the specified ManualGrayRelease
 - **PATCH** : partially update status of the specified ManualGrayRelease
 - **PUT** : replace status of the specified ManualGrayRelease

`/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/manualgrayreleases`

HTTP method

DELETE

Description

delete collection of ManualGrayRelease

HTTP responses

HTTP code	Response body
200 - OK	Status  schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind ManualGrayRelease

HTTP responses

HTTP code	Response body
200 - OK	<code>ManualGrayReleaseList</code> schema
401 - Unauthorized	Empty

HTTP method

`POST`

Description

create a new ManualGrayRelease

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate

Parameter	Type	Description
		fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	ManualGrayRelease schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	ManualGrayRelease schema
201 - Created	ManualGrayRelease schema
202 - Accepted	ManualGrayRelease schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/manualgrayreleases/{name}

HTTP method

DELETE

Description

delete the specified ManualGrayRelease

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will

Parameter	Type	Description
		result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
202 - Accepted	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

[GET](#)

Description

read the specified ManualGrayRelease

HTTP responses

HTTP code	Response body
200 - OK	ManualGrayRelease schema
401 - Unauthorized	Empty

HTTP method

[PATCH](#)

Description

partially update the specified ManualGrayRelease

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>ManualGrayRelease</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified `ManualGrayRelease`

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>ManualGrayRelease</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ManualGrayRelease</code> schema

HTTP code	Response body
201 - Created	<code>ManualGrayRelease</code> schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/manualgrayreleases/{name}/status

HTTP method

`GET`

Description

read status of the specified ManualGrayRelease

HTTP responses

HTTP code	Response body
200 - OK	<code>ManualGrayRelease</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified ManualGrayRelease

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>ManualGrayRelease</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified ManualGrayRelease

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun

Parameter	Type	Description
		directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>ManualGrayRelease</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ManualGrayRelease</code> schema
201 - Created	<code>ManualGrayRelease</code> schema
401 - Unauthorized	Empty

Description

MicroService is the Schema for the microservices API

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗

Property	Type	Description
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	MicroServiceSpec defines the desired state of MicroService
<code>status</code>	<code>object</code>	MicroServiceStatus defines the observed state of MicroService

.spec

Description

MicroServiceSpec defines the desired state of MicroService

Type

`object`

Property	Type	Description
<code>daemonsets</code>	<code>array</code>	
<code>deployments</code>	<code>array</code>	
<code>services</code>	<code>array</code>	
<code>statefulsets</code>	<code>array</code>	

.spec.daemonsets

Type

array

.spec.daemonsets[]

Type

object

Required

name

Property	Type	Description
name	string	
version	string	

.spec.deployments

Type

array

.spec.deployments[]

Type

object

Required

name

Property	Type	Description
name	string	
version	string	

.spec.services

Type

array

.spec.services[]

Type

string

.spec.statefulsets

Type

array

.spec.statefulsets[]

Type

object

Required

name

Property	Type	Description
name	string	
version	string	

.status

Description

MicroServiceStatus defines the observed state of MicroService

Type

object

API Endpoints

The following API endpoints are available:

- `/apis/asm.alauda.io/v1beta1/namespaces/{namespace}/microservices`
 - **DELETE** : delete collection of MicroService
 - **GET** : list objects of kind MicroService
 - **POST** : create a new MicroService
- `/apis/asm.alauda.io/v1beta1/namespaces/{namespace}/microservices/{name}`
 - **DELETE** : delete the specified MicroService
 - **GET** : read the specified MicroService
 - **PATCH** : partially update the specified MicroService
 - **PUT** : replace the specified MicroService
- `/apis/asm.alauda.io/v1beta1/namespaces/{namespace}/microservices/{name}/status`
 - **GET** : read status of the specified MicroService
 - **PATCH** : partially update status of the specified MicroService
 - **PUT** : replace status of the specified MicroService

/apis/asm.alauda.io/v1beta1/namespaces/{namespace}/microservices

HTTP method

DELETE

Description

delete collection of MicroService

HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code> ↗ schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

list objects of kind MicroService

HTTP responses

HTTP code	Response body
200 - OK	<code>MicroServiceList</code> schema
401 - Unauthorized	Empty

HTTP method

`POST`

Description

create a new MicroService

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last

Parameter	Type	Description
		duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	MicroService schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	MicroService schema
201 - Created	MicroService schema
202 - Accepted	MicroService schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1beta1/namespaces/{namespace}/microservices/{name}

HTTP method

DELETE

Description

delete the specified MicroService

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	Status <code>schema</code>
202 - Accepted	Status <code>schema</code>
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified MicroService

HTTP responses

HTTP code	Response body
200 - OK	<code>MicroService</code> <code>schema</code>
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified MicroService

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>MicroService</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified MicroService

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>MicroService</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>MicroService</code> schema
201 - Created	<code>MicroService</code> schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1beta1/namespaces/{namespace}/microservices/{name}/status

HTTP method

`GET`

Description

read status of the specified MicroService

HTTP responses

HTTP code	Response body
200 - OK	<code>MicroService</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified MicroService

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further

Parameter	Type	Description
		processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>MicroService</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified MicroService

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>MicroService</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>MicroService</code> schema
201 - Created	<code>MicroService</code> schema

HTTP code	Response body
401 - Unauthorized	Empty

Description

OutlierDetection is the Schema for the outlierdetections API

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗

Property	Type	Description
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	OutlierDetectionSpec defines the desired state of OutlierDetection
<code>status</code>	<code>object</code>	OutlierDetectionStatus defines the observed state of OutlierDetection

.spec

Description

OutlierDetectionSpec defines the desired state of OutlierDetection

Type

`object`

Property	Type	Description
<code>baseEjectionTime</code>	<code>string</code>	Minimum ejection duration. A host will remain ejected for a period equal to the product of minimum ejection duration and the number of times the host has been ejected. This technique allows the system to automatically increase the ejection period for unhealthy upstream servers. format: 1h/1m/1s/1ms. MUST BE >=1ms. Default is 30s.

Property	Type	Description
<code>consecutiveErrors</code>	<code>integer</code>	Number of errors before a host is ejected from the connection pool. Defaults to 5. When the upstream host is accessed over HTTP, a 502, 503, or 504 return code qualifies as an error. When the upstream host is accessed over an opaque TCP connection, connect timeouts and connection error/failure events qualify as an error.
<code>host</code>	<code>string</code>	
<code>interval</code>	<code>string</code>	Time interval between ejection sweep analysis. format: 1h/1m/1s/1ms. MUST BE >=1ms. Default is 10s.
<code>maxEjectionPercent</code>	<code>integer</code>	Maximum % of hosts in the load balancing pool for the upstream service that can be ejected. Defaults to 10%.

.status

Description

OutlierDetectionStatus defines the observed state of OutlierDetection

Type

`object`

API Endpoints

The following API endpoints are available:

- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/outlierdetections`
 - **DELETE** : delete collection of OutlierDetection
 - **GET** : list objects of kind OutlierDetection
 - **POST** : create a new OutlierDetection
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/outlierdetections/{name}`
 - **DELETE** : delete the specified OutlierDetection
 - **GET** : read the specified OutlierDetection
 - **PATCH** : partially update the specified OutlierDetection
 - **PUT** : replace the specified OutlierDetection
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/outlierdetections/{name}/status`
 - **GET** : read status of the specified OutlierDetection
 - **PATCH** : partially update status of the specified OutlierDetection
 - **PUT** : replace status of the specified OutlierDetection

`/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/outlierdetections`

HTTP method

DELETE

Description

delete collection of OutlierDetection

HTTP responses

HTTP code	Response body
200 - OK	Status  schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind OutlierDetection

HTTP responses

HTTP code	Response body
200 - OK	<code>OutlierDetectionList</code> schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new OutlierDetection

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields.

Parameter	Type	Description
		This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>OutlierDetection</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>OutlierDetection</code> schema
201 - Created	<code>OutlierDetection</code> schema
202 - Accepted	<code>OutlierDetection</code> schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/outlierdetections/{name}

HTTP method

DELETE

Description

delete the specified OutlierDetection

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code>  schema
202 - Accepted	<code>Status</code>  schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified OutlierDetection

HTTP responses

HTTP code	Response body
200 - OK	<code>OutlierDetection</code>  schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified OutlierDetection

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>OutlierDetection</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified `OutlierDetection`

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>OutlierDetection</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>OutlierDetection</code> schema

HTTP code	Response body
201 - Created	<code>OutlierDetection</code> schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/outlierdetections/{name}/status

HTTP method

`GET`

Description

read status of the specified OutlierDetection

HTTP responses

HTTP code	Response body
200 - OK	<code>OutlierDetection</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified OutlierDetection

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>OutlierDetection</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified OutlierDetection

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun

Parameter	Type	Description
		directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>OutlierDetection</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>OutlierDetection</code> schema
201 - Created	<code>OutlierDetection</code> schema
401 - Unauthorized	Empty

Description

ServiceMeshGroup is the Schema for the servicemeshgroups API

Type

object

Specification

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	ServiceMeshGroupSpec defines the desired state of ServiceMeshGroup
status	object	ServiceMeshGroupStatus defines the observed state of ServiceMeshGroup

.spec

Description

ServiceMeshGroupSpec defines the desired state of ServiceMeshGroup

Type

object

Required

primary

Property	Type	Description
caConfig	object	

Property	Type	Description
clusters	array	
groupId	string	
istioConfig	object	
istioVersion	string	
k8sVersion	string	
meshCommonConfig	object	
meshConfig	object	
multiCluster	object	
network	string	
primary	string	Deprecated: PrimaryCluster 仅做为主从结构的兼容性保留，不再使用
selector	object	A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

.spec.caConfig

Type

object

Property	Type	Description
certmanager	object	Certmanager is generated CA for Istio
insecure	object	Deprecated: Insecure, use Certmanager instead For compatibility reasons, insecure CA is used by default

.spec.caConfig.certmanager

Description

Certmanager is generated CA for Istio

Type

object

.spec.caConfig.insecure

Description

Deprecated: Insecure, use Certmanager instead For compatibility reasons, insecure CA is used by default

Type

object

.spec.clusters

Type

array

.spec.clusters[]

Type

string

.spec.istioConfig

Type

object

Property	Type	Description
		Locality-weighted load balancing allows administrators to control the distribution of traffic to endpoints based on the using arbitrary labels that designate a hierarchy of localities in {region}/{zone}/{sub-zone} form. For additional detail refer to [Locality Weight] (https://www.envoyproxy.io/docs/envoy/latest/intro/arch_overview/upstream/load_balancing/locality_weight) Given a mesh with workloads and their service deployed to "us-west/zone1/" and "us-west/zone2/". This example specifies that the traffic will be sent to endpoints in "us-west/zone1/", i.e the same zone, and the remaining 20% will go to endpoints in "us-west/zone2/". A similar setting is specified for traffic originating in "us-west/zone2/".
localityLbSetting	object	<pre>distribute: - from: us-west/zone1/* to: "us-west/zone1/*": 80 "us-west/zone2/*": 20</pre>
		If the goal of the operator is not to distribute load across zones and regions but rather to restrict the regionality of failure, a 'failover' policy can be set. The following example sets up a locality failover policy for regions. Assume a service resides in zones within us-east, us-west & eu-west this example specifies that when endpoints within us-east become unhealthy traffic should failover to endpoints in any zone or sub-zone within eu-west and similarly us-west should failover to us-east.
		<pre>failover: - from: us-east to: eu-west - from: us-west to: us-east</pre>

.spec.istioConfig.localityLbSetting

Description

Locality-weighted load balancing allows administrators to control the distribution of traffic to endpoints based on the localities of where the traffic originates and where it will terminate. These localities are specified using arbitrary labels that designate a hierarchy of localities in {region}/{zone}/{sub-zone} form. For additional detail refer to [Locality Weight] (https://www.envoyproxy.io/docs/envoy/latest/intro/arch_overview/upstream/load_balancing/locality_weight) The following example shows how to setup locality weights mesh-wide. Given a mesh with workloads and their service deployed to "us-west/zone1/" and "us-west/zone2/". This example specifies that when traffic accessing a service originates from workloads in "us-west/zone1/", 80% of the traffic will be sent to endpoints in "us-west/zone1/", i.e the same zone, and the remaining 20% will go to endpoints in "us-west/zone2/". This setup is intended to favor routing traffic to endpoints in the same locality. A similar setting is specified for traffic originating in "us-west/zone2/".

```
yaml distribute: - from: us-west/zone1/* to: "us-west/zone1/*": 80 "us-west/zone2/*": 20 - from: us-west/zone2/* to: "us-west/zone1/*": 20 "us-west/zone2/*": 80
```

If the goal of the operator is not to distribute load across zones and regions but rather to restrict the regionality of failure to meet other operational requirements an operator can set a 'failover' policy instead of a 'distribute' policy. The following example sets up a locality failover policy for regions. Assume a service resides in zones within us-east, us-west & eu-west this example specifies that when endpoints within us-east become unhealthy traffic should failover to endpoints in any zone or sub-zone within eu-west and similarly us-west should failover to us-east.

```
yaml failover: - from: us-east to: eu-west - from: us-west to: us-east
```

Type

object

Property	Type	Description
<code>distribute</code>	array	Explicitly specify loadbalancing weight across different zones and geographical locations. Refer to Locality weighted load balancing ✓ If empty, the locality weight is set according to the endpoints number within it.
<code>enabled</code>	boolean	e.g. true means that turn on locality load balancing for this DestinationRule no matter what mesh wide settings is.
<code>failover</code>	array	Optional: only one of distribute, failover or failoverPriority can be set. Explicitly specify the region traffic will land on when endpoints in local region becomes unhealthy. Should be used together with OutlierDetection to detect unhealthy endpoints. Note: if no OutlierDetection specified, this will not take effect.
<code>failoverPriority</code>	array	failoverPriority is an ordered list of labels used to sort endpoints to do priority based load balancing. This is to support traffic failover across different groups of endpoints. Suppose there are total N labels specified: 1. Endpoints matching all N labels with the client proxy have priority P(0) i.e. the highest priority. 2. Endpoints matching the first N-1 labels with the client proxy have priority P(1) i.e. second highest priority. 3. By extension of this logic, endpoints matching only the first label with the client proxy has priority P(N-1) i.e. second lowest priority. 4. All the other endpoints have priority P(N) i.e. lowest priority. Note: For a label to be considered for match, the previous labels must match, i.e. nth label would be considered matched only if first n-1 labels match. It can be any label specified on both client and server workloads. The following labels which have special semantic meaning are also supported: • <code>topology.istio.io/network</code> is used to match the network metadata of an endpoint, which can be specified by pod/namespace label <code>topology.istio.io/network</code>, sidecar env <code>ISTIO_META_NETWORK</code> or MeshNetworks. - <code>topology.istio.io/cluster</code> is used to match the clusterID of an endpoint, which can be specified by pod label <code>topology.istio.io/cluster</code> or pod env <code>ISTIO_META_CLUSTER_ID</code>. - <code>topology.kubernetes.io/region</code> is used to match the region metadata of an endpoint, which maps to Kubernetes node label <code>topology.kubernetes.io/region</code> or the deprecated label <code>failure-domain.beta.kubernetes.io/region</code>. - <code>topology.kubernetes.io/zone</code> is used to match the zone metadata of an endpoint, which maps to Kubernetes node label <code>topology.kubernetes.io/zone</code> or the deprecated label <code>failure-domain.beta.kubernetes.io/zone</code>. - <code>topology.istio.io/subzone</code> is used to match the subzone metadata of an endpoint, which maps to Istio node label <code>topology.istio.io/subzone</code>. The below topology config indicates the following priority levels: <pre>yaml failoverPriority: - "topology.istio.io/network" - "topology.kubernetes.io/region" - "topology.kubernetes.io/zone" - "topology.istio.io/subzone"</pre> 1. endpoints match same [network, region, zone, subzone] label with the client proxy have the highest priority. 2. endpoints have same [network, region, zone] label but different [subzone] label with the client proxy have the second highest priority. 3. endpoints have same [network, region] label but different [zone] label with the client proxy have the third highest priority. 4. endpoints have same [network] but different [region] labels with the client proxy have the fourth highest priority. 5. all the other endpoints have the same lowest priority. Optional: only one of distribute, failover or failoverPriority can be set. And it should be used together with OutlierDetection to detect unhealthy endpoints, otherwise has no effect.

Description

Explicitly specify loadbalancing weight across different zones and geographical locations. Refer to [Locality weighted load balancing] (https://www.envoyproxy.io/docs/envoy/latest/intro/arch_overview/upstream/load_balancing/locality_weight) If empty, the locality weight is set according to the endpoints number within it.

Type

array

.spec.istioConfig.localityLbSetting.distribute[]

Description

Describes how traffic originating in the 'from' zone or sub-zone is distributed over a set of 'to' zones. Syntax for specifying a zone is {region}/{zone}/{sub-zone} and terminal wildcards are allowed on any segment of the specification. Examples: `\*` - matches all localities `us-west/\*` - all zones and sub-zones within the us-west region `us-west/zone-1/\*` - all sub-zones within us-west/zone-1

Type

object

Property	Type	Description
from	string	Originating locality, '/' separated, e.g. 'region/zone/sub_zone'.
to	object	Map of upstream localities to traffic distribution weights. The sum of all weights should be 100. Any locality not present will receive no traffic.

.spec.istioConfig.localityLbSetting.distribute[].to

Description

Map of upstream localities to traffic distribution weights. The sum of all weights should be 100. Any locality not present will receive no traffic.

Type

object

.spec.istioConfig.localityLbSetting.failover

Description

Optional: only one of distribute, failover or failoverPriority can be set. Explicitly specify the region traffic will land on when endpoints in local region becomes unhealthy. Should be used together with OutlierDetection to detect unhealthy endpoints. Note: if no OutlierDetection specified, this will not take effect.

Type

array

.spec.istioConfig.localityLbSetting.failover[]

Description

Specify the traffic failover policy across regions. Since zone and sub-zone failover is supported by default this only needs to be specified for regions when the operator needs to constrain traffic failover so that the default behavior of failing over to any endpoint globally does not apply. This is useful when failing over traffic across regions would not improve service health or may need to be restricted for other reasons like regulatory controls.

Type

object

Property	Type	Description
from	string	Originating region.
to	string	Destination region the traffic will fail over to when endpoints in the 'from' region becomes unhealthy.

.spec.istioConfig.localityLbSetting.failoverPriority

Description

failoverPriority is an ordered list of labels used to sort endpoints to do priority based load balancing. This is to support traffic failover across different groups of endpoints. Suppose there are total N labels specified: 1. Endpoints matching all N labels with the client proxy have priority P(0) i.e. the highest priority. 2. Endpoints matching the first N-1 labels with the client proxy have priority P(1) i.e. second highest priority. 3. By extension of this logic, endpoints matching only the first label with the client proxy has priority P(N-1) i.e. second lowest priority. 4. All the other endpoints have priority P(N) i.e. lowest priority. Note: For a label to be considered for match, the previous labels must match, i.e. nth label would be considered matched only if first n-1 labels match. It can be any label specified on both client and server workloads. The following labels which have special semantic meaning are also supported: - `topology.istio.io/network` is used to match the network metadata of an endpoint, which can be specified by pod/namespace label `topology.istio.io/network`, sidecar env `ISTIO\_META\_NETWORK` or MeshNetworks. - `topology.istio.io/cluster` is used to match the clusterID of an endpoint, which can be specified by pod label `topology.istio.io/cluster` or pod env `ISTIO\_META\_CLUSTER\_ID`. - `topology.kubernetes.io/region` is used to match the region metadata of an endpoint, which maps to Kubernetes node label `topology.kubernetes.io/region` or the deprecated label `failure-domain.beta.kubernetes.io/region`. - `topology.kubernetes.io/zone` is used to match the zone metadata of an endpoint, which maps to Kubernetes node label `topology.kubernetes.io/zone` or the deprecated label `failure-domain.beta.kubernetes.io/zone`. - `topology.istio.io/subzone` is used to match the subzone metadata of an endpoint, which maps to Istio node label `topology.istio.io/subzone`. The below topology config indicates the following priority levels: ``yaml failoverPriority: - "topology.istio.io/network" - "topology.kubernetes.io/region" - "topology.kubernetes.io/zone" - "topology.istio.io/subzone" `` 1. endpoints match same [network, region, zone, subzone] label with the client proxy have the highest priority. 2. endpoints have same [network, region, zone] label but different [subzone] label with the client proxy have the second highest priority. 3. endpoints have same [network, region] label but different [zone] label with the client proxy have the third highest priority. 4. endpoints have same [network] but different [region] labels with the client proxy have the fourth highest priority. 5. all the other endpoints have the same lowest priority. Optional: only one of distribute, failover or failoverPriority can be set. And it should be used together with `OutlierDetection` to detect unhealthy endpoints, otherwise has no effect.

Type

array

.spec.istioConfig.localityLbSetting.failoverPriority[]

Type

string

.spec.meshCommonConfig

Type

object

Property	Type	Description
elasticsearch	object	
isDefaultMonitor	boolean	

Property	Type	Description
<code>istioSidecar</code>	<code>object</code>	Deprecated: IstioSidecar
<code>jaeger</code>	<code>object</code>	
<code>kafka</code>	<code>object</code>	
<code>monitorType</code>	<code>string</code>	
<code>prometheusBasicAuth</code>	<code>object</code>	
<code>prometheusURL</code>	<code>string</code>	
<code>traceSampling</code>	<code>number</code>	Deprecated: TraceSampling

`.spec.meshCommonConfig.elasticsearch`

Type

`object`

Property	Type	Description
<code>enabled</code>	<code>boolean</code>	
<code>isDefault</code>	<code>boolean</code>	
<code>password</code>	<code>string</code>	
<code>secretName</code>	<code>string</code>	
<code>secretNamespace</code>	<code>string</code>	
<code>url</code>	<code>string</code>	
<code>username</code>	<code>string</code>	

`.spec.meshCommonConfig.istioSidecar`

Description

Deprecated: IstioSidecar

Type

`object`

Property	Type	Description
<code>cpuValue</code>	<code>string</code>	
<code>memoryValue</code>	<code>string</code>	

`.spec.meshCommonConfig.jaeger`

Type

object

Property	Type	Description
indexPrefix	string	
strategy	string	

.spec.meshCommonConfig.kafka

Type

object

Property	Type	Description
authentication	string	
enabled	boolean	
password	string	
secretName	string	
secretNamespace	string	
tls	object	
url	string	
username	string	

.spec.meshCommonConfig.kafka.tls

Type

object

Property	Type	Description
enabled	boolean	
secretName	string	
secretNamespace	string	

.spec.meshCommonConfig.prometheusBasicAuth

Type

object

Property	Type	Description
enabled	boolean	
password	string	
secretName	string	

Property	Type	Description
<code>secretNamespace</code>	<code>string</code>	
<code>username</code>	<code>string</code>	

`.spec.meshConfig`

Type

`object`

`.spec.multiCluster`

Type

`object`

Property	Type	Description
<code>enabled</code>	<code>boolean</code>	Indicates that whether the multi-cluster feature is enabled.
<code>isMultiNetwork</code>	<code>boolean</code>	Indicates whether the servicemeshgroup is targeting a multi-network environment.

`.spec.selector`

Description

A label selector is a label query over a set of resources. The result of `matchLabels` and `matchExpressions` are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.selector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.selector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

`key``operator`

Property	Type	Description
<code>key</code>	<code>string</code>	key is the label key that the selector applies to.
<code>operator</code>	<code>string</code>	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.selector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.selector.matchExpressions[].values[]`

Type

`string`

`.spec.selector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

`object`

`.status`

Description

ServiceMeshGroupStatus defines the observed state of ServiceMeshGroup

Type

`object`

Property	Type	Description
<code>clusterCount</code>	<code>integer</code>	
<code>latestUpdateTime</code>	<code>string</code>	
<code>meshStatus</code>	<code>object</code>	Individual status of each component controlled by the operator. The map key is the name of the component.
<code>message</code>	<code>string</code>	Optional message providing additional information about the existing overall status.
<code>nonReadyClusterCount</code>	<code>integer</code>	
<code>nonReadyClusters</code>	<code>array</code>	
<code>status</code>	<code>string</code>	Overall status of all clusters controlled by the operator. If all clusters have status <code>NONE</code>, overall status is <code>NONE</code>. * If all clusters are <code>HEALTHY</code>, overall status is <code>HEALTHY</code>. * If one or more clusters are <code>RECONCILING</code> and others are <code>HEALTHY</code>, overall status is <code>RECONCILING</code>. * If one or more clusters are <code>UPDATING</code> and others are <code>HEALTHY</code>, overall status is <code>UPDATING</code>. * If clusters are a mix of <code>RECONCILING</code>, <code>UPDATING</code> and <code>HEALTHY</code>, overall status is <code>UPDATING</code>. * If any component is in <code>ERROR</code> state, overall status is <code>ERROR</code>. * If further action is needed for reconciliation to proceed, overall status is <code>ACTION_REQUIRED</code>.

.status.meshStatus

Description

Individual status of each component controlled by the operator. The map key is the name of the component.

Type

`object`

.status.nonReadyClusters

Type

`array`

.status.nonReadyClusters[]

Type

`string`

API Endpoints

The following API endpoints are available:

- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/servicemeshgroups`
 - `DELETE` : delete collection of ServiceMeshGroup

- `GET` : list objects of kind ServiceMeshGroup
- `POST` : create a new ServiceMeshGroup
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/servicemeshgroups/{name}`
 - `DELETE` : delete the specified ServiceMeshGroup
 - `GET` : read the specified ServiceMeshGroup
 - `PATCH` : partially update the specified ServiceMeshGroup
 - `PUT` : replace the specified ServiceMeshGroup
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/servicemeshgroups/{name}/status`
 - `GET` : read status of the specified ServiceMeshGroup
 - `PATCH` : partially update status of the specified ServiceMeshGroup
 - `PUT` : replace status of the specified ServiceMeshGroup

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/servicemeshgroups

HTTP method

DELETE

Description

delete collection of ServiceMeshGroup

HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code> schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind ServiceMeshGroup

HTTP responses

HTTP code	Response body
200 - OK	<code>ServiceMeshGroupList</code> schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new ServiceMeshGroup

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>ServiceMeshGroup</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ServiceMeshGroup</code> schema
201 - Created	<code>ServiceMeshGroup</code> schema
202 - Accepted	<code>ServiceMeshGroup</code> schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/servicemeshgroups/{name}

HTTP method

DELETE

Description

delete the specified ServiceMeshGroup

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code> schema
202 - Accepted	<code>Status</code> schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified ServiceMeshGroup

HTTP responses

HTTP code	Response body
200 - OK	<code>ServiceMeshGroup</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified ServiceMeshGroup

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>ServiceMeshGroup</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified ServiceMeshGroup

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the

Parameter	Type	Description
		default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	ServiceMeshGroup schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	ServiceMeshGroup schema
201 - Created	ServiceMeshGroup schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/servicemeshgroups/{name}/status

HTTP method

GET

Description

read status of the specified ServiceMeshGroup

HTTP responses

HTTP code	Response body
200 - OK	ServiceMeshGroup schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified ServiceMeshGroup

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be

Parameter	Type	Description
		dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>ServiceMeshGroup</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified ServiceMeshGroup

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>ServiceMeshGroup</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ServiceMeshGroup</code> schema
201 - Created	<code>ServiceMeshGroup</code> schema
401 - Unauthorized	Empty

Description

ServiceMesh is the Schema for the servicemeshs API

Type

object

Specification

API Endpoints

The following API endpoints are available:

- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/servicemeshes`
 - `DELETE` : delete collection of ServiceMesh
 - `GET` : list objects of kind ServiceMesh
 - `POST` : create a new ServiceMesh
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/servicemeshes/{name}`
 - `DELETE` : delete the specified ServiceMesh
 - `GET` : read the specified ServiceMesh
 - `PATCH` : partially update the specified ServiceMesh
 - `PUT` : replace the specified ServiceMesh

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/service meshes

HTTP method

[DELETE](#)

Description

delete collection of ServiceMesh

HTTP responses

HTTP code	Response body
200 - OK	Status schema
401 - Unauthorized	Empty

HTTP method

[GET](#)

Description

list objects of kind ServiceMesh

HTTP responses

HTTP code	Response body
200 - OK	ServiceMeshList schema
401 - Unauthorized	Empty

HTTP method

[POST](#)

Description

create a new ServiceMesh

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>ServiceMesh</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ServiceMesh</code> schema
201 - Created	<code>ServiceMesh</code> schema

HTTP code	Response body
202 - Accepted	ServiceMesh schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/servicemeshes/{name}

HTTP method

[DELETE](#)

Description

delete the specified ServiceMesh

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	Status  schema
202 - Accepted	Status  schema
401 - Unauthorized	Empty

HTTP method

[GET](#)

Description

read the specified ServiceMesh

HTTP responses

HTTP code	Response body
200 - OK	<code>ServiceMesh</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified ServiceMesh

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server

Parameter	Type	Description
		will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>ServiceMesh</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified ServiceMesh

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the

Parameter	Type	Description
		request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	ServiceMesh schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	ServiceMesh schema
201 - Created	ServiceMesh schema
401 - Unauthorized	Empty

Description

SidecarBypass is the Schema for the sidecarbypasses API

Type

object

Specification

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

Property	Type	Description
metadata	ObjectMeta	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	SidecarBypassSpec defines the desired state of SidecarBypass
status	object	SidecarBypassStatus defines the observed state of SidecarBypass

.spec

Description

SidecarBypassSpec defines the desired state of SidecarBypass

Type

object

.status

Description

SidecarBypassStatus defines the observed state of SidecarBypass

Type

object

Property	Type	Description
conditions	array	Current state of SidecarBypass

Property	Type	Description
<code>message</code>	<code>string</code>	A human-readable message indicating details about why the SidecarBypass is in this condition.
<code>phase</code>	<code>string</code>	The phase of a SidecarBypass is a simple, high-level summary of where the SidecarBypass is in its lifecycle.
<code>reason</code>	<code>string</code>	A brief CamelCase message indicating details about why the SidecarBypass is in this state.
<code>state</code>	<code>object</code>	Details about the SidecarBypass's current condition.

.status.conditions

Description

Current state of SidecarBypass

Type

`array`

.status.conditions[]

Description

Condition contains details for one aspect of the current state of this API Resource. --- This struct is intended for direct use as an array at the field path `.status.conditions`. For example, type `FooStatus struct{ // Represents the observations of a foo's current state. // Known .status.conditions.type are: "Available", "Progressing", and "Degraded" // +patchMergeKey=type // +patchStrategy=merge // +listType=map // +listMapKey=type`

Conditions []metav1.Condition `json:"conditions,omitempty" patchStrategy:"merge" patchMergeKey:"type" protobuf:"bytes,1,rep,name=conditions" // other fields }

Type

object

Required

lastTransitionTime message reason status type

Property	Type	Description
lastTransitionTime	string	lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.
message	string	message is a human readable message indicating details about the transition. This may be an empty string.
observedGeneration	integer	observedGeneration represents the .metadata.generation that the condition was set based upon. For instance, if .metadata.generation is currently 12, but the .status.conditions[x].observedGeneration is 9, the condition is out of date with respect to the current state of the instance.
reason	string	reason contains a programmatic identifier indicating the reason for the condition's last transition. Producers of specific condition types may define expected values and meanings for this

Property	Type	Description
		field, and whether the values are considered a guaranteed API. The value should be a CamelCase string. This field may not be empty.
status	string	status of the condition, one of True, False, Unknown.
type	string	type of condition in CamelCase or in foo.example.com/CamelCase. --- Many .condition.type values are consistent across resources like Available, but because arbitrary conditions can be useful (see .node.status.conditions), the ability to deconflict is important. The regex it matches is (dns1123SubdomainFmt)?(qualifiedNameFmt)

.status.state

Description

Details about the SidecarBypass's current condition.

Type

object

Property	Type	Description
bypassing	object	Details about a bypassing bypass state.

Property	Type	Description
<code>ready</code>	<code>object</code>	Details about a ready bypass state.
<code>restored</code>	<code>object</code>	Details about a restored bypass state.
<code>restoring</code>	<code>object</code>	Details about a restoring bypass state.

.status.state.bypassing

Description

Details about a bypassing bypass state.

Type

`object`

Property	Type	Description
<code>message</code>	<code>string</code>	Message regarding why the SidecarBypass is not yet running.
<code>reason</code>	<code>string</code>	(brief) reason the SidecarBypass is not yet running.

.status.state.ready

Description

Details about a ready bypass state.

Type

object

Property	Type	Description
readyAt	string	Time at which the bypass is ready.

.status.state.restored

Description

Details about a restored bypass state.

Type

object

Property	Type	Description
restoredAt	string	Time at which the bypass is restored.

.status.state.restoring

Description

Details about a restoring bypass state.

Type

object

Property	Type	Description
message	string	Message regarding why the SidecarBypass is not yet terminated.

Property	Type	Description
<code>reason</code>	<code>string</code>	(brief) reason the SidecarBypass is not yet terminated.

API Endpoints

The following API endpoints are available:

- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/sidecarbypasses`
 - `DELETE` : delete collection of SidecarBypass
 - `GET` : list objects of kind SidecarBypass
 - `POST` : create a new SidecarBypass
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/sidecarbypasses/{name}`
 - `DELETE` : delete the specified SidecarBypass
 - `GET` : read the specified SidecarBypass
 - `PATCH` : partially update the specified SidecarBypass
 - `PUT` : replace the specified SidecarBypass
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/sidecarbypasses/{name}/status`
 - `GET` : read status of the specified SidecarBypass
 - `PATCH` : partially update status of the specified SidecarBypass
 - `PUT` : replace status of the specified SidecarBypass

`/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/sidecarbypasses`

HTTP method

DELETE

Description

delete collection of SidecarBypass

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind SidecarBypass

HTTP responses

HTTP code	Response body
200 - OK	SidecarBypassList schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new SidecarBypass

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further

Parameter	Type	Description
		processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>SidecarBypass</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>SidecarBypass</code> schema
201 - Created	<code>SidecarBypass</code> schema
202 - Accepted	<code>SidecarBypass</code> schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/sidecarbypasses/{name}

HTTP method

DELETE

Description

delete the specified SidecarBypass

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
202 - Accepted	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified SidecarBypass

HTTP responses

HTTP code	Response body
200 - OK	<code>SidecarBypass</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified SidecarBypass

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>SidecarBypass</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified SidecarBypass

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+. - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>SidecarBypass</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>SidecarBypass</code> schema
201 - Created	<code>SidecarBypass</code> schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/sidecarbypasses/{name}/status

HTTP method

GET

Description

read status of the specified SidecarBypass

HTTP responses

HTTP code	Response body
200 - OK	<code>SidecarBypass</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified SidecarBypass

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields.

Parameter	Type	Description
		This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>SidecarBypass</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified SidecarBypass

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object,

Parameter	Type	Description
		and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	SidecarBypass schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	SidecarBypass schema
201 - Created	SidecarBypass schema
401 - Unauthorized	Empty

Description

SidecarHotUpgradePlan is the Schema for the sidecarhotupgradeplans API

Type

object

Specification

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗

Property	Type	Description
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	
<code>status</code>	<code>object</code>	SidecarHotUpgradePlanStatus

.spec

Type

`object`

Required

`targetSidecarVersion`

Property	Type	Description
<code>gracePeriod</code>	<code>string</code>	Tolerance time. Default 60s.
<code>targetSidecarVersion</code>	<code>string</code>	The target version of the hot update.

.status

Description

SidecarHotUpgradePlanStatus

Type

`object`

Property	Type	Description
<code>observedRef</code>	<code>object</code>	
<code>progress</code>	<code>object</code>	HotUpgradeProgress
<code>reason</code>	<code>string</code>	
<code>state</code>	<code>string</code>	

`.status.observedRef`

Type

`object`

Property	Type	Description
<code>generation</code>	<code>integer</code>	
<code>uid</code>	<code>string</code>	UID is a type that holds unique ID values, including UUIDs. Because we don't ONLY use UUIDs, this is an alias to string. Being a type captures intent and helps make sure that UUIDs and names do not get conflated.

`.status.progress`

Description

HotUpgradeProgress

Type

`object`

Property	Type	Description
<code>failed</code>	<code>integer</code>	
<code>Pods</code>	<code>array</code>	
<code>success</code>	<code>integer</code>	
<code>total</code>	<code>integer</code>	

`.status.progress.pods`

Type

`array`

`.status.progress.pods[]`

Type

`object`

Required

`name``target`

Property	Type	Description
<code>message</code>	<code>string</code>	
<code>name</code>	<code>string</code>	
<code>planUID</code>	<code>string</code>	UID is a type that holds unique ID values, including UUIDs. Because we don't ONLY use UUIDs, this is an alias to string. Being a type captures intent and helps make sure that UUIDs and names do not get conflated.
<code>state</code>	<code>string</code>	

Property	Type	Description
target	string	

API Endpoints

The following API endpoints are available:

- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/sidecarhotupgradeplans`
 - DELETE** : delete collection of SidecarHotUpgradePlan
 - GET** : list objects of kind SidecarHotUpgradePlan
 - POST** : create a new SidecarHotUpgradePlan
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/sidecarhotupgradeplans/{name}`
 - DELETE** : delete the specified SidecarHotUpgradePlan
 - GET** : read the specified SidecarHotUpgradePlan
 - PATCH** : partially update the specified SidecarHotUpgradePlan
 - PUT** : replace the specified SidecarHotUpgradePlan
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/sidecarhotupgradeplans/{name}/status`
 - GET** : read status of the specified SidecarHotUpgradePlan
 - PATCH** : partially update status of the specified SidecarHotUpgradePlan
 - PUT** : replace status of the specified SidecarHotUpgradePlan

`/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/sidecarhotupgradeplans`

HTTP method

DELETE

Description

delete collection of SidecarHotUpgradePlan

HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code>  schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind SidecarHotUpgradePlan

HTTP responses

HTTP code	Response body
200 - OK	<code>SidecarHotUpgradePlanList</code>  schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new SidecarHotUpgradePlan

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+. - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>SidecarHotUpgradePlan</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>SidecarHotUpgradePlan</code> schema
201 - Created	<code>SidecarHotUpgradePlan</code> schema
202 - Accepted	<code>SidecarHotUpgradePlan</code> schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/sidecarhotupgradeplans/{name}

HTTP method

DELETE

Description

delete the specified SidecarHotUpgradePlan

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
202 - Accepted	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified SidecarHotUpgradePlan

HTTP responses

HTTP code	Response body
200 - OK	<code>SidecarHotUpgradePlan</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified SidecarHotUpgradePlan

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>SidecarHotUpgradePlan</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified SidecarHotUpgradePlan

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+. - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>SidecarHotUpgradePlan</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>SidecarHotUpgradePlan</code> schema
201 - Created	<code>SidecarHotUpgradePlan</code> schema
401 - Unauthorized	Empty

`/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/sidecarhotupgradeplans/{name}/status`

HTTP method

GET

Description

read status of the specified SidecarHotUpgradePlan

HTTP responses

HTTP code	Response body
200 - OK	<code>SidecarHotUpgradePlan</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified SidecarHotUpgradePlan

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields.

Parameter	Type	Description
		This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>SidecarHotUpgradePlan</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified SidecarHotUpgradePlan

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object,

Parameter	Type	Description
		and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	SidecarHotUpgradePlan schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	SidecarHotUpgradePlan schema
201 - Created	SidecarHotUpgradePlan schema
401 - Unauthorized	Empty

Description

WasmModuleInstall is the Schema for the wasmmoduleinstalls API

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

Property	Type	Description
metadata	ObjectMeta	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	WasmModuleInstallSpec defines the desired state of WasmModuleInstall
status	object	WasmModuleInstallStatus defines the observed state of WasmModuleInstall

.spec

Description

WasmModuleInstallSpec defines the desired state of WasmModuleInstall

Type

object

Required

- image
- targetMesh
- template

Property	Type	Description
image	object	
imagePullOptions	object	
moduleConfiguration	object	
targetMesh	string	
template	object	

.spec.image

Type

object

Required

repo

tag

Property	Type	Description
repo	string	
tag	string	

.spec.imagePullOptions

Type

object

Property	Type	Description
credentialsId	string	

.spec.moduleConfiguration

Type

object

Required

type

Property	Type	Description
jsonData	object	
stringData	string	

Property	Type	Description
<code>type</code>	<code>string</code>	

`.spec.moduleConfiguration.jsonData`

Type

`object`

`.spec.template`

Type

`object`

Required

`templateRef`

Property	Type	Description
<code>templateRef</code>	<code>string</code>	
<code>userValues</code>	<code>string</code>	

`.status`

Description

WasmModuleInstallStatus defines the observed state of WasmModuleInstall

Type

`object`

API Endpoints

The following API endpoints are available:

- `/apis/asm.alauda.io/v1beta1/namespaces/{namespace}/wasmmoduleinstalls`
 - **DELETE** : delete collection of WasmModuleInstall
 - **GET** : list objects of kind WasmModuleInstall
 - **POST** : create a new WasmModuleInstall
- `/apis/asm.alauda.io/v1beta1/namespaces/{namespace}/wasmmoduleinstalls/{name}`
 - **DELETE** : delete the specified WasmModuleInstall
 - **GET** : read the specified WasmModuleInstall
 - **PATCH** : partially update the specified WasmModuleInstall
 - **PUT** : replace the specified WasmModuleInstall
- `/apis/asm.alauda.io/v1beta1/namespaces/{namespace}/wasmmoduleinstalls/{name}/status`
 - **GET** : read status of the specified WasmModuleInstall
 - **PATCH** : partially update status of the specified WasmModuleInstall
 - **PUT** : replace status of the specified WasmModuleInstall

`/apis/asm.alauda.io/v1beta1/namespaces/{namespace}/wasmmoduleinstalls`

HTTP method

DELETE

Description

delete collection of WasmModuleInstall

HTTP responses

HTTP code	Response body
200 - OK	Status  schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

list objects of kind WasmModuleInstall

HTTP responses

HTTP code	Response body
200 - OK	<code>WasmModuleInstallList</code> schema
401 - Unauthorized	Empty

HTTP method

`POST`

Description

create a new WasmModuleInstall

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields.

Parameter	Type	Description
		This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>WasmModuleInstall</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>WasmModuleInstall</code> schema
201 - Created	<code>WasmModuleInstall</code> schema
202 - Accepted	<code>WasmModuleInstall</code> schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1beta1/namespaces/{namespace}/wasmmoduleinstalls/{name}

HTTP method

`DELETE`

Description

delete the specified WasmModuleInstall

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code>  schema
202 - Accepted	<code>Status</code>  schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified WasmModuleInstall

HTTP responses

HTTP code	Response body
200 - OK	<code>WasmModuleInstall</code>  schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified WasmModuleInstall

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>WasmModuleInstall</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified `WasmModuleInstall`

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>WasmModuleInstall</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>WasmModuleInstall</code> schema

HTTP code	Response body
201 - Created	<code>WasmModuleInstall</code> schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1beta1/namespaces/{namespace}/wasmmoduleinstalls/{name}/status

HTTP method

`GET`

Description

read status of the specified WasmModuleInstall

HTTP responses

HTTP code	Response body
200 - OK	<code>WasmModuleInstall</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified WasmModuleInstall

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>WasmModuleInstall</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified WasmModuleInstall

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun

Parameter	Type	Description
		directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>WasmModuleInstall</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>WasmModuleInstall</code> schema
201 - Created	<code>WasmModuleInstall</code> schema
401 - Unauthorized	Empty

Description

WasmModuleTemplate is the Schema for the wasmmoduletemplates API

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗

Property	Type	Description
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	WasmModuleTemplateSpec defines the desired state of WasmModuleTemplate
<code>status</code>	<code>object</code>	WasmModuleTemplateStatus defines the observed state of WasmModuleTemplate

.spec

Description

WasmModuleTemplateSpec defines the desired state of WasmModuleTemplate

Type

`object`

Required

`imageRepo` `template`

Property	Type	Description
<code>imagePullOptions</code>	<code>object</code>	
<code>imageRepo</code>	<code>string</code>	
<code>template</code>	<code>object</code>	

.spec.imagePullOptions

Type

object

Property	Type	Description
credentialsId	string	

.spec.template

Type

object

Required

data type

Property	Type	Description
data	string	
type	string	
values	string	

.status

Description

WasmModuleTemplateStatus defines the observed state of WasmModuleTemplate

Type

object

API Endpoints

The following API endpoints are available:

- `/apis/asm.alauda.io/v1beta1/namespaces/{namespace}/wasmmoduletemplates`
 - **DELETE** : delete collection of WasmModuleTemplate
 - **GET** : list objects of kind WasmModuleTemplate
 - **POST** : create a new WasmModuleTemplate
- `/apis/asm.alauda.io/v1beta1/namespaces/{namespace}/wasmmoduletemplates/{name}`
 - **DELETE** : delete the specified WasmModuleTemplate
 - **GET** : read the specified WasmModuleTemplate
 - **PATCH** : partially update the specified WasmModuleTemplate
 - **PUT** : replace the specified WasmModuleTemplate
- `/apis/asm.alauda.io/v1beta1/namespaces/{namespace}/wasmmoduletemplates/{name}/status`
 - **GET** : read status of the specified WasmModuleTemplate
 - **PATCH** : partially update status of the specified WasmModuleTemplate
 - **PUT** : replace status of the specified WasmModuleTemplate

`/apis/asm.alauda.io/v1beta1/namespaces/{namespace}/wasmmoduletemplates`

HTTP method

DELETE

Description

delete collection of WasmModuleTemplate

HTTP responses

HTTP code	Response body
200 - OK	Status schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind WasmModuleTemplate

HTTP responses

HTTP code	Response body
200 - OK	<code>WasmModuleTemplateList</code> schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new WasmModuleTemplate

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields.

Parameter	Type	Description
		This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>WasmModuleTemplate</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>WasmModuleTemplate</code> schema
201 - Created	<code>WasmModuleTemplate</code> schema
202 - Accepted	<code>WasmModuleTemplate</code> schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1beta1/namespaces/{namespace}/wasmmoduletemplates/{name}

HTTP method

`DELETE`

Description

delete the specified WasmModuleTemplate

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code>  schema
202 - Accepted	<code>Status</code>  schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified WasmModuleTemplate

HTTP responses

HTTP code	Response body
200 - OK	<code>WasmModuleTemplate</code>  schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified WasmModuleTemplate

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>WasmModuleTemplate</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified `WasmModuleTemplate`

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>WasmModuleTemplate</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>WasmModuleTemplate</code> schema

HTTP code	Response body
201 - Created	<code>WasmModuleTemplate</code> schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1beta1/namespaces/{namespace}/wasmmoduletemplates/{name}/status

HTTP method

`GET`

Description

read status of the specified WasmModuleTemplate

HTTP responses

HTTP code	Response body
200 - OK	<code>WasmModuleTemplate</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified WasmModuleTemplate

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>WasmModuleTemplate</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified WasmModuleTemplate

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun

Parameter	Type	Description
		directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>WasmModuleTemplate</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>WasmModuleTemplate</code> schema
201 - Created	<code>WasmModuleTemplate</code> schema
401 - Unauthorized	Empty

Type

`object`

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources ↗
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds ↗

Property	Type	Description
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	
<code>status</code>	<code>object</code>	

.spec

Type

`object`

Required

`destmatch` `allowlist`

Property	Type	Description
<code>allowlist</code>	<code>array</code>	
<code>destmatch</code>	<code>object</code>	INSERT ADDITIONAL SPEC FIELDS - desired state of cluster Important: Run "make" to regenerate code after modifying this file

.spec.allowlist

Type

`array`

.spec.allowlist[]

Type

object

Required

app

Property	Type	Description
app	string	
version	string	

.spec.destmatch

Description

INSERT ADDITIONAL SPEC FIELDS - desired state of cluster Important: Run "make" to regenerate code after modifying this file

Type

object

Required

service namespace

Property	Type	Description
namespace	string	
service	string	

.status

Type

object

API Endpoints

The following API endpoints are available:

- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/whitelists`
 - `DELETE` : delete collection of WhiteList
 - `GET` : list objects of kind WhiteList
 - `POST` : create a new WhiteList
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/whitelists/{name}`
 - `DELETE` : delete the specified WhiteList
 - `GET` : read the specified WhiteList
 - `PATCH` : partially update the specified WhiteList
 - `PUT` : replace the specified WhiteList
- `/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/whitelists/{name}/status`
 - `GET` : read status of the specified WhiteList
 - `PATCH` : partially update status of the specified WhiteList
 - `PUT` : replace status of the specified WhiteList

`/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/whitelists`

HTTP method

`DELETE`

Description

delete collection of WhiteList

HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code>  schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

list objects of kind WhiteList

HTTP responses

HTTP code	Response body
200 - OK	<code>WhiteListList</code>  schema
401 - Unauthorized	Empty

HTTP method

`POST`

Description

create a new WhiteList

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last

Parameter	Type	Description
		duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>WhiteList</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>WhiteList</code> schema
201 - Created	<code>WhiteList</code> schema
202 - Accepted	<code>WhiteList</code> schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/whitelists/{name}

HTTP method

DELETE

Description

delete the specified WhiteList

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code> ↗ schema
202 - Accepted	<code>Status</code> ↗ schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified WhiteList

HTTP responses

HTTP code	Response body
200 - OK	<code>WhiteList</code> ↗ schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified WhiteList

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>WhiteList</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified WhiteList

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>WhiteList</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>WhiteList</code> schema
201 - Created	<code>WhiteList</code> schema
401 - Unauthorized	Empty

/apis/asm.alauda.io/v1alpha1/namespaces/{namespace}/whitelists/{name}/status

HTTP method

`GET`

Description

read status of the specified WhiteList

HTTP responses

HTTP code	Response body
200 - OK	<code>WhiteList</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified WhiteList

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further

Parameter	Type	Description
		processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	WhiteList schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified WhiteList

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>WhiteList</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>WhiteList</code> schema
201 - Created	<code>WhiteList</code> schema

HTTP code	Response body
401 - Unauthorized	Empty