

5.1.15 - ResourceClaim v1alpha2

ResourceClaim describes which resources are needed by a resource consumer.

```
apiVersion: resource.k8s.io/v1alpha2

import "k8s.io/api/resource/v1alpha2"
```

ResourceClaim

ResourceClaim describes which resources are needed by a resource consumer. Its status tracks whether the resource has been allocated and what the resulting attributes are.

This is an alpha type and requires enabling the DynamicResourceAllocation feature gate.

- **apiVersion**: resource.k8s.io/v1alpha2

- **kind**: ResourceClaim

- **metadata** ([ObjectMeta](#))

Standard object metadata

- **spec** ([ResourceClaimSpec](#)), required

Spec describes the desired attributes of a resource that then needs to be allocated. It can only be set once when creating the ResourceClaim.

- **status** ([ResourceClaimStatus](#))

Status describes whether the resource is available and with which attributes.

ResourceClaimSpec

ResourceClaimSpec defines how a resource is to be allocated.

- **resourceClassName** (string), required

ResourceClassName references the driver and additional parameters via the name of a ResourceClass that was created as part of the driver deployment.

- **allocationMode** (string)

Allocation can start immediately or when a Pod wants to use the resource. "WaitForFirstConsumer" is the default.

- **parametersRef** ([ResourceClaimParametersReference](#))

ParametersRef references a separate object with arbitrary parameters that will be used by the driver when allocating a resource for the claim.

The object must be in the same namespace as the ResourceClaim.

ResourceClaimParametersReference contains enough information to let

you locate the parameters for a `ResourceClaim`. The object must be in the same namespace as the `ResourceClaim`.

- **parametersRef.kind** (string), required

Kind is the type of resource being referenced. This is the same value as in the parameter object's metadata, for example "ConfigMap".

- **parametersRef.name** (string), required

Name is the name of resource being referenced.

- **parametersRef.apiGroup** (string)

APIGroup is the group for the resource being referenced. It is empty for the core API. This matches the group in the APIVersion that is used when creating the resources.

ResourceClaimStatus

`ResourceClaimStatus` tracks whether the resource has been allocated and what the resulting attributes are.

- **allocation** (`AllocationResult`)

`Allocation` is set by the resource driver once a resource or set of resources has been allocated successfully. If this is not specified, the resources have not been allocated yet.

AllocationResult contains attributes of an allocated resource.

- **allocation.availableOnNodes** (`NodeSelector`)

This field will get set by the resource driver after it has allocated the resource to inform the scheduler where it can schedule Pods using the `ResourceClaim`.

Setting this field is optional. If null, the resource is available everywhere.

A node selector represents the union of the results of one or more label queries over a set of nodes; that is, it represents the OR of the selectors represented by the node selector terms.

- **allocation.availableOnNodes.nodeSelectorTerms**

(`[]NodeSelectorTerm`), required

Required. A list of node selector terms. The terms are ORed.

A null or empty node selector term matches no objects. The requirements of them are ANDed. The `TopologySelectorTerm` type implements a subset of the `NodeSelectorTerm`.

- **allocation.availableOnNodes.nodeSelectorTerms.matchExpressions**

(`[]NodeSelectorRequirement`)

A list of node selector requirements by node's labels.

- **allocation.availableOnNodes.nodeSelectorTerms.matchFields**

(`[]NodeSelectorRequirement`)

A list of node selector requirements by node's fields.

- **allocation.resourceHandles** ([]ResourceHandle)

Atomic: will be replaced during a merge

ResourceHandles contain the state associated with an allocation that should be maintained throughout the lifetime of a claim. Each ResourceHandle contains data that should be passed to a specific kubelet plugin once it lands on a node. This data is returned by the driver after a successful allocation and is opaque to Kubernetes. Driver documentation may explain to users how to interpret this data if needed.

Setting this field is optional. It has a maximum size of 32 entries. If null (or empty), it is assumed this allocation will be processed by a single kubelet plugin with no ResourceHandle data attached. The name of the kubelet plugin invoked will match the DriverName set in the ResourceClaimStatus this AllocationResult is embedded in.

ResourceHandle holds opaque resource data for processing by a specific kubelet plugin.

- **allocation.resourceHandles.data** (string)

Data contains the opaque data associated with this ResourceHandle. It is set by the controller component of the resource driver whose name matches the DriverName set in the ResourceClaimStatus this ResourceHandle is embedded in. It is set at allocation time and is intended for processing by the kubelet plugin whose name matches the DriverName set in this ResourceHandle.

The maximum size of this field is 16KiB. This may get increased in the future, but not reduced.

- **allocation.resourceHandles.driverName** (string)

DriverName specifies the name of the resource driver whose kubelet plugin should be invoked to process this ResourceHandle's data once it lands on a node. This may differ from the DriverName set in ResourceClaimStatus this ResourceHandle is embedded in.

- **allocation.shareable** (boolean)

Shareable determines whether the resource supports more than one consumer at a time.

- **deallocationRequested** (boolean)

DeallocationRequested indicates that a ResourceClaim is to be deallocated.

The driver then must deallocate this claim and reset the field together with clearing the Allocation field.

While DeallocationRequested is set, no new consumers may be added to ReservedFor.

- **driverName** (string)

DriverName is a copy of the driver name from the ResourceClass at the time when allocation started.

- **reservedFor** ([]ResourceClaimConsumerReference)

Map: unique values on key uid will be kept during a merge

ReservedFor indicates which entities are currently allowed to use the claim. A Pod which references a ResourceClaim which is not reserved for that Pod will not be started.

There can be at most 32 such reservations. This may get increased in the future, but not reduced.

ResourceClaimConsumerReference contains enough information to let you locate the consumer of a ResourceClaim. The user must be a resource in the same namespace as the ResourceClaim.

- **reservedFor.name** (string), required

Name is the name of resource being referenced.

- **reservedFor.resource** (string), required

Resource is the type of resource being referenced, for example "pods".

- **reservedFor.uid** (string), required

UID identifies exactly one incarnation of the resource.

- **reservedFor.apiGroup** (string)

APIGroup is the group for the resource being referenced. It is empty for the core API. This matches the group in the APIVersion that is used when creating the resources.

ResourceClaimList

ResourceClaimList is a collection of claims.

- **apiVersion**: resource.k8s.io/v1alpha2

- **kind**: ResourceClaimList

- **metadata** ([ListMeta](#))

Standard list metadata

- **items** ([\[\]ResourceClaim](#)), required

Items is the list of resource claims.

Operations

get read the specified ResourceClaim

HTTP Request

GET /apis/resource.k8s.io/v1alpha2/namespaces/{namespace}/resourceclaims/{name}

Parameters

- **name** (*in path*): string, required

name of the ResourceClaim

- **namespace** (*in path*): string, required

[namespace](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ResourceClaim](#)): OK

401: Unauthorized

get read status of the specified ResourceClaim

HTTP Request

GET /apis/resource.k8s.io/v1alpha2/namespaces/{namespace}/resourceclaims/{name}/status

Parameters

- **name** (*in path*): string, required

name of the ResourceClaim

- **namespace** (*in path*): string, required

[namespace](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ResourceClaim](#)): OK

401: Unauthorized

list list or watch objects of kind ResourceClaim

HTTP Request

GET /apis/resource.k8s.io/v1alpha2/namespaces/{namespace}/resourceclaims

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([ResourceClaimList](#)): OK

401: Unauthorized

list list or watch objects of kind ResourceClaim

HTTP Request

GET /apis/resource.k8s.io/v1alpha2/resourceclaims

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([ResourceClaimList](#)): OK

401: Unauthorized

create create a ResourceClaim

HTTP Request

POST /apis/resource.k8s.io/v1alpha2/namespaces/{namespace}/resourceclaims

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [ResourceClaim](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ResourceClaim](#)): OK201 ([ResourceClaim](#)): Created202 ([ResourceClaim](#)): Accepted

401: Unauthorized

update replace the specified ResourceClaim

HTTP Request

PUT /apis/resource.k8s.io/v1alpha2/namespaces/{namespace}/resourceclaims/{name}

Parameters

- **name** (*in path*): string, required
name of the ResourceClaim
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [ResourceClaim](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([ResourceClaim](#)): OK

201 ([ResourceClaim](#)): Created

401: Unauthorized

update replace status of the specified ResourceClaim

HTTP Request

PUT /apis/resource.k8s.io/v1alpha2/namespaces/{namespace}/resourceclaims/{name}/status

Parameters

- **name** (*in path*): string, required
name of the ResourceClaim
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [ResourceClaim](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ResourceClaim](#)): OK

201 ([ResourceClaim](#)): Created

401: Unauthorized

patch partially update the specified ResourceClaim

HTTP Request

PATCH /apis/resource.k8s.io/v1alpha2/namespaces/{namespace}/resourceclaims/{name}

Parameters

- **name** (*in path*): string, required

name of the ResourceClaim

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ResourceClaim](#)): OK

201 ([ResourceClaim](#)): Created

401: Unauthorized

patch partially update status of the specified ResourceClaim

HTTP Request

PATCH /apis/resource.k8s.io/v1alpha2/namespaces/{namespace}/resourceclaims/{name}/status

Parameters

- **name** (*in path*): string, required
name of the ResourceClaim
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [Patch](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **force** (*in query*): boolean
[force](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([ResourceClaim](#)): OK

201 ([ResourceClaim](#)): Created

401: Unauthorized

delete delete a ResourceClaim

HTTP Request

DELETE /apis/resource.k8s.io/v1alpha2/namespaces/{namespace}/resourceclaims/{name}

Parameters

- **name** (*in path*): string, required
name of the ResourceClaim
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [DeleteOptions](#)
- **dryRun** (*in query*): string
[dryRun](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

Response

200 ([ResourceClaim](#)): OK

202 ([ResourceClaim](#)): Accepted

401: Unauthorized

deletecollection delete collection of ResourceClaim

HTTP Request

DELETE /apis/resource.k8s.io/v1alpha2/namespaces/{namespace}/resourceclaims

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [DeleteOptions](#)

- **continue** (*in query*): string

[continue](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.1.16 - ResourceClaimTemplate v1alpha2

ResourceClaimTemplate is used to produce ResourceClaim objects.

```
apiVersion: resource.k8s.io/v1alpha2  
import "k8s.io/api/resource/v1alpha2"
```

ResourceClaimTemplate

ResourceClaimTemplate is used to produce ResourceClaim objects.

- **apiVersion**: resource.k8s.io/v1alpha2

- **kind**: ResourceClaimTemplate

- **metadata** ([ObjectMeta](#))

Standard object metadata

- **spec** ([ResourceClaimTemplateSpec](#)), required

Describes the ResourceClaim that is to be generated.

This field is immutable. A ResourceClaim will get created by the control plane for a Pod when needed and then not get updated anymore.

ResourceClaimTemplateSpec

ResourceClaimTemplateSpec contains the metadata and fields for a ResourceClaim.

- **spec** ([ResourceClaimSpec](#)), required

Spec for the ResourceClaim. The entire content is copied unchanged into the ResourceClaim that gets created from this template. The same fields as in a ResourceClaim are also valid here.

- **metadata** ([ObjectMeta](#))

ObjectMeta may contain labels and annotations that will be copied into the PVC when creating it. No other fields are allowed and will be rejected during validation.

ResourceClaimTemplateList

ResourceClaimTemplateList is a collection of claim templates.

- **apiVersion**: resource.k8s.io/v1alpha2

- **kind**: ResourceClaimTemplateList

- **metadata** ([ListMeta](#))

Standard list metadata

- **items** ([][ResourceClaimTemplate](#)), required

Items is the list of resource claim templates.

Operations

get read the specified ResourceClaimTemplate

HTTP Request

```
GET /apis/resource.k8s.io/v1alpha2/namespaces/{namespace}/  
resourceclaimtemplates/{name}
```

Parameters

- **name** (*in path*): string, required
name of the ResourceClaimTemplate
- **namespace** (*in path*): string, required
[namespace](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([ResourceClaimTemplate](#)): OK

401: Unauthorized

list list or watch objects of kind ResourceClaimTemplate

HTTP Request

```
GET /apis/resource.k8s.io/v1alpha2/namespaces/{namespace}/  
resourceclaimtemplates
```

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string
[pretty](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)
- **watch** (*in query*): boolean
[watch](#)

Response

200 ([ResourceClaimTemplateList](#)): OK

401: Unauthorized

list list or watch objects of kind ResourceClaimTemplate

HTTP Request

GET /apis/resource.k8s.io/v1alpha2/resourceclaimtemplates

Parameters

- **allowWatchBookmarks** (*in query*): boolean
[allowWatchBookmarks](#)
- **continue** (*in query*): string
[continue](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([ResourceClaimTemplateList](#)): OK

401: Unauthorized

create create a ResourceClaimTemplate

HTTP Request

POST /apis/resource.k8s.io/v1alpha2/namespaces/{namespace}/resourceclaimtemplates

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [ResourceClaimTemplate](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ResourceClaimTemplate](#)): OK

201 ([ResourceClaimTemplate](#)): Created

202 ([ResourceClaimTemplate](#)): Accepted

401: Unauthorized

update replace the specified ResourceClaimTemplate

HTTP Request

PUT /apis/resource.k8s.io/v1alpha2/namespaces/{namespace}/

resourceclaimtemplates/{name}

Parameters

- **name** (*in path*): string, required
name of the ResourceClaimTemplate
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [ResourceClaimTemplate](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([ResourceClaimTemplate](#)): OK

201 ([ResourceClaimTemplate](#)): Created

401: Unauthorized

patch partially update the specified ResourceClaimTemplate

HTTP Request

PATCH /apis/resource.k8s.io/v1alpha2/namespaces/{namespace}/resourceclaimtemplates/{name}

Parameters

- **name** (*in path*): string, required
name of the ResourceClaimTemplate
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [Patch](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ResourceClaimTemplate](#)): OK201 ([ResourceClaimTemplate](#)): Created

401: Unauthorized

delete delete a ResourceClaimTemplate

HTTP Request

DELETE /apis/resource.k8s.io/v1alpha2/namespaces/{namespace}/resourceclaimtemplates/{name}

Parameters

- **name** (*in path*): string, required

name of the ResourceClaimTemplate

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [DeleteOptions](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

Response

200 ([ResourceClaimTemplate](#)): OK202 ([ResourceClaimTemplate](#)): Accepted

401: Unauthorized

deletecollection delete collection of ResourceClaimTemplate

HTTP Request

DELETE /apis/resource.k8s.io/v1alpha2/namespaces/{namespace}/resourceclaimtemplates

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [DeleteOptions](#)

- **continue** (*in query*): string

[continue](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.1.17 - ResourceClass v1alpha2

ResourceClass is used by administrators to influence how resources are allocated.

```
apiVersion: resource.k8s.io/v1alpha2

import "k8s.io/api/resource/v1alpha2"
```

ResourceClass

ResourceClass is used by administrators to influence how resources are allocated.

This is an alpha type and requires enabling the DynamicResourceAllocation feature gate.

- **apiVersion**: resource.k8s.io/v1alpha2

- **kind**: ResourceClass

- **metadata** ([ObjectMeta](#))

Standard object metadata

- **driverName** (string), required

DriverName defines the name of the dynamic resource driver that is used for allocation of a ResourceClaim that uses this class.

Resource drivers have a unique name in forward domain order (acme.example.com).

- **parametersRef** (ResourceClassParametersReference)

ParametersRef references an arbitrary separate object that may hold parameters that will be used by the driver when allocating a resource that uses this class. A dynamic resource driver can distinguish between parameters stored here and those stored in ResourceClaimSpec.

ResourceClassParametersReference contains enough information to let you locate the parameters for a ResourceClass.

- **parametersRef.kind** (string), required

Kind is the type of resource being referenced. This is the same value as in the parameter object's metadata.

- **parametersRef.name** (string), required

Name is the name of resource being referenced.

- **parametersRef.apiGroup** (string)

APIGroup is the group for the resource being referenced. It is empty for the core API. This matches the group in the APIVersion that is used when creating the resources.

- **parametersRef.namespace** (string)

Namespace that contains the referenced resource. Must be empty for cluster-scoped resources and non-empty for namespaced resources.

- **suitableNodes** (NodeSelector)

Only nodes matching the selector will be considered by the scheduler when trying to find a Node that fits a Pod when that Pod uses a ResourceClaim that has not been allocated yet.

Setting this field is optional. If null, all nodes are candidates.

A node selector represents the union of the results of one or more label queries over a set of nodes; that is, it represents the OR of the selectors represented by the node selector terms.

- **suitableNodes.nodeSelectorTerms** ([]NodeSelectorTerm), required

Required. A list of node selector terms. The terms are ORed.

A null or empty node selector term matches no objects. The requirements of them are ANDed. The TopologySelectorTerm type implements a subset of the NodeSelectorTerm.

- **suitableNodes.nodeSelectorTerms.matchExpressions** ([]NodeSelectorRequirement)

A list of node selector requirements by node's labels.

- **suitableNodes.nodeSelectorTerms.matchFields** ([]NodeSelectorRequirement)

A list of node selector requirements by node's fields.

ResourceclassList

ResourceclassList is a collection of classes.

- **apiVersion**: resource.k8s.io/v1alpha2

- **kind**: ResourceclassList

- **metadata** ([ListMeta](#))

Standard list metadata

- **items** ([]ResourceClass), required

Items is the list of resource classes.

Operations

get read the specified ResourceClass

HTTP Request

GET /apis/resource.k8s.io/v1alpha2/resourceclasses/{name}

Parameters

- **name** (*in path*): string, required

name of the ResourceClass

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ResourceClass](#)): OK

401: Unauthorized

list list or watch objects of kind ResourceClass

HTTP Request

GET /apis/resource.k8s.io/v1alpha2/resourceclasses

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([ResourceClassList](#)): OK

401: Unauthorized

create create a ResourceClass

HTTP Request

POST /apis/resource.k8s.io/v1alpha2/resourceclasses

Parameters

- **body**: [ResourceClass](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([ResourceClass](#)): OK

201 ([ResourceClass](#)): Created

202 ([ResourceClass](#)): Accepted

401: Unauthorized

update replace the specified ResourceClass

HTTP Request

PUT /apis/resource.k8s.io/v1alpha2/resourceclasses/{name}

Parameters

- **name** (*in path*): string, required
name of the ResourceClass
- **body**: [ResourceClass](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([ResourceClass](#)): OK

201 ([ResourceClass](#)): Created

401: Unauthorized

patch partially update the specified ResourceClass

HTTP Request

PATCH /apis/resource.k8s.io/v1alpha2/resourceclasses/{name}

Parameters

- **name** (*in path*): string, required

name of the ResourceClass

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ResourceClass](#)): OK

201 ([ResourceClass](#)): Created

401: Unauthorized

delete delete a ResourceClass

HTTP Request

DELETE /apis/resource.k8s.io/v1alpha2/resourceclasses/{name}

Parameters

- **name** (*in path*): string, required

name of the ResourceClass

- **body**: [DeleteOptions](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)

Response

200 ([ResourceClass](#)): OK

202 ([ResourceClass](#)): Accepted

401: Unauthorized

deletecollection delete collection of ResourceClass

HTTP Request

DELETE /apis/resource.k8s.io/v1alpha2/resourceclasses

Parameters

- **body**: [DeleteOptions](#)
- **continue** (*in query*): string
[continue](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.2 - Service Resources

5.2.1 - Service

Service is a named abstraction of software service (for example, mysql) consisting of local port (for example 3306) that the proxy listens on, and the selector that determines which pods will answer requests sent through the proxy.

```
apiVersion: v1

import "k8s.io/api/core/v1"
```

Service

Service is a named abstraction of software service (for example, mysql) consisting of local port (for example 3306) that the proxy listens on, and the selector that determines which pods will answer requests sent through the proxy.

- **apiVersion**: v1
- **kind**: Service
- **metadata** ([ObjectMeta](#))

Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **spec** ([ServiceSpec](#))

Spec defines the behavior of a service. <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status>

- **status** ([ServiceStatus](#))

Most recently observed status of the service. Populated by the system. Read-only. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status>

ServiceSpec

ServiceSpec describes the attributes that a user creates on a service.

- **selector** (map[string]string)

Route service traffic to pods with label keys and values matching this selector. If empty or not present, the service is assumed to have an external process managing its endpoints, which Kubernetes will not modify. Only applies to types ClusterIP, NodePort, and LoadBalancer. Ignored if type is ExternalName. More info: <https://kubernetes.io/docs/concepts/services-networking/service/>

- **ports** ([]ServicePort)

Patch strategy: merge on key port

Map: unique values on keys port, protocol will be kept during a merge

The list of ports that are exposed by this service. More info: <https://kubernetes.io/docs/concepts/services-networking/service/#virtual-ips-and-service-proxies>

ServicePort contains information on service's port.

- **ports.port** (int32), required

The port that will be exposed by this service.

- **ports.targetPort** (IntOrString)

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>

IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

- **ports.protocol** (string)

The IP protocol for this port. Supports "TCP", "UDP", and "SCTP". Default is TCP.

- **ports.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.

- **ports.nodePort** (int32)

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>

- **ports.appProtocol** (string)

The application protocol for this port. This is used as a hint for implementations to offer richer behavior for protocols that they understand. This field follows standard Kubernetes label syntax. Valid values are either:

- Un-prefixed protocol names - reserved for IANA standard service names (as per RFC-6335 and <https://www.iana.org/assignments/service-names>).

- Kubernetes-defined prefixed names:
 - 'kubernetes.io/h2c' - HTTP/2 over cleartext as described in <https://www.rfc-editor.org/rfc/rfc7540>
 - 'kubernetes.io/ws' - WebSocket over cleartext as described in <https://www.rfc-editor.org/rfc/rfc6455>
 - 'kubernetes.io/wss' - WebSocket over TLS as described in <https://www.rfc-editor.org/rfc/rfc6455>
- Other protocols should use implementation-defined prefixed names such as mycompany.com/my-custom-protocol.

- **type** (string)

type determines how the Service is exposed. Defaults to ClusterIP. Valid options are ExternalName, ClusterIP, NodePort, and LoadBalancer. "ClusterIP" allocates a cluster-internal IP address for load-balancing to endpoints. Endpoints are determined by the selector or if that is not specified, by manual construction of an Endpoints object or EndpointSlice objects. If clusterIP is "None", no virtual IP is allocated and the endpoints are published as a set of endpoints rather than a virtual IP. "NodePort" builds on ClusterIP and allocates a port on every node which routes to the same endpoints as the clusterIP. "LoadBalancer" builds on NodePort and creates an external load-balancer (if supported in the current cloud) which routes to the same endpoints as the clusterIP. "ExternalName" aliases this service to the specified externalName. Several other fields do not apply to ExternalName services. More info: <https://kubernetes.io/docs/concepts/services-networking/service/#publishing-services-service-types>

- **ipFamilies** ([]string)

Atomic: will be replaced during a merge

IPFamilies is a list of IP families (e.g. IPv4, IPv6) assigned to this service. This field is usually assigned automatically based on cluster configuration and the ipFamilyPolicy field. If this field is specified manually, the requested family is available in the cluster, and ipFamilyPolicy allows it, it will be used; otherwise creation of the service will fail. This field is conditionally mutable: it allows for adding or removing a secondary IP family, but it does not allow changing the primary IP family of the Service. Valid values are "IPv4" and "IPv6". This field only applies to Services of types ClusterIP, NodePort, and LoadBalancer, and does not apply to "headless" services. This field will be wiped when updating a Service to type ExternalName.

This field may hold a maximum of two entries (dual-stack families, in either order). These families must correspond to the values of the clusterIPs field, if specified. Both clusterIPs and ipFamilies are governed by the ipFamilyPolicy field.

- **ipFamilyPolicy** (string)

IPFamilyPolicy represents the dual-stack-ness requested or required by this Service. If there is no value provided, then this field will be set to SingleStack. Services can be "SingleStack" (a single IP family), "PreferDualStack" (two IP families on dual-stack configured clusters or a single IP family on single-stack clusters), or "RequireDualStack" (two IP families on dual-stack configured clusters, otherwise fail). The ipFamilies and clusterIPs fields depend on the value of this field. This field will be wiped when updating a service to type ExternalName.

- **clusterIP** (string)

clusterIP is the IP address of the service and is usually assigned randomly. If an address is specified manually, is in-range (as per system configuration), and is not in use, it will be allocated to the service; otherwise creation of the service will fail. This field may not be changed through updates unless the type field is also being changed to ExternalName (which requires this field to be blank) or the type field is being changed from ExternalName (in which case this field may optionally be specified, as describe above). Valid values are "None", empty string (""), or a valid IP address. Setting this to "None" makes a "headless service" (no virtual IP), which is useful when direct endpoint connections are preferred and proxying is not required. Only applies to types ClusterIP, NodePort, and LoadBalancer. If this field is specified when creating a Service of type ExternalName, creation will fail. This field will be wiped when updating a Service to type ExternalName. More info: <https://kubernetes.io/docs/concepts/services-networking/service/#virtual-ips-and-service-proxies>

- **clusterIPs** ([]string)

Atomic: will be replaced during a merge

ClusterIPs is a list of IP addresses assigned to this service, and are usually assigned randomly. If an address is specified manually, is in-range (as per system configuration), and is not in use, it will be allocated to the service; otherwise creation of the service will fail. This field may not be changed through updates unless the type field is also being changed to ExternalName (which requires this field to be empty) or the type field is being changed from ExternalName (in which case this field may optionally be specified, as describe above). Valid values are "None", empty string (""), or a valid IP address. Setting this to "None" makes a "headless service" (no virtual IP), which is useful when direct endpoint connections are preferred and proxying is not required. Only applies to types ClusterIP, NodePort, and LoadBalancer. If this field is specified when creating a Service of type ExternalName, creation will fail. This field will be wiped when updating a Service to type ExternalName. If this field is not specified, it will be initialized from the clusterIP field. If this field is specified, clients must ensure that clusterIPs[0] and clusterIP have the same value.

This field may hold a maximum of two entries (dual-stack IPs, in either order). These IPs must correspond to the values of the ipFamilies field. Both clusterIPs and ipFamilies are governed by the ipFamilyPolicy field. More info: <https://kubernetes.io/docs/concepts/services-networking/service/#virtual-ips-and-service-proxies>

- **externalIPs** ([]string)

externalIPs is a list of IP addresses for which nodes in the cluster will also accept traffic for this service. These IPs are not managed by Kubernetes. The user is responsible for ensuring that traffic arrives at a node with this IP. A common example is external load-balancers that are not part of the Kubernetes system.

- **sessionAffinity** (string)

Supports "ClientIP" and "None". Used to maintain session affinity. Enable client IP based session affinity. Must be ClientIP or None. Defaults to None. More info: <https://kubernetes.io/docs/concepts/services-networking/service/#virtual-ips-and-service-proxies>

- **loadBalancerIP** (string)

Only applies to Service Type: LoadBalancer. This feature depends on whether the underlying cloud-provider supports specifying the loadBalancerIP when a load balancer is created. This field will be ignored if the cloud-provider does not support the feature.

Deprecated: This field was under-specified and its meaning varies across implementations. Using it is non-portable and it may not support dual-stack. Users are encouraged to use implementation-specific annotations when available.

- **loadBalancerSourceRanges** ([]string)

If specified and supported by the platform, this will restrict traffic through the cloud-provider load-balancer will be restricted to the specified client IPs. This field will be ignored if the cloud-provider does not support the feature." More info: <https://kubernetes.io/docs/tasks/access-application-cluster/create-external-load-balancer/>

- **loadBalancerClass** (string)

loadBalancerClass is the class of the load balancer implementation this Service belongs to. If specified, the value of this field must be a label-style identifier, with an optional prefix, e.g. "internal-vip" or "example.com/internal-vip". Unprefixed names are reserved for end-users. This field can only be set when the Service type is 'LoadBalancer'. If not set, the default load balancer implementation is used, today this is typically done through the cloud provider integration, but should apply for any default implementation. If set, it is assumed that a load balancer implementation is watching for Services with a matching class. Any default load balancer implementation (e.g. cloud providers) should ignore Services that set this field. This field can only be set when creating or updating a Service to type 'LoadBalancer'. Once set, it can not be changed. This field will be wiped when a service is updated to a non 'LoadBalancer' type.

- **externalName** (string)

externalName is the external reference that discovery mechanisms will return as an alias for this service (e.g. a DNS CNAME record). No proxying will be involved. Must be a lowercase RFC-1123 hostname (<https://tools.ietf.org/html/rfc1123>) and requires type to be "ExternalName".

- **externalTrafficPolicy** (string)

externalTrafficPolicy describes how nodes distribute service traffic they receive on one of the Service's "externally-facing" addresses (NodePorts, ExternalIPs, and LoadBalancer IPs). If set to "Local", the proxy will configure the service in a way that assumes that external load balancers will take care of balancing the service traffic between nodes, and so each node will deliver traffic only to the node-local endpoints of the service, without masquerading the client source IP. (Traffic mistakenly sent to a node with no endpoints will be dropped.) The default value, "Cluster", uses the standard behavior of routing to all endpoints evenly (possibly modified by topology and other features). Note that traffic sent to an External IP or LoadBalancer IP from within the cluster will always get "Cluster" semantics, but clients sending to a NodePort from within the cluster may need to take traffic policy into account when picking a node.

- **internalTrafficPolicy** (string)

InternalTrafficPolicy describes how nodes distribute service traffic they receive on the ClusterIP. If set to "Local", the proxy will assume

that pods only want to talk to endpoints of the service on the same node as the pod, dropping the traffic if there are no local endpoints. The default value, "Cluster", uses the standard behavior of routing to all endpoints evenly (possibly modified by topology and other features).

- **healthCheckNodePort** (int32)

healthCheckNodePort specifies the healthcheck nodePort for the service. This only applies when type is set to LoadBalancer and externalTrafficPolicy is set to Local. If a value is specified, is in-range, and is not in use, it will be used. If not specified, a value will be automatically allocated. External systems (e.g. load-balancers) can use this port to determine if a given node holds endpoints for this service or not. 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). This field cannot be updated once set.

- **publishNotReadyAddresses** (boolean)

publishNotReadyAddresses indicates that any agent which deals with endpoints for this Service should disregard any indications of ready/not-ready. The primary use case for setting this field is for a StatefulSet's Headless Service to propagate SRV DNS records for its Pods for the purpose of peer discovery. The Kubernetes controllers that generate Endpoints and EndpointSlice resources for Services interpret this to mean that all endpoints are considered "ready" even if the Pods themselves are not. Agents which consume only Kubernetes generated endpoints through the Endpoints or EndpointSlice resources can safely assume this behavior.

- **sessionAffinityConfig** (SessionAffinityConfig)

sessionAffinityConfig contains the configurations of session affinity.

SessionAffinityConfig represents the configurations of session affinity.

- **sessionAffinityConfig.clientIP** (ClientIPConfig)

clientIP contains the configurations of Client IP based session affinity.

ClientIPConfig represents the configurations of Client IP based session affinity.

- **sessionAffinityConfig.clientIP.timeoutSeconds** (int32)

timeoutSeconds specifies the seconds of ClientIP type session sticky time. The value must be $>0 \ \&\amp;$ ≤ 86400 (for 1 day) if ServiceAffinity == "ClientIP". Default value is 10800 (for 3 hours).

- **allocateLoadBalancerNodePorts** (boolean)

allocateLoadBalancerNodePorts defines if NodePorts will be automatically allocated for services with type LoadBalancer. Default is "true". It may be set to "false" if the cluster load-balancer does not rely on NodePorts. If the caller requests specific NodePorts (by specifying a value), those requests will be respected, regardless of this field. This field may only be set for services with type LoadBalancer and will be cleared if the type is changed to any other type.

ServiceStatus

ServiceStatus represents the current status of a service.

- **conditions** ([]Condition)

Patch strategy: merge on key type

Map: unique values on key type will be kept during a merge

Current service state

Condition contains details for one aspect of the current state of this API Resource.

- **conditions.lastTransitionTime** (Time), required

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.

Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

- **conditions.message** (string), required

message is a human readable message indicating details about the transition. This may be an empty string.

- **conditions.reason** (string), required

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.

- **conditions.status** (string), required

status of the condition, one of True, False, Unknown.

- **conditions.type** (string), required

type of condition in CamelCase or in foo.example.com/CamelCase.

- **conditions.observedGeneration** (int64)

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.

- **loadBalancer** (LoadBalancerStatus)

LoadBalancer contains the current status of the load-balancer, if one is present.

LoadBalancerStatus represents the status of a load-balancer.

- **loadBalancer.ingress** ([]LoadBalancerIngress)

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

LoadBalancerIngress represents the status of a load-balancer

ingress point: traffic intended for the service should be sent to an ingress point.

- **loadBalancer.ingress.hostname** (string)

Hostname is set for load-balancer ingress points that are DNS based (typically AWS load-balancers)

- **loadBalancer.ingress.ip** (string)

IP is set for load-balancer ingress points that are IP based (typically GCE or OpenStack load-balancers)

- **loadBalancer.ingress.ipMode** (string)

IPMode specifies how the load-balancer IP behaves, and may only be specified when the ip field is specified.

Setting this to "VIP" indicates that traffic is delivered to the node with the destination set to the load-balancer's IP and port. Setting this to "Proxy" indicates that traffic is delivered to the node or pod with the destination set to the node's IP and node port or the pod's IP and port. Service implementations may use this information to adjust traffic routing.

- **loadBalancer.ingress.ports** ([]PortStatus)

Atomic: will be replaced during a merge

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

**

- **loadBalancer.ingress.ports.port** (int32), required

Port is the port number of the service port of which status is recorded here

- **loadBalancer.ingress.ports.protocol** (string), required

Protocol is the protocol of the service port of which status is recorded here The supported values are: "TCP", "UDP", "SCTP"

- **loadBalancer.ingress.ports.error** (string)

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.

ServiceList

ServiceList holds a list of services.

- **apiVersion**: v1

- **kind**: ServiceList

- **metadata** ([ListMeta](#))

Standard list metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds>

- **items** ([\[\]Service](#)), required

List of services

Operations

get read the specified Service

HTTP Request

GET /api/v1/namespaces/{namespace}/services/{name}

Parameters

- **name** (*in path*): string, required

name of the Service

- **namespace** (*in path*): string, required

[namespace](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Service](#)): OK

401: Unauthorized

get read status of the specified Service

HTTP Request

GET /api/v1/namespaces/{namespace}/services/{name}/status

Parameters

- **name** (*in path*): string, required

name of the Service

- **namespace** (*in path*): string, required

[namespace](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Service](#)): OK

401: Unauthorized

list list or watch objects of kind Service

HTTP Request

GET /api/v1/namespaces/{namespace}/services

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([ServiceList](#)): OK

401: Unauthorized

list list or watch objects of kind Service

HTTP Request

GET /api/v1/services

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([ServiceList](#)): OK

401: Unauthorized

create create a Service

HTTP Request

POST /api/v1/namespaces/{namespace}/services

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [Service](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Service](#)): OK

201 ([Service](#)): Created

202 ([Service](#)): Accepted

401: Unauthorized

update replace the specified Service

HTTP Request

PUT /api/v1/namespaces/{namespace}/services/{name}

Parameters

- **name** (*in path*): string, required

name of the Service

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [Service](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Service](#)): OK

201 ([Service](#)): Created

401: Unauthorized

update replace status of the specified Service

HTTP Request

PUT /api/v1/namespaces/{namespace}/services/{name}/status

Parameters

- **name** (*in path*): string, required
name of the Service
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [Service](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([Service](#)): OK

201 ([Service](#)): Created

401: Unauthorized

patch partially update the specified Service

HTTP Request

PATCH /api/v1/namespaces/{namespace}/services/{name}

Parameters

- **name** (*in path*): string, required
name of the Service
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [Patch](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Service](#)): OK201 ([Service](#)): Created

401: Unauthorized

patch partially update status of the specified Service

HTTP Request

PATCH /api/v1/namespaces/{namespace}/services/{name}/status

Parameters

- **name** (*in path*): string, required

name of the Service

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Service](#)): OK201 ([Service](#)): Created

401: Unauthorized

delete delete a Service

HTTP Request

DELETE /api/v1/namespaces/{namespace}/services/{name}

Parameters

- **name** (*in path*): string, required
[name](#)
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [DeleteOptions](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)

Response

200 ([Service](#)): OK

202 ([Service](#)): Accepted

401: Unauthorized

deletecollection delete collection of Service

HTTP Request

DELETE /api/v1/namespaces/{namespace}/services

Parameters

- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [DeleteOptions](#)
- **continue** (*in query*): string
[continue](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.2.2 - Endpoints

Endpoints is a collection of endpoints that implement the actual service.

```
apiVersion: v1

import "k8s.io/api/core/v1"
```

Endpoints

Endpoints is a collection of endpoints that implement the actual service.

Example:

```
Name: "mysvc",
Subsets: [
  {
    Addresses: [{"ip": "10.10.1.1"}, {"ip": "10.10.2.2"}],
    Ports: [{"name": "a", "port": 8675}, {"name": "b", "port": 309}]
  },
  {
    Addresses: [{"ip": "10.10.3.3"}],
    Ports: [{"name": "a", "port": 93}, {"name": "b", "port": 76}]
  },
]
```

- **apiVersion:** v1
- **kind:** Endpoints
- **metadata** ([ObjectMeta](#))

Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **subsets** ([]EndpointSubset)

The set of all endpoints is the union of all subsets. Addresses are placed into subsets according to the IPs they share. A single address with multiple ports, some of which are ready and some of which are not (because they come from different containers) will result in the address being displayed in different subsets for the different ports. No address will appear in both Addresses and NotReadyAddresses in the same subset. Sets of addresses and ports that comprise a service.

*EndpointSubset is a group of addresses with a common set of ports. The expanded set of endpoints is the Cartesian product of Addresses x Ports. For example, given:

```
{ Addresses: [{"ip": "10.10.1.1"}, {"ip": "10.10.2.2"}], Ports: [{"name": "a", "port": 8675}, {"name": "b", "port": 309}] }
```

The resulting set of endpoints can be viewed as:

```
a: [ 10.10.1.1:8675, 10.10.2.2:8675 ], b: [ 10.10.1.1:309, 10.10.2.2:309 ]*
```

- **subsets.addresses** ([]EndpointAddress)

IP addresses which offer the related ports that are marked as ready. These endpoints should be considered safe for load

balancers and clients to utilize.

EndpointAddress is a tuple that describes single IP address.

- **subsets.addresses.ip** (string), required

The IP of this endpoint. May not be loopback (127.0.0.0/8 or ::1), link-local (169.254.0.0/16 or fe80::/10), or link-local multicast (224.0.0.0/24 or ff02::/16).

- **subsets.addresses.hostname** (string)

The Hostname of this endpoint

- **subsets.addresses.nodeName** (string)

Optional: Node hosting this endpoint. This can be used to determine endpoints local to a node.

- **subsets.addresses.targetRef** ([ObjectReference](#))

Reference to object providing the endpoint.

- **subsets.notReadyAddresses** ([]EndpointAddress)

IP addresses which offer the related ports but are not currently marked as ready because they have not yet finished starting, have recently failed a readiness check, or have recently failed a liveness check.

EndpointAddress is a tuple that describes single IP address.

- **subsets.notReadyAddresses.ip** (string), required

The IP of this endpoint. May not be loopback (127.0.0.0/8 or ::1), link-local (169.254.0.0/16 or fe80::/10), or link-local multicast (224.0.0.0/24 or ff02::/16).

- **subsets.notReadyAddresses.hostname** (string)

The Hostname of this endpoint

- **subsets.notReadyAddresses.nodeName** (string)

Optional: Node hosting this endpoint. This can be used to determine endpoints local to a node.

- **subsets.notReadyAddresses.targetRef** ([ObjectReference](#))

Reference to object providing the endpoint.

- **subsets.ports** ([]EndpointPort)

Port numbers available on the related IP addresses.

EndpointPort is a tuple that describes a single port.

- **subsets.ports.port** (int32), required

The port number of the endpoint.

- **subsets.ports.protocol** (string)

The IP protocol for this port. Must be UDP, TCP, or SCTP.
Default is TCP.

- **subsets.ports.name** (string)

The name of this port. This must match the 'name' field in the corresponding ServicePort. Must be a DNS_LABEL.

Optional only if one port is defined.

- **subsets.ports.appProtocol** (string)

The application protocol for this port. This is used as a hint for implementations to offer richer behavior for protocols that they understand. This field follows standard Kubernetes label syntax. Valid values are either:

- Un-prefixed protocol names - reserved for IANA standard service names (as per RFC-6335 and <https://www.iana.org/assignments/service-names>).
- Kubernetes-defined prefixed names:
 - 'kubernetes.io/h2c' - HTTP/2 over cleartext as described in <https://www.rfc-editor.org/rfc/rfc7540>
 - 'kubernetes.io/ws' - WebSocket over cleartext as described in <https://www.rfc-editor.org/rfc/rfc6455>
 - 'kubernetes.io/wss' - WebSocket over TLS as described in <https://www.rfc-editor.org/rfc/rfc6455>
- Other protocols should use implementation-defined prefixed names such as mycompany.com/my-custom-protocol.

EndpointsList

EndpointsList is a list of endpoints.

- **apiVersion**: v1

- **kind**: EndpointsList

- **metadata** ([ListMeta](#))

Standard list metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds>

- **items** ([] [Endpoints](#)), required

List of endpoints.

Operations

get read the specified Endpoints

HTTP Request

GET /api/v1/namespaces/{namespace}/endpoints/{name}

Parameters

- **name** (*in path*): string, required

name of the Endpoints

- **namespace** (*in path*): string, required

[namespace](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Endpoints](#)): OK

401: Unauthorized

list list or watch objects of kind Endpoints

HTTP Request

GET /api/v1/namespaces/{namespace}/endpoints

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([EndpointsList](#)): OK

401: Unauthorized

list list or watch objects of kind Endpoints

HTTP Request

GET /api/v1/endpoints

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([EndpointsList](#)): OK

401: Unauthorized

create create Endpoints

HTTP Request

POST /api/v1/namespaces/{namespace}/endpoints

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [Endpoints](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Endpoints](#)): OK

201 ([Endpoints](#)): Created

202 ([Endpoints](#)): Accepted

401: Unauthorized

update replace the specified Endpoints

HTTP Request

PUT /api/v1/namespaces/{namespace}/endpoints/{name}

Parameters

- **name** (*in path*): string, required

name of the Endpoints

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [Endpoints](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Endpoints](#)): OK

201 ([Endpoints](#)): Created

401: Unauthorized

patch partially update the specified Endpoints

HTTP Request

PATCH /api/v1/namespaces/{namespace}/endpoints/{name}

Parameters

- **name** (*in path*): string, required

name of the Endpoints

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Endpoints](#)): OK

201 ([Endpoints](#)): Created

401: Unauthorized

delete delete Endpoints

HTTP Request

DELETE /api/v1/namespaces/{namespace}/endpoints/{name}

Parameters

- **name** (*in path*): string, required

name of the Endpoints

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [DeleteOptions](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

Response

200 ([Status](#)): OK

202 ([Status](#)): Accepted

401: Unauthorized

deletecollection delete collection of Endpoints

HTTP Request

DELETE /api/v1/namespaces/{namespace}/endpoints

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [DeleteOptions](#)

- **continue** (*in query*): string

[continue](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.2.3 - EndpointSlice

EndpointSlice represents a subset of the endpoints that implement a service.

```
apiVersion: discovery.k8s.io/v1
import "k8s.io/api/discovery/v1"
```

EndpointSlice

EndpointSlice represents a subset of the endpoints that implement a service. For a given service there may be multiple EndpointSlice objects, selected by labels, which must be joined to produce the full set of endpoints.

- **apiVersion**: discovery.k8s.io/v1

- **kind**: EndpointSlice

- **metadata** ([ObjectMeta](#))

Standard object's metadata.

- **addressType** (string), required

addressType specifies the type of address carried by this EndpointSlice. All addresses in this slice must be the same type. This field is immutable after creation. The following address types are currently supported: * IPv4: Represents an IPv4 Address. * IPv6: Represents an IPv6 Address. * FQDN: Represents a Fully Qualified Domain Name.

- **endpoints** ([]Endpoint), required

Atomic: will be replaced during a merge

endpoints is a list of unique endpoints in this slice. Each slice may include a maximum of 1000 endpoints.

Endpoint represents a single logical "backend" implementing a service.

- **endpoints.addresses** ([]string), required

Set: unique values will be kept during a merge

addresses of this endpoint. The contents of this field are interpreted according to the corresponding EndpointSlice addressType field. Consumers must handle different types of addresses in the context of their own capabilities. This must contain at least one address but no more than 100. These are all assumed to be fungible and clients may choose to only use the first element. Refer to: <https://issue.k8s.io/106267>

- **endpoints.conditions** (EndpointConditions)

conditions contains information about the current status of the endpoint.

EndpointConditions represents the current condition of an endpoint.

- **endpoints.conditions.ready** (boolean)

ready indicates that this endpoint is prepared to receive

traffic, according to whatever system is managing the endpoint. A nil value indicates an unknown state. In most cases consumers should interpret this unknown state as ready. For compatibility reasons, ready should never be "true" for terminating endpoints, except when the normal readiness behavior is being explicitly overridden, for example when the associated Service has set the publishNotReadyAddresses flag.

- **endpoints.conditions.serving** (boolean)

serving is identical to ready except that it is set regardless of the terminating state of endpoints. This condition should be set to true for a ready endpoint that is terminating. If nil, consumers should defer to the ready condition.

- **endpoints.conditions.terminating** (boolean)

terminating indicates that this endpoint is terminating. A nil value indicates an unknown state. Consumers should interpret this unknown state to mean that the endpoint is not terminating.

- **endpoints.deprecatedTopology** (map[string]string)

deprecatedTopology contains topology information part of the v1beta1 API. This field is deprecated, and will be removed when the v1beta1 API is removed (no sooner than kubernetes v1.24). While this field can hold values, it is not writable through the v1 API, and any attempts to write to it will be silently ignored. Topology information can be found in the zone and nodeName fields instead.

- **endpoints.hints** (EndpointHints)

hints contains information associated with how an endpoint should be consumed.

EndpointHints provides hints describing how an endpoint should be consumed.

- **endpoints.hints.forZones** ([]ForZone)

Atomic: will be replaced during a merge

forZones indicates the zone(s) this endpoint should be consumed by to enable topology aware routing.

ForZone provides information about which zones should consume this endpoint.

- **endpoints.hints.forZones.name** (string), required

name represents the name of the zone.

- **endpoints.hostname** (string)

hostname of this endpoint. This field may be used by consumers of endpoints to distinguish endpoints from each other (e.g. in DNS names). Multiple endpoints which use the same hostname should be considered fungible (e.g. multiple A values in DNS). Must be lowercase and pass DNS Label (RFC 1123) validation.

- **endpoints.nodeName** (string)

nodeName represents the name of the Node hosting this

endpoint. This can be used to determine endpoints local to a Node.

- **endpoints.targetRef** ([ObjectReference](#))

targetRef is a reference to a Kubernetes object that represents this endpoint.

- **endpoints.zone** (string)

zone is the name of the Zone this endpoint exists in.

- **ports** ([]EndpointPort)

Atomic: will be replaced during a merge

ports specifies the list of network ports exposed by each endpoint in this slice. Each port must have a unique name. When ports is empty, it indicates that there are no defined ports. When a port is defined with a nil port value, it indicates "all ports". Each slice may include a maximum of 100 ports.

EndpointPort represents a Port used by an EndpointSlice

- **ports.port** (int32)

port represents the port number of the endpoint. If this is not specified, ports are not restricted and must be interpreted in the context of the specific consumer.

- **ports.protocol** (string)

protocol represents the IP protocol for this port. Must be UDP, TCP, or SCTP. Default is TCP.

- **ports.name** (string)

name represents the name of this port. All ports in an EndpointSlice must have a unique name. If the EndpointSlice is derived from a Kubernetes service, this corresponds to the Service.ports[].name. Name must either be an empty string or pass DNS_LABEL validation: * must be no more than 63 characters long. * must consist of lower case alphanumeric characters or '-'. * must start and end with an alphanumeric character. Default is empty string.

- **ports.appProtocol** (string)

The application protocol for this port. This is used as a hint for implementations to offer richer behavior for protocols that they understand. This field follows standard Kubernetes label syntax. Valid values are either:

- Un-prefixed protocol names - reserved for IANA standard service names (as per RFC-6335 and <https://www.iana.org/assignments/service-names>).

- Kubernetes-defined prefixed names:

- 'kubernetes.io/h2c' - HTTP/2 over cleartext as described in <https://www.rfc-editor.org/rfc/rfc7540>
- 'kubernetes.io/ws' - WebSocket over cleartext as described in <https://www.rfc-editor.org/rfc/rfc6455>
- 'kubernetes.io/wss' - WebSocket over TLS as described in <https://www.rfc-editor.org/rfc/rfc6455>

- Other protocols should use implementation-defined prefixed names such as mycompany.com/my-custom-

protocol.

EndpointSliceList

EndpointSliceList represents a list of endpoint slices

- **apiVersion**: discovery.k8s.io/v1

- **kind**: EndpointSliceList

- **metadata** ([ListMeta](#))

Standard list metadata.

- **items** ([][EndpointSlice](#)), required

items is the list of endpoint slices

Operations

get read the specified EndpointSlice

HTTP Request

```
GET /apis/discovery.k8s.io/v1/namespaces/{namespace}/endpointslices/  
{name}
```

Parameters

- **name** (*in path*): string, required

name of the EndpointSlice

- **namespace** (*in path*): string, required

[namespace](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([EndpointSlice](#)): OK

401: Unauthorized

list list or watch objects of kind EndpointSlice

HTTP Request

```
GET /apis/discovery.k8s.io/v1/namespaces/{namespace}/endpointslices
```

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([EndpointSliceList](#)): OK

401: Unauthorized

list list or watch objects of kind EndpointSlice

HTTP Request

GET /apis/discovery.k8s.io/v1/endpointslices

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)
- **watch** (*in query*): boolean
[watch](#)

Response

200 ([EndpointSliceList](#)): OK

401: Unauthorized

create create an EndpointSlice

HTTP Request

POST /apis/discovery.k8s.io/v1/namespaces/{namespace}/endpointslices

Parameters

- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [EndpointSlice](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([EndpointSlice](#)): OK

201 ([EndpointSlice](#)): Created

202 ([EndpointSlice](#)): Accepted

401: Unauthorized

update replace the specified EndpointSlice

HTTP Request

PUT /apis/discovery.k8s.io/v1/namespaces/{namespace}/endpointslices/{name}

Parameters

- **name** (*in path*): string, required
name of the EndpointSlice
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [EndpointSlice](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([EndpointSlice](#)): OK

201 ([EndpointSlice](#)): Created

401: Unauthorized

patch partially update the specified EndpointSlice

HTTP Request

PATCH /apis/discovery.k8s.io/v1/namespaces/{namespace}/endpointslices/{name}

Parameters

- **name** (*in path*): string, required
name of the EndpointSlice
- **namespace** (*in path*): string, required
[namespace](#)

- **body**: [Patch](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **force** (*in query*): boolean
[force](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([EndpointSlice](#)): OK

201 ([EndpointSlice](#)): Created

401: Unauthorized

delete delete an EndpointSlice

HTTP Request

DELETE /apis/discovery.k8s.io/v1/namespaces/{namespace}/
endpointslices/{name}

Parameters

- **name** (*in path*): string, required
name of the EndpointSlice
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [DeleteOptions](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)

Response

200 ([Status](#)): OK

202 ([Status](#)): Accepted

401: Unauthorized

deletecollection delete collection of EndpointSlice

HTTP Request

DELETE /apis/discovery.k8s.io/v1/namespaces/{namespace}/endpointslices

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [DeleteOptions](#)

- **continue** (*in query*): string

[continue](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.2.4 - Ingress

Ingress is a collection of rules that allow inbound connections to reach the endpoints defined by a backend.

```
apiVersion: networking.k8s.io/v1

import "k8s.io/api/networking/v1"
```

Ingress

Ingress is a collection of rules that allow inbound connections to reach the endpoints defined by a backend. An Ingress can be configured to give services externally-reachable urls, load balance traffic, terminate SSL, offer name based virtual hosting etc.

- **apiVersion**: networking.k8s.io/v1
- **kind**: Ingress
- **metadata** ([ObjectMeta](#))

Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **spec** ([IngressSpec](#))

spec is the desired state of the Ingress. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status>

- **status** ([IngressStatus](#))

status is the current state of the Ingress. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status>

IngressSpec

IngressSpec describes the Ingress the user wishes to exist.

- **defaultBackend** ([IngressBackend](#))

defaultBackend is the backend that should handle requests that don't match any rule. If Rules are not specified, DefaultBackend must be specified. If DefaultBackend is not set, the handling of requests that do not match any of the rules will be up to the Ingress controller.

- **ingressClassName** (string)

ingressClassName is the name of an IngressClass cluster resource. Ingress controller implementations use this field to know whether they should be serving this Ingress resource, by a transitive connection (controller -> IngressClass -> Ingress resource). Although the `kubernetes.io/ingress.class` annotation (simple constant name) was never formally defined, it was widely supported by Ingress controllers to create a direct binding between Ingress controller and Ingress resources. Newly created Ingress resources should prefer using the field. However, even though the annotation is officially

deprecated, for backwards compatibility reasons, ingress controllers should still honor that annotation if present.

- **rules** ([]IngressRule)

Atomic: will be replaced during a merge

rules is a list of host rules used to configure the Ingress. If unspecified, or no rule matches, all traffic is sent to the default backend.

IngressRule represents the rules mapping the paths under a specified host to the related backend services. Incoming requests are first evaluated for a host match, then routed to the backend associated with the matching IngressRuleValue.

- **rules.host** (string)

host is the fully qualified domain name of a network host, as defined by RFC 3986. Note the following deviations from the "host" part of the URI as defined in RFC 3986: 1. IPs are not allowed. Currently an IngressRuleValue can only apply to the IP in the Spec of the parent Ingress. 2. The : delimiter is not respected because ports are not allowed. Currently the port of an Ingress is implicitly :80 for http and :443 for https. Both these may change in the future. Incoming requests are matched against the host before the IngressRuleValue. If the host is unspecified, the Ingress routes all traffic based on the specified IngressRuleValue.

host can be "precise" which is a domain name without the terminating dot of a network host (e.g. "foo.bar.com") or "wildcard", which is a domain name prefixed with a single wildcard label (e.g. ".foo.com"). *The wildcard character* " must appear by itself as the first DNS label and matches only a single label. You cannot have a wildcard label by itself (e.g. Host == "*"). Requests will be matched against the Host field in the following way: 1. If host is precise, the request matches this rule if the http host header is equal to Host. 2. If host is a wildcard, then the request matches this rule if the http host header is to equal to the suffix (removing the first label) of the wildcard rule.

- **rules.http** (HTTPIngressRuleValue)

HTTPIngressRuleValue is a list of http selectors pointing to backends. In the example: http://? -> backend where where parts of the url correspond to RFC 3986, this resource will be used to match against everything after the last '/' and before the first '?' or '#'.

- **rules.http.paths** ([]HTTPIngressPath), required

Atomic: will be replaced during a merge

paths is a collection of paths that map requests to backends.

HTTPIngressPath associates a path with a backend. Incoming urls matching the path are forwarded to the backend.

- **rules.http.paths.backend** ([IngressBackend](#)), required

backend defines the referenced service endpoint to which the traffic will be forwarded to.

- **rules.http.paths.pathType** (string), required

pathType determines the interpretation of the path matching. PathType can be one of the following values: * Exact: Matches the URL path exactly. * Prefix: Matches based on a URL path prefix split by '/'. Matching is done on a path element by element basis. A path element refers to the list of labels in the path split by the '/' separator. A request is a match for path p if every p is an element-wise prefix of p of the request path. Note that if the last element of the path is a substring of the last element in request path, it is not a match (e.g. /foo/bar matches /foo/bar/baz, but does not match /foo/barbaz).

- ImplementationSpecific: Interpretation of the Path matching is up to the IngressClass. Implementations can treat this as a separate PathType or treat it identically to Prefix or Exact path types. Implementations are required to support all path types.

- **rules.http.paths.path** (string)

path is matched against the path of an incoming request. Currently it can contain characters disallowed from the conventional "path" part of a URL as defined by RFC 3986. Paths must begin with a '/' and must be present when using PathType with value "Exact" or "Prefix".

- **tls** ([]IngressTLS)

Atomic: will be replaced during a merge

tls represents the TLS configuration. Currently the Ingress only supports a single TLS port, 443. If multiple members of this list specify different hosts, they will be multiplexed on the same port according to the hostname specified through the SNI TLS extension, if the ingress controller fulfilling the ingress supports SNI.

IngressTLS describes the transport layer security associated with an ingress.

- **tls.hosts** ([]string)

Atomic: will be replaced during a merge

hosts is a list of hosts included in the TLS certificate. The values in this list must match the name/s used in the tlsSecret. Defaults to the wildcard host setting for the loadbalancer controller fulfilling this Ingress, if left unspecified.

- **tls.secretName** (string)

secretName is the name of the secret used to terminate TLS traffic on port 443. Field is left optional to allow TLS routing based on SNI hostname alone. If the SNI host in a listener conflicts with the "Host" header field used by an IngressRule, the SNI host is used for termination and value of the "Host" header is used for routing.

IngressBackend

IngressBackend describes all endpoints for a given service and port.

- **resource** ([TypedLocalObjectReference](#))

resource is an ObjectRef to another Kubernetes resource in the namespace of the Ingress object. If resource is specified, a service.Name and service.Port must not be specified. This is a mutually exclusive setting with "Service".

- **service** (IngressServiceBackend)

service references a service as a backend. This is a mutually exclusive setting with "Resource".

IngressServiceBackend references a Kubernetes Service as a Backend.

- **service.name** (string), required

name is the referenced service. The service must exist in the same namespace as the Ingress object.

- **service.port** (ServiceBackendPort)

port of the referenced service. A port name or port number is required for a IngressServiceBackend.

ServiceBackendPort is the service port being referenced.

- **service.port.name** (string)

name is the name of the port on the Service. This is a mutually exclusive setting with "Number".

- **service.port.number** (int32)

number is the numerical port number (e.g. 80) on the Service. This is a mutually exclusive setting with "Name".

IngressStatus

IngressStatus describe the current state of the Ingress.

- **loadBalancer** (IngressLoadBalancerStatus)

loadBalancer contains the current status of the load-balancer.

IngressLoadBalancerStatus represents the status of a load-balancer.

- **loadBalancer.ingress** ([]IngressLoadBalancerIngress)

ingress is a list containing ingress points for the load-balancer.

IngressLoadBalancerIngress represents the status of a load-balancer ingress point.

- **loadBalancer.ingress.hostname** (string)

hostname is set for load-balancer ingress points that are DNS based.

- **loadBalancer.ingress.ip** (string)

ip is set for load-balancer ingress points that are IP based.

- **loadBalancer.ingress.ports** ([]IngressPortStatus)

Atomic: will be replaced during a merge

ports provides information about the ports exposed by this LoadBalancer.

IngressPortStatus represents the error condition of a service port

- **loadBalancer.ingress.ports.port** (int32), required

port is the port number of the ingress port.

- **loadBalancer.ingress.ports.protocol** (string), required

protocol is the protocol of the ingress port. The supported values are: "TCP", "UDP", "SCTP"

- **loadBalancer.ingress.ports.error** (string)

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.

IngressList

IngressList is a collection of Ingress.

- **items** ([][Ingress](#)), required

items is the list of Ingress.

- **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** ([ListMeta](#))

Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

Operations

get read the specified Ingress

HTTP Request

GET /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name}

Parameters

- **name** (*in path*): string, required
name of the Ingress
- **namespace** (*in path*): string, required
[namespace](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([Ingress](#)): OK

401: Unauthorized

get read status of the specified Ingress

HTTP Request

GET /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name}/status

Parameters

- **name** (*in path*): string, required
name of the Ingress
- **namespace** (*in path*): string, required
[namespace](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([Ingress](#)): OK

401: Unauthorized

list list or watch objects of kind Ingress

HTTP Request

GET /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses

Parameters

- **namespace** (*in path*): string, required
[namespace](#)

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([IngressList](#)): OK

401: Unauthorized

list list or watch objects of kind Ingress

HTTP Request

GET /apis/networking.k8s.io/v1/ingresses

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([IngressList](#)): OK

401: Unauthorized

create create an Ingress

HTTP Request

POST /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [Ingress](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Ingress](#)): OK

201 ([Ingress](#)): Created

202 ([Ingress](#)): Accepted

401: Unauthorized

update replace the specified Ingress

HTTP Request

PUT /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name}

Parameters

- **name** (*in path*): string, required

name of the Ingress

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [Ingress](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Ingress](#)): OK

201 ([Ingress](#)): Created

401: Unauthorized

update replace status of the specified Ingress

HTTP Request

PUT /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name}/status

Parameters

- **name** (*in path*): string, required

name of the Ingress

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [Ingress](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([Ingress](#)): OK

201 ([Ingress](#)): Created

401: Unauthorized

patch partially update the specified Ingress

HTTP Request

PATCH /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name}

Parameters

- **name** (*in path*): string, required
name of the Ingress
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [Patch](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **force** (*in query*): boolean
[force](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([Ingress](#)): OK

201 ([Ingress](#)): Created

401: Unauthorized

patch partially update status of the specified Ingress

HTTP Request

PATCH /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name}/status

Parameters

- **name** (*in path*): string, required
name of the Ingress
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [Patch](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **force** (*in query*): boolean
[force](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([Ingress](#)): OK

201 ([Ingress](#)): Created

401: Unauthorized

delete delete an Ingress

HTTP Request

DELETE /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name}

Parameters

- **name** (*in path*): string, required
name of the Ingress

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [DeleteOptions](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

Response

200 ([Status](#)): OK

202 ([Status](#)): Accepted

401: Unauthorized

deletecollection delete collection of Ingress

HTTP Request

DELETE /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [DeleteOptions](#)

- **continue** (*in query*): string

[continue](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.2.5 - IngressClass

IngressClass represents the class of the Ingress, referenced by the Ingress Spec.

```
apiVersion: networking.k8s.io/v1
import "k8s.io/api/networking/v1"
```

IngressClass

IngressClass represents the class of the Ingress, referenced by the Ingress Spec. The `ingressclass.kubernetes.io/is-default-class` annotation can be used to indicate that an IngressClass should be considered default. When a single IngressClass resource has this annotation set to true, new Ingress resources without a class specified will be assigned this default class.

- **apiVersion**: networking.k8s.io/v1

- **kind**: IngressClass

- **metadata** ([ObjectMeta](#))

Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **spec** ([IngressClassSpec](#))

spec is the desired state of the IngressClass. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status>

IngressClassSpec

IngressClassSpec provides information about the class of an Ingress.

- **controller** (string)

controller refers to the name of the controller that should handle this class. This allows for different "flavors" that are controlled by the same controller. For example, you may have different parameters for the same implementing controller. This should be specified as a domain-prefixed path no more than 250 characters in length, e.g. "acme.io/ingress-controller". This field is immutable.

- **parameters** ([IngressClassParametersReference](#))

parameters is a link to a custom resource containing additional configuration for the controller. This is optional if the controller does not require extra parameters.

IngressClassParametersReference identifies an API object. This can be used to specify a cluster or namespace-scoped resource.

- **parameters.kind** (string), required

kind is the type of resource being referenced.

- **parameters.name** (string), required

name is the name of resource being referenced.

- **parameters.apiGroup** (string)

apiGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.

- **parameters.namespace** (string)

namespace is the namespace of the resource being referenced. This field is required when scope is set to "Namespace" and must be unset when scope is set to "Cluster".

- **parameters.scope** (string)

scope represents if this refers to a cluster or namespace scoped resource. This may be set to "Cluster" (default) or "Namespace".

IngressClassList

IngressClassList is a collection of IngressClasses.

- **apiVersion**: networking.k8s.io/v1
- **kind**: IngressClassList
- **metadata** ([ListMeta](#))

Standard list metadata.

- **items** ([] [IngressClass](#)), required

items is the list of IngressClasses.

Operations

get read the specified IngressClass

HTTP Request

GET /apis/networking.k8s.io/v1/ingressclasses/{name}

Parameters

- **name** (*in path*): string, required

name of the IngressClass

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([IngressClass](#)): OK

401: Unauthorized

list list or watch objects of kind IngressClass

HTTP Request

GET /apis/networking.k8s.io/v1/ingressclasses

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([IngressClassList](#)): OK

401: Unauthorized

create create an IngressClass

HTTP Request

POST /apis/networking.k8s.io/v1/ingressclasses

Parameters

- **body**: [IngressClass](#), required

- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([IngressClass](#)): OK
201 ([IngressClass](#)): Created
202 ([IngressClass](#)): Accepted
401: Unauthorized

update replace the specified IngressClass

HTTP Request

PUT /apis/networking.k8s.io/v1/ingressclasses/{name}

Parameters

- **name** (*in path*): string, required
name of the IngressClass
- **body**: [IngressClass](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([IngressClass](#)): OK
201 ([IngressClass](#)): Created
401: Unauthorized

patch partially update the specified IngressClass

HTTP Request

PATCH /apis/networking.k8s.io/v1/ingressclasses/{name}

Parameters

- **name** (*in path*): string, required

name of the IngressClass

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([IngressClass](#)): OK

201 ([IngressClass](#)): Created

401: Unauthorized

delete delete an IngressClass

HTTP Request

DELETE /apis/networking.k8s.io/v1/ingressclasses/{name}

Parameters

- **name** (*in path*): string, required

name of the IngressClass

- **body**: [DeleteOptions](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

Response

200 ([Status](#)): OK

202 ([Status](#)): Accepted

401: Unauthorized

deletecollection delete collection of IngressClass

HTTP Request

DELETE /apis/networking.k8s.io/v1/ingressclasses

Parameters

- **body**: [DeleteOptions](#)
- **continue** (*in query*): string
[continue](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.3 - Config and Storage Resources

5.3.1 - ConfigMap

ConfigMap holds configuration data for pods to consume.

```
apiVersion: v1

import "k8s.io/api/core/v1"
```

ConfigMap

ConfigMap holds configuration data for pods to consume.

- **apiVersion**: v1
- **kind**: ConfigMap
- **metadata** ([ObjectMeta](#))

Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **binaryData** (map[string][]byte)

BinaryData contains the binary data. Each key must consist of alphanumeric characters, '-', '_' or '!'. BinaryData can contain byte sequences that are not in the UTF-8 range. The keys stored in BinaryData must not overlap with the ones in the Data field, this is enforced during validation process. Using this field will require 1.10+ apiserver and kubelet.

- **data** (map[string]string)

Data contains the configuration data. Each key must consist of alphanumeric characters, '-', '_' or '!'. Values with non-UTF-8 byte sequences must use the BinaryData field. The keys stored in Data must not overlap with the keys in the BinaryData field, this is enforced during validation process.

- **immutable** (boolean)

Immutable, if set to true, ensures that data stored in the ConfigMap cannot be updated (only object metadata can be modified). If not set to true, the field can be modified at any time. Defaulted to nil.

ConfigMapList

ConfigMapList is a resource containing a list of ConfigMap objects.

- **apiVersion**: v1
- **kind**: ConfigMapList
- **metadata** ([ListMeta](#))

More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **items** ([][ConfigMap](#)), required

Items is the list of ConfigMaps.

Operations

get read the specified ConfigMap

HTTP Request

GET /api/v1/namespaces/{namespace}/configmaps/{name}

Parameters

- **name** (*in path*): string, required
name of the ConfigMap
- **namespace** (*in path*): string, required
[namespace](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([ConfigMap](#)): OK

401: Unauthorized

list list or watch objects of kind ConfigMap

HTTP Request

GET /api/v1/namespaces/{namespace}/configmaps

Parameters

- **namespace** (*in path*): string, required
[namespace](#)

- **allowWatchBookmarks** (*in query*): boolean
[allowWatchBookmarks](#)

- **continue** (*in query*): string
[continue](#)

- **fieldSelector** (*in query*): string
[fieldSelector](#)

- **labelSelector** (*in query*): string
[labelSelector](#)

- **limit** (*in query*): integer
[limit](#)

- **pretty** (*in query*): string
[pretty](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)
- **watch** (*in query*): boolean
[watch](#)

Response

200 ([ConfigMapList](#)): OK

401: Unauthorized

list list or watch objects of kind ConfigMap

HTTP Request

GET /api/v1/configmaps

Parameters

- **allowWatchBookmarks** (*in query*): boolean
[allowWatchBookmarks](#)
- **continue** (*in query*): string
[continue](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([ConfigMapList](#)): OK

401: Unauthorized

create create a ConfigMap

HTTP Request

POST /api/v1/namespaces/{namespace}/configmaps

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [ConfigMap](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ConfigMap](#)): OK

201 ([ConfigMap](#)): Created

202 ([ConfigMap](#)): Accepted

401: Unauthorized

update replace the specified ConfigMap

HTTP Request

PUT /api/v1/namespaces/{namespace}/configmaps/{name}

Parameters

- **name** (*in path*): string, required
name of the ConfigMap
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [ConfigMap](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([ConfigMap](#)): OK

201 ([ConfigMap](#)): Created

401: Unauthorized

patch partially update the specified ConfigMap

HTTP Request

PATCH /api/v1/namespaces/{namespace}/configmaps/{name}

Parameters

- **name** (*in path*): string, required
name of the ConfigMap
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [Patch](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **force** (*in query*): boolean
[force](#)
- **pretty** (*in query*): string
[pretty](#)

[pretty](#)

Response

200 ([ConfigMap](#)): OK

201 ([ConfigMap](#)): Created

401: Unauthorized

delete delete a ConfigMap

HTTP Request

DELETE /api/v1/namespaces/{namespace}/configmaps/{name}

Parameters

- **name** (*in path*): string, required
[name](#) of the ConfigMap
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [DeleteOptions](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)

Response

200 ([Status](#)): OK

202 ([Status](#)): Accepted

401: Unauthorized

deletecollection delete collection of ConfigMap

HTTP Request

DELETE /api/v1/namespaces/{namespace}/configmaps

Parameters

- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [DeleteOptions](#)

- **continue** (*in query*): string

[continue](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.3.2 - Secret

Secret holds secret data of a certain type.

```
apiVersion: v1

import "k8s.io/api/core/v1"
```

Secret

Secret holds secret data of a certain type. The total bytes of the values in the Data field must be less than MaxSecretSize bytes.

- **apiVersion**: v1

- **kind**: Secret

- **metadata** ([ObjectMeta](#))

Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **data** (map[string][]byte)

Data contains the secret data. Each key must consist of alphanumeric characters, '-', '_' or '!'. The serialized form of the secret data is a base64 encoded string, representing the arbitrary (possibly non-string) data value here. Described in <https://tools.ietf.org/html/rfc4648#section-4>

- **immutable** (boolean)

Immutable, if set to true, ensures that data stored in the Secret cannot be updated (only object metadata can be modified). If not set to true, the field can be modified at any time. Defaulted to nil.

- **stringData** (map[string]string)

stringData allows specifying non-binary secret data in string form. It is provided as a write-only input field for convenience. All keys and values are merged into the data field on write, overwriting any existing values. The stringData field is never output when reading from the API.

- **type** (string)

Used to facilitate programmatic handling of secret data. More info: <https://kubernetes.io/docs/concepts/configuration/secret/#secret-types>

SecretList

SecretList is a list of Secret.

- **apiVersion**: v1

- **kind**: SecretList

- **metadata** ([ListMeta](#))

Standard list metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types->

[kinds](#)

- **items** ([][Secret](#)), required

Items is a list of secret objects. More info: <https://kubernetes.io/docs/concepts/configuration/secret>

Operations

[get](#) read the specified Secret

HTTP Request

GET /api/v1/namespaces/{namespace}/secrets/{name}

Parameters

- **name** (*in path*): string, required
name of the Secret
- **namespace** (*in path*): string, required
[namespace](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([Secret](#)): OK

401: Unauthorized

[list](#) list or watch objects of kind Secret

HTTP Request

GET /api/v1/namespaces/{namespace}/secrets

Parameters

- **namespace** (*in path*): string, required
[namespace](#)
- **allowWatchBookmarks** (*in query*): boolean
[allowWatchBookmarks](#)
- **continue** (*in query*): string
[continue](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **labelSelector** (*in query*): string
[labelSelector](#)

- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)
- **watch** (*in query*): boolean
[watch](#)

Response

200 ([SecretList](#)): OK

401: Unauthorized

list list or watch objects of kind Secret

HTTP Request

GET /api/v1/secrets

Parameters

- **allowWatchBookmarks** (*in query*): boolean
[allowWatchBookmarks](#)
- **continue** (*in query*): string
[continue](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([SecretList](#)): OK

401: Unauthorized

create create a Secret

HTTP Request

POST /api/v1/namespaces/{namespace}/secrets

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [Secret](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Secret](#)): OK

201 ([Secret](#)): Created

202 ([Secret](#)): Accepted

401: Unauthorized

update replace the specified Secret

HTTP Request

PUT /api/v1/namespaces/{namespace}/secrets/{name}

Parameters

- **name** (*in path*): string, required
name of the Secret
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [Secret](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([Secret](#)): OK

201 ([Secret](#)): Created

401: Unauthorized

patch partially update the specified Secret

HTTP Request

PATCH /api/v1/namespaces/{namespace}/secrets/{name}

Parameters

- **name** (*in path*): string, required
name of the Secret
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [Patch](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Secret](#)): OK201 ([Secret](#)): Created

401: Unauthorized

delete delete a Secret

HTTP Request

DELETE /api/v1/namespaces/{namespace}/secrets/{name}

Parameters

- **name** (*in path*): string, required

name of the Secret

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [DeleteOptions](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

Response

200 ([Status](#)): OK202 ([Status](#)): Accepted

401: Unauthorized

deletecollection delete collection of Secret

HTTP Request

DELETE /api/v1/namespaces/{namespace}/secrets

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [DeleteOptions](#)
- **continue** (*in query*): string
[continue](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.3.3 - Volume

Volume represents a named volume in a pod that may be accessed by any container in the pod.

```
import "k8s.io/api/core/v1"
```

Volume

Volume represents a named volume in a pod that may be accessed by any container in the pod.

- **name** (string), required

name of the volume. Must be a DNS_LABEL and unique within the pod. More info: <https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names>

Exposed Persistent volumes

- **persistentVolumeClaim** (PersistentVolumeClaimVolumeSource)

persistentVolumeClaimVolumeSource represents a reference to a PersistentVolumeClaim in the same namespace. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims>

PersistentVolumeClaimVolumeSource references the user's PVC in the same namespace. This volume finds the bound PV and mounts that volume for the pod. A PersistentVolumeClaimVolumeSource is, essentially, a wrapper around another type of volume that is owned by someone else (the system).

- **persistentVolumeClaim.claimName** (string), required

claimName is the name of a PersistentVolumeClaim in the same namespace as the pod using this volume. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims>

- **persistentVolumeClaim.readOnly** (boolean)

readOnly Will force the ReadOnly setting in VolumeMounts. Default false.

Projections

- **configMap** (ConfigMapVolumeSource)

configMap represents a configMap that should populate this volume

*Adapts a ConfigMap into a volume.

The contents of the target ConfigMap's Data field will be presented in a volume as files using the keys in the Data field as the file names, unless the items element is populated with specific mappings of keys to paths. ConfigMap volumes support ownership management and SELinux relabeling.*

- **configMap.name** (string)

Name of the referent. More info: <https://kubernetes.io/docs/>

[concepts/overview/working-with-objects/names/#names](https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names)

- **configMap.optional** (boolean)

optional specify whether the ConfigMap or its keys must be defined

- **configMap.defaultMode** (int32)

defaultMode is optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

- **configMap.items** ([][KeyToPath](#))

items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'

- **secret** (SecretVolumeSource)

secret represents a secret that should populate this volume. More info: <https://kubernetes.io/docs/concepts/storage/volumes#secret>

*Adapts a Secret into a volume.

The contents of the target Secret's Data field will be presented in a volume as files using the keys in the Data field as the file names. Secret volumes support ownership management and SELinux relabeling.*

- **secret.secretName** (string)

secretName is the name of the secret in the pod's namespace to use. More info: <https://kubernetes.io/docs/concepts/storage/volumes#secret>

- **secret.optional** (boolean)

optional field specify whether the Secret or its keys must be defined

- **secret.defaultMode** (int32)

defaultMode is Optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

- **secret.items** ([][KeyToPath](#))

items If unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified,

the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

- **downwardAPI** (DownwardAPIVolumeSource)

downwardAPI represents downward API about the pod that should populate this volume

DownwardAPIVolumeSource represents a volume containing downward API info. Downward API volumes support ownership management and SELinux relabeling.

- **downwardAPI.defaultMode** (int32)

Optional: mode bits to use on created files by default. Must be a Optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

- **downwardAPI.items** ([]DownwardAPIVolumeFile)

Items is a list of downward API volume file

- **projected** (ProjectedVolumeSource)

projected items for all in one resources secrets, configmaps, and downward API

Represents a projected volume source

- **projected.defaultMode** (int32)

defaultMode are the mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

- **projected.sources** ([]VolumeProjection)

sources is the list of volume projections

Projection that may be projected along with other supported volume types

- **projected.sources.configMap** (ConfigMapProjection)

configMap information about the configMap data to project

*Adapts a ConfigMap into a projected volume.

The contents of the target ConfigMap's Data field will be presented in a projected volume as files using the keys in the Data field as the file names, unless the items element is populated with specific mappings of keys to paths.

Note that this is identical to a configmap volume source

without the default mode.*

- **projected.sources.configMap.name** (string)

Name of the referent. More info: <https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names>

- **projected.sources.configMap.optional** (boolean)

optional specify whether the ConfigMap or its keys must be defined

- **projected.sources.configMap.items** ([][KeyToPath](#))

items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '!'.

- **projected.sources.downwardAPI**

(DownwardAPIProjection)

downwardAPI information about the downwardAPI data to project

Represents downward API info for projecting into a projected volume. Note that this is identical to a downwardAPI volume source without the default mode.

- **projected.sources.downwardAPI.items**

([][DownwardAPIVolumeFile](#))

Items is a list of DownwardAPIVolume file

- **projected.sources.secret** (SecretProjection)

secret information about the secret data to project

*Adapts a secret into a projected volume.

The contents of the target Secret's Data field will be presented in a projected volume as files using the keys in the Data field as the file names. Note that this is identical to a secret volume source without the default mode.*

- **projected.sources.secret.name** (string)

Name of the referent. More info: <https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names>

- **projected.sources.secret.optional** (boolean)

optional field specify whether the Secret or its key must be defined

- **projected.sources.secret.items** ([][KeyToPath](#))

items if unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will

be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

- **projected.sources.serviceAccountToken**

(ServiceAccountTokenProjection)

serviceAccountToken is information about the serviceAccountToken data to project

ServiceAccountTokenProjection represents a projected service account token volume. This projection can be used to insert a service account token into the pods runtime filesystem for use against APIs (Kubernetes API Server or otherwise).

- **projected.sources.serviceAccountToken.path**

(string), required

path is the path relative to the mount point of the file to project the token into.

- **projected.sources.serviceAccountToken.audience**

(string)

audience is the intended audience of the token. A recipient of a token must identify itself with an identifier specified in the audience of the token, and otherwise should reject the token. The audience defaults to the identifier of the apiserver.

- **projected.sources.serviceAccountToken.expirationSeconds**

(int64)

expirationSeconds is the requested duration of validity of the service account token. As the token approaches expiration, the kubelet volume plugin will proactively rotate the service account token. The kubelet will start trying to rotate the token if the token is older than 80 percent of its time to live or if the token is older than 24 hours. Defaults to 1 hour and must be at least 10 minutes.

Local / Temporary Directory

- **emptyDir** (EmptyDirVolumeSource)

emptyDir represents a temporary directory that shares a pod's lifetime. More info: <https://kubernetes.io/docs/concepts/storage/volumes#emptydir>

Represents an empty directory for a pod. Empty directory volumes support ownership management and SELinux relabeling.

- **emptyDir.medium** (string)

medium represents what type of storage medium should back this directory. The default is "" which means to use the node's default medium. Must be an empty string (default) or Memory.

More info: <https://kubernetes.io/docs/concepts/storage/volumes#emptydir>

- **emptyDir.sizeLimit** ([Quantity](#))

sizeLimit is the total amount of local storage required for this EmptyDir volume. The size limit is also applicable for memory medium. The maximum usage on memory medium EmptyDir would be the minimum value between the SizeLimit specified here and the sum of memory limits of all containers in a pod. The default is nil which means that the limit is undefined.

More info: <https://kubernetes.io/docs/concepts/storage/volumes#emptydir>

- **hostPath** (HostPathVolumeSource)

hostPath represents a pre-existing file or directory on the host machine that is directly exposed to the container. This is generally used for system agents or other privileged things that are allowed to see the host machine. Most containers will NOT need this. More info: <https://kubernetes.io/docs/concepts/storage/volumes#hostpath>

Represents a host path mapped into a pod. Host path volumes do not support ownership management or SELinux relabeling.

- **hostPath.path** (string), required

path of the directory on the host. If the path is a symlink, it will follow the link to the real path. More info: <https://kubernetes.io/docs/concepts/storage/volumes#hostpath>

- **hostPath.type** (string)

type for HostPath Volume Defaults to "" More info: <https://kubernetes.io/docs/concepts/storage/volumes#hostpath>

Persistent volumes

- **awsElasticBlockStore** (AWSElasticBlockStoreVolumeSource)

awsElasticBlockStore represents an AWS Disk resource that is attached to a kubelet's host machine and then exposed to the pod.

More info: <https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore>

*Represents a Persistent Disk resource in AWS.

An AWS EBS disk must exist before mounting to a container. The disk must also be in the same AWS zone as the kubelet. An AWS EBS disk can only be mounted as read/write once. AWS EBS volumes support ownership management and SELinux relabeling.*

- **awsElasticBlockStore.volumeID** (string), required

volumeID is unique ID of the persistent disk resource in AWS (Amazon EBS volume). More info: <https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore>

- **awsElasticBlockStore.fsType** (string)

fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs".

Implicitly inferred to be "ext4" if unspecified. More info:

<https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore>

- **awsElasticBlockStore.partition** (int32)

partition is the partition in the volume that you want to mount. If omitted, the default is to mount by volume name. Examples:

For volume /dev/sda1, you specify the partition as "1".
Similarly, the volume partition for /dev/sda is "0" (or you can leave the property empty).

- **awsElasticBlockStore.readOnly** (boolean)

readOnly value true will force the readOnly setting in VolumeMounts. More info: <https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore>

- **azureDisk** (AzureDiskVolumeSource)

azureDisk represents an Azure Data Disk mount on the host and bind mount to the pod.

AzureDisk represents an Azure Data Disk mount on the host and bind mount to the pod.

- **azureDisk.diskName** (string), required

diskName is the Name of the data disk in the blob storage

- **azureDisk.diskURI** (string), required

diskURI is the URI of data disk in the blob storage

- **azureDisk.cachingMode** (string)

cachingMode is the Host Caching mode: None, Read Only, Read Write.

- **azureDisk.fsType** (string)

fsType is Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.

- **azureDisk.kind** (string)

kind expected values are Shared: multiple blob disks per storage account Dedicated: single blob disk per storage account Managed: azure managed data disk (only in managed availability set). defaults to shared

- **azureDisk.readOnly** (boolean)

readOnly Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

- **azureFile** (AzureFileVolumeSource)

azureFile represents an Azure File Service mount on the host and bind mount to the pod.

AzureFile represents an Azure File Service mount on the host and bind mount to the pod.

- **azureFile.secretName** (string), required

secretName is the name of secret that contains Azure Storage Account Name and Key

- **azureFile.shareName** (string), required

shareName is the azure share Name

- **azureFile.readOnly** (boolean)

readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

- **cephfs** (CephFSVolumeSource)

cephFS represents a Ceph FS mount on the host that shares a pod's lifetime

Represents a Ceph Filesystem mount that lasts the lifetime of a pod. Cephfs volumes do not support ownership management or SELinux relabeling.

- **cephfs.monitors** ([]string), required

monitors is Required: Monitors is a collection of Ceph monitors More info: <https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it>

- **cephfs.path** (string)

path is Optional: Used as the mounted root, rather than the full Ceph tree, default is /

- **cephfs.readOnly** (boolean)

readOnly is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts. More info: <https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it>

- **cephfs.secretFile** (string)

secretFile is Optional: SecretFile is the path to key ring for User, default is /etc/ceph/user.secret More info: <https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it>

- **cephfs.secretRef** ([LocalObjectReference](#))

secretRef is Optional: SecretRef is reference to the authentication secret for User, default is empty. More info: <https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it>

- **cephfs.user** (string)

user is optional: User is the rados user name, default is admin More info: <https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it>

- **cinder** (CinderVolumeSource)

cinder represents a cinder volume attached and mounted on kubelets host machine. More info: <https://examples.k8s.io/mysql-cinder-pd/README.md>

Represents a cinder volume resource in Openstack. A Cinder volume must exist before mounting to a container. The volume must also be in the same region as the kubelet. Cinder volumes support ownership management and SELinux relabeling.

- **cinder.volumeID** (string), required

volumeID used to identify the volume in cinder. More info: <https://examples.k8s.io/mysql-cinder-pd/README.md>

- **cinder.fsType** (string)

fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: <https://examples.k8s.io/mysql-cinder-pd/README.md>

- **cinder.readOnly** (boolean)

readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts. More info: <https://examples.k8s.io/mysql-cinder-pd/README.md>

- **cinder.secretRef** ([LocalObjectReference](#))

secretRef is optional: points to a secret object containing parameters used to connect to OpenStack.

- **csi** (CSIVolumeSource)

csi (Container Storage Interface) represents ephemeral storage that is handled by certain external CSI drivers (Beta feature).

Represents a source location of a volume to mount, managed by an external CSI driver

- **csi.driver** (string), required

driver is the name of the CSI driver that handles this volume. Consult with your admin for the correct name as registered in the cluster.

- **csi.fsType** (string)

fsType to mount. Ex. "ext4", "xfs", "ntfs". If not provided, the empty value is passed to the associated CSI driver which will determine the default filesystem to apply.

- **csi.nodePublishSecretRef** ([LocalObjectReference](#))

nodePublishSecretRef is a reference to the secret object containing sensitive information to pass to the CSI driver to complete the CSI NodePublishVolume and NodeUnpublishVolume calls. This field is optional, and may be empty if no secret is required. If the secret object contains more than one secret, all secret references are passed.

- **csi.readOnly** (boolean)

readOnly specifies a read-only configuration for the volume. Defaults to false (read/write).

- **csi.volumeAttributes** (map[string]string)

volumeAttributes stores driver-specific properties that are passed to the CSI driver. Consult your driver's documentation for supported values.

- **ephemeral** (EphemeralVolumeSource)

ephemeral represents a volume that is handled by a cluster storage driver. The volume's lifecycle is tied to the pod that defines it - it will be created before the pod starts, and deleted when the pod is removed.

Use this if: a) the volume is only needed while the pod runs, b) features of normal volumes like restoring from snapshot or capacity tracking are needed, c) the storage driver is specified through a storage class, and d) the storage driver supports dynamic volume provisioning through a PersistentVolumeClaim (see EphemeralVolumeSource for more information on the connection between this volume type and PersistentVolumeClaim).

Use PersistentVolumeClaim or one of the vendor-specific APIs for volumes that persist for longer than the lifecycle of an individual

pod.

Use CSI for light-weight local ephemeral volumes if the CSI driver is meant to be used that way - see the documentation of the driver for more information.

A pod can use both types of ephemeral volumes and persistent volumes at the same time.

Represents an ephemeral volume that is handled by a normal storage driver.

- **ephemeral.volumeClaimTemplate**

(PersistentVolumeClaimTemplate)

Will be used to create a stand-alone PVC to provision the volume. The pod in which this EphemeralVolumeSource is embedded will be the owner of the PVC, i.e. the PVC will be deleted together with the pod. The name of the PVC will be `\<pod name>-<volume name>` where `\<volume name>` is the name from the `PodSpec.Volumes` array entry. Pod validation will reject the pod if the concatenated name is not valid for a PVC (for example, too long).

An existing PVC with that name that is not owned by the pod will *not* be used for the pod to avoid using an unrelated volume by mistake. Starting the pod is then blocked until the unrelated PVC is removed. If such a pre-created PVC is meant to be used by the pod, the PVC has to be updated with an owner reference to the pod once the pod exists. Normally this should not be necessary, but it may be useful when manually reconstructing a broken cluster.

This field is read-only and no changes will be made by Kubernetes to the PVC after it has been created.

Required, must not be nil.

PersistentVolumeClaimTemplate is used to produce PersistentVolumeClaim objects as part of an EphemeralVolumeSource.

- **ephemeral.volumeClaimTemplate.spec**

([PersistentVolumeClaimSpec](#)), required

The specification for the PersistentVolumeClaim. The entire content is copied unchanged into the PVC that gets created from this template. The same fields as in a PersistentVolumeClaim are also valid here.

- **ephemeral.volumeClaimTemplate.metadata**

([ObjectMeta](#))

May contain labels and annotations that will be copied into the PVC when creating it. No other fields are allowed and will be rejected during validation.

- **fc** (FCVolumeSource)

fc represents a Fibre Channel resource that is attached to a kubelet's host machine and then exposed to the pod.

Represents a Fibre Channel volume. Fibre Channel volumes can only be mounted as read/write once. Fibre Channel volumes support ownership management and SELinux relabeling.

- **fc.fsType** (string)

fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.

- **fc.lun** (int32)

lun is Optional: FC target lun number

- **fc.readOnly** (boolean)

readOnly is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

- **fc.targetWWNs** ([]string)

targetWWNs is Optional: FC target worldwide names (WWNs)

- **fc.wwids** ([]string)

wwids Optional: FC volume world wide identifiers (wwids)

Either wwid or combination of targetWWNs and lun must be set, but not both simultaneously.

- **flexVolume** (FlexVolumeSource)

flexVolume represents a generic volume resource that is provisioned/attached using an exec based plugin.

FlexVolume represents a generic volume resource that is provisioned/attached using an exec based plugin.

- **flexVolume.driver** (string), required

driver is the name of the driver to use for this volume.

- **flexVolume.fsType** (string)

fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". The default filesystem depends on FlexVolume script.

- **flexVolume.options** (map[string]string)

options is Optional: this field holds extra command options if any.

- **flexVolume.readOnly** (boolean)

readOnly is Optional: defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

- **flexVolume.secretRef** ([LocalObjectReference](#))

secretRef is Optional: secretRef is reference to the secret object containing sensitive information to pass to the plugin scripts. This may be empty if no secret object is specified. If the secret object contains more than one secret, all secrets are passed to the plugin scripts.

- **flocker** (FlockerVolumeSource)

flocker represents a Flocker volume attached to a kubelet's host machine. This depends on the Flocker control service being running

Represents a Flocker volume mounted by the Flocker agent. One and only one of datasetName and datasetUUID should be set. Flocker volumes do not support ownership management or SELinux relabeling.

- **flocker.datasetName** (string)

datasetName is Name of the dataset stored as metadata -> name on the dataset for Flocker should be considered as deprecated

- **flocker.datasetUUID** (string)

datasetUUID is the UUID of the dataset. This is unique identifier of a Flocker dataset

- **gcePersistentDisk** (GCEPersistentDiskVolumeSource)

gcePersistentDisk represents a GCE Disk resource that is attached to a kubelet's host machine and then exposed to the pod. More info: <https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk>

*Represents a Persistent Disk resource in Google Compute Engine.

A GCE PD must exist before mounting to a container. The disk must also be in the same GCE project and zone as the kubelet. A GCE PD can only be mounted as read/write once or read-only many times. GCE PDs support ownership management and SELinux relabeling.*

- **gcePersistentDisk.pdName** (string), required

pdName is unique name of the PD resource in GCE. Used to identify the disk in GCE. More info: <https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk>

- **gcePersistentDisk.fsType** (string)

fsType is filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: <https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk>

- **gcePersistentDisk.partition** (int32)

partition is the partition in the volume that you want to mount. If omitted, the default is to mount by volume name. Examples: For volume /dev/sda1, you specify the partition as "1". Similarly, the volume partition for /dev/sda is "0" (or you can leave the property empty). More info: <https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk>

- **gcePersistentDisk.readOnly** (boolean)

readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false. More info: <https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk>

- **glusterfs** (GlusterfsVolumeSource)

glusterfs represents a Glusterfs mount on the host that shares a pod's lifetime. More info: <https://examples.k8s.io/volumes/glusterfs/README.md>

Represents a Glusterfs mount that lasts the lifetime of a pod. Glusterfs volumes do not support ownership management or SELinux relabeling.

- **glusterfs.endpoints** (string), required

endpoints is the endpoint name that details Glusterfs topology. More info: <https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod>

- **glusterfs.path** (string), required

path is the Glusterfs volume path. More info: <https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod>

- **glusterfs.readOnly** (boolean)

readOnly here will force the Glusterfs volume to be mounted with read-only permissions. Defaults to false. More info: <https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod>

- **iscsi** (ISCSIVolumeSource)

iscsi represents an iSCSI Disk resource that is attached to a kubelet's host machine and then exposed to the pod. More info: <https://examples.k8s.io/volumes/iscsi/README.md>

Represents an iSCSI disk. iSCSI volumes can only be mounted as read/write once. iSCSI volumes support ownership management and SELinux relabeling.

- **iscsi.iqn** (string), required

iqn is the target iSCSI Qualified Name.

- **iscsi.lun** (int32), required

lun represents iSCSI Target Lun number.

- **iscsi.targetPortal** (string), required

targetPortal is iSCSI Target Portal. The Portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

- **iscsi.chapAuthDiscovery** (boolean)

chapAuthDiscovery defines whether support iSCSI Discovery CHAP authentication

- **iscsi.chapAuthSession** (boolean)

chapAuthSession defines whether support iSCSI Session CHAP authentication

- **iscsi.fsType** (string)

fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: <https://kubernetes.io/docs/concepts/storage/volumes#iscsi>

- **iscsi.initiatorName** (string)

initiatorName is the custom iSCSI Initiator Name. If initiatorName is specified with iscsiInterface simultaneously, new iSCSI interface <target portal>:<volume name> will be created for the connection.

- **iscsi.iscsiInterface** (string)

iscsiInterface is the interface Name that uses an iSCSI transport. Defaults to 'default' (tcp).

- **iscsi.portals** ([]string)

portals is the iSCSI Target Portal List. The portal is either an IP or ip_addr:port if the port is other than default (typically TCP

ports 860 and 3260).

- **iscsi.readOnly** (boolean)

readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false.

- **iscsi.secretRef** ([LocalObjectReference](#))

secretRef is the CHAP Secret for iSCSI target and initiator authentication

- **nfs** (NFSVolumeSource)

nfs represents an NFS mount on the host that shares a pod's lifetime More info: <https://kubernetes.io/docs/concepts/storage/volumes#nfs>

Represents an NFS mount that lasts the lifetime of a pod. NFS volumes do not support ownership management or SELinux relabeling.

- **nfs.path** (string), required

path that is exported by the NFS server. More info: <https://kubernetes.io/docs/concepts/storage/volumes#nfs>

- **nfs.server** (string), required

server is the hostname or IP address of the NFS server. More info: <https://kubernetes.io/docs/concepts/storage/volumes#nfs>

- **nfs.readOnly** (boolean)

readOnly here will force the NFS export to be mounted with read-only permissions. Defaults to false. More info: <https://kubernetes.io/docs/concepts/storage/volumes#nfs>

- **photonPersistentDisk** (PhotonPersistentDiskVolumeSource)

photonPersistentDisk represents a PhotonController persistent disk attached and mounted on kubelets host machine

Represents a Photon Controller persistent disk resource.

- **photonPersistentDisk.pdID** (string), required

pdID is the ID that identifies Photon Controller persistent disk

- **photonPersistentDisk.fsType** (string)

fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.

- **portworxVolume** (PortworxVolumeSource)

portworxVolume represents a portworx volume attached and mounted on kubelets host machine

PortworxVolumeSource represents a Portworx volume resource.

- **portworxVolume.volumeID** (string), required

volumeID uniquely identifies a Portworx volume

- **portworxVolume.fsType** (string)

fsType represents the filesystem type to mount Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs". Implicitly inferred to be "ext4" if unspecified.

- **portworxVolume.readOnly** (boolean)

readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

- **quobyte** (QuobyteVolumeSource)

quobyte represents a Quobyte mount on the host that shares a pod's lifetime

Represents a Quobyte mount that lasts the lifetime of a pod. Quobyte volumes do not support ownership management or SELinux relabeling.

- **quobyte.registry** (string), required

registry represents a single or multiple Quobyte Registry services specified as a string as host:port pair (multiple entries are separated with commas) which acts as the central registry for volumes

- **quobyte.volume** (string), required

volume is a string that references an already created Quobyte volume by name.

- **quobyte.group** (string)

group to map volume access to Default is no group

- **quobyte.readOnly** (boolean)

readOnly here will force the Quobyte volume to be mounted with read-only permissions. Defaults to false.

- **quobyte.tenant** (string)

tenant owning the given Quobyte volume in the Backend Used with dynamically provisioned Quobyte volumes, value is set by the plugin

- **quobyte.user** (string)

user to map volume access to Defaults to serviceaccount user

- **rbd** (RBDVolumeSource)

rbd represents a Rados Block Device mount on the host that shares a pod's lifetime. More info: <https://examples.k8s.io/volumes/rbd/README.md>

Represents a Rados Block Device mount that lasts the lifetime of a pod. RBD volumes support ownership management and SELinux relabeling.

- **rbd.image** (string), required

image is the rados image name. More info: <https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it>

- **rbd.monitors** ([]string), required

monitors is a collection of Ceph monitors. More info: <https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it>

- **rbd.fsType** (string)

fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: <https://kubernetes.io/docs/concepts/storage/volumes#rbd>

- **rbd.keyring** (string)

keyring is the path to key ring for RBDUser. Default is /etc/ceph/keyring. More info: <https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it>

- **rbd.pool** (string)

pool is the rados pool name. Default is rbd. More info: <https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it>

- **rbd.readOnly** (boolean)

readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false. More info: <https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it>

- **rbd.secretRef** ([LocalObjectReference](#))

secretRef is name of the authentication secret for RBDUser. If provided overrides keyring. Default is nil. More info: <https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it>

- **rbd.user** (string)

user is the rados user name. Default is admin. More info: <https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it>

- **scaleIO** (ScaleIOVolumeSource)

scaleIO represents a ScaleIO persistent volume attached and mounted on Kubernetes nodes.

ScaleIOVolumeSource represents a persistent ScaleIO volume

- **scaleIO.gateway** (string), required

gateway is the host address of the ScaleIO API Gateway.

- **scaleIO.secretRef** ([LocalObjectReference](#)), required

secretRef references to the secret for ScaleIO user and other sensitive information. If this is not provided, Login operation will fail.

- **scaleIO.system** (string), required

system is the name of the storage system as configured in ScaleIO.

- **scaleIO.fsType** (string)

fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Default is "xfs".

- **scaleIO.protectionDomain** (string)

protectionDomain is the name of the ScaleIO Protection Domain for the configured storage.

- **scaleIO.readOnly** (boolean)

readOnly Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

- **scaleIO.sslEnabled** (boolean)

sslEnabled Flag enable/disable SSL communication with

Gateway, default false

- **scaleIO.storageMode** (string)

storageMode indicates whether the storage for a volume should be ThickProvisioned or ThinProvisioned. Default is ThinProvisioned.

- **scaleIO.storagePool** (string)

storagePool is the ScaleIO Storage Pool associated with the protection domain.

- **scaleIO.volumeName** (string)

volumeName is the name of a volume already created in the ScaleIO system that is associated with this volume source.

- **storageos** (StorageOSVolumeSource)

storageOS represents a StorageOS volume attached and mounted on Kubernetes nodes.

Represents a StorageOS persistent volume resource.

- **storageos.fsType** (string)

fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.

- **storageos.readOnly** (boolean)

readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

- **storageos.secretRef** ([LocalObjectReference](#))

secretRef specifies the secret to use for obtaining the StorageOS API credentials. If not specified, default values will be attempted.

- **storageos.volumeName** (string)

volumeName is the human-readable name of the StorageOS volume. Volume names are only unique within a namespace.

- **storageos.volumeNamespace** (string)

volumeNamespace specifies the scope of the volume within StorageOS. If no namespace is specified then the Pod's namespace will be used. This allows the Kubernetes name scoping to be mirrored within StorageOS for tighter integration. Set VolumeName to any name to override the default behaviour. Set to "default" if you are not using namespaces within StorageOS. Namespaces that do not pre-exist within StorageOS will be created.

- **vsphereVolume** (VsphereVirtualDiskVolumeSource)

vsphereVolume represents a vSphere volume attached and mounted on kubelets host machine

Represents a vSphere volume resource.

- **vsphereVolume.volumePath** (string), required

volumePath is the path that identifies vSphere volume vmdk

- **vsphereVolume.fsType** (string)

fsType is filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.

- **vsphereVolume.storagePolicyID** (string)

storagePolicyID is the storage Policy Based Management (SPBM) profile ID associated with the StoragePolicyName.

- **vsphereVolume.storagePolicyName** (string)

storagePolicyName is the storage Policy Based Management (SPBM) profile name.

Deprecated

- **gitRepo** (GitRepoVolumeSource)

gitRepo represents a git repository at a particular revision.

DEPRECATED: GitRepo is deprecated. To provision a container with a git repo, mount an EmptyDir into an InitContainer that clones the repo using git, then mount the EmptyDir into the Pod's container.

*Represents a volume that is populated with the contents of a git repository. Git repo volumes do not support ownership management. Git repo volumes support SELinux relabeling.

DEPRECATED: GitRepo is deprecated. To provision a container with a git repo, mount an EmptyDir into an InitContainer that clones the repo using git, then mount the EmptyDir into the Pod's container.*

- **gitRepo.repository** (string), required

repository is the URL

- **gitRepo.directory** (string)

directory is the target directory name. Must not contain or start with '..'. If '.' is supplied, the volume directory will be the git repository. Otherwise, if specified, the volume will contain the git repository in the subdirectory with the given name.

- **gitRepo.revision** (string)

revision is the commit hash for the specified revision.

DownwardAPIVolumeFile

DownwardAPIVolumeFile represents information to create the file containing the pod field

- **path** (string), required

Required: Path is the relative path name of the file to be created.

Must not be absolute or contain the '..' path. Must be utf-8 encoded.

The first item of the relative path must not start with '..'

- **fieldRef** ([ObjectFieldSelector](#))

Required: Selects a field of the pod: only annotations, labels, name and namespace are supported.

- **mode** (int32)

Optional: mode bits used to set permissions on this file, must be an octal value between 0000 and 0777 or a decimal value between 0

and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

- **resourceFieldRef** ([ResourceFieldSelector](#))

Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, requests.cpu and requests.memory) are currently supported.

KeyToPath

Maps a string key to a path within a volume.

- **key** (string), required

key is the key to project.

- **path** (string), required

path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

- **mode** (int32)

mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

5.3.4 - PersistentVolumeClaim

PersistentVolumeClaim is a user's request for and claim to a persistent volume.

```
apiVersion: v1
import "k8s.io/api/core/v1"
```

PersistentVolumeClaim

PersistentVolumeClaim is a user's request for and claim to a persistent volume

- **apiVersion**: v1
- **kind**: PersistentVolumeClaim
- **metadata** ([ObjectMeta](#))

Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **spec** ([PersistentVolumeClaimSpec](#))

spec defines the desired characteristics of a volume requested by a pod author. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims>

- **status** ([PersistentVolumeClaimStatus](#))

status represents the current information/status of a persistent volume claim. Read-only. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims>

PersistentVolumeClaimSpec

PersistentVolumeClaimSpec describes the common attributes of storage devices and allows a Source for provider-specific attributes

- **accessModes** ([]string)

accessModes contains the desired access modes the volume should have. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1>

- **selector** ([LabelSelector](#))

selector is a label query over volumes to consider for binding.

- **resources** ([ResourceRequirements](#))

resources represents the minimum resources the volume should have. If RecoverVolumeExpansionFailure feature is enabled users are allowed to specify resource requirements that are lower than previous value but must still be higher than capacity recorded in the status field of the claim. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources>

ResourceRequirements describes the compute resource requirements.

- **resources.claims** ([]ResourceClaim)

Map: unique values on key name will be kept during a merge

Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container.

This is an alpha field and requires enabling the DynamicResourceAllocation feature gate.

This field is immutable. It can only be set for containers.

ResourceClaim references one entry in PodSpec.ResourceClaims.

- **resources.claims.name** (string), required

Name must match the name of one entry in pod.spec.resourceClaims of the Pod where this field is used. It makes that resource available inside a container.

- **resources.limits** (map[string][Quantity](#))

Limits describes the maximum amount of compute resources allowed. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

- **resources.requests** (map[string][Quantity](#))

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

- **volumeName** (string)

volumeName is the binding reference to the PersistentVolume backing this claim.

- **storageClassName** (string)

storageClassName is the name of the StorageClass required by the claim. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#class-1>

- **volumeMode** (string)

volumeMode defines what type of volume is required by the claim. Value of Filesystem is implied when not included in claim spec.

Beta level

- **dataSource** ([TypedLocalObjectReference](#))

dataSource field can be used to specify either:
* An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot)
* An existing PVC (PersistentVolumeClaim) If the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If the namespace is specified, then dataSourceRef will not be copied to dataSource.

- **dataSourceRef** (TypedObjectReference)

dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the dataSource field and as such if both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same value and must be empty. There are three important differences between dataSource and dataSourceRef: * While dataSource only allows two specific types of objects, dataSourceRef allows any non-core object, as well as PersistentVolumeClaim objects.

- While dataSource ignores disallowed values (dropping them), dataSourceRef preserves all values, and generates an error if a disallowed value is specified.
- While dataSource only allows local objects, dataSourceRef allows objects in any namespaces. (Beta) Using this field requires the AnyVolumeDataSource feature gate to be enabled. (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

**

- **dataSourceRef.kind** (string), required

Kind is the type of resource being referenced

- **dataSourceRef.name** (string), required

Name is the name of resource being referenced

- **dataSourceRef.apiGroup** (string)

APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.

- **dataSourceRef.namespace** (string)

Namespace is the namespace of resource being referenced

Note that when a namespace is specified, a gateway.networking.k8s.io/ReferenceGrant object is required in the referent namespace to allow that namespace's owner to accept the reference. See the ReferenceGrant documentation for details. (Alpha) This field requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

PersistentVolumeClaimStatus

PersistentVolumeClaimStatus is the current status of a persistent volume claim.

- **accessModes** ([]string)

accessModes contains the actual access modes the volume backing the PVC has. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1>

- **allocatedResourceStatuses** (map[string]string)

allocatedResourceStatuses stores status of resource being resized for the given PVC. Key names follow standard Kubernetes label syntax. Valid values are either: * Un-prefixed keys: - storage - the capacity of the volume. * Custom resources must use implementation-defined prefixed names such as "example.com/my-custom-resource" Apart from above values - keys that are unprefixed or have kubernetes.io prefix are considered reserved and hence may not be used.

ClaimResourceStatus can be in any of following states: - ControllerResizeInProgress: State set when resize controller starts resizing the volume in control-plane. - ControllerResizeFailed: State set when resize has failed in resize controller with a terminal error. - NodeResizePending: State set when resize controller has finished resizing the volume but further resizing of volume is needed on the node. - NodeResizeInProgress: State set when kubelet starts resizing the volume. - NodeResizeFailed: State set when resizing has failed in kubelet with a terminal error. Transient errors don't set NodeResizeFailed. For example: if expanding a PVC for more capacity - this field can be one of the following states: - pvc.status.allocatedResourceStatus['storage'] = "ControllerResizeInProgress" - pvc.status.allocatedResourceStatus['storage'] = "ControllerResizeFailed" - pvc.status.allocatedResourceStatus['storage'] = "NodeResizePending" - pvc.status.allocatedResourceStatus['storage'] = "NodeResizeInProgress" - pvc.status.allocatedResourceStatus['storage'] = "NodeResizeFailed" When this field is not set, it means that no resize operation is in progress for the given PVC.

A controller that receives PVC update with previously unknown resourceName or ClaimResourceStatus should ignore the update for the purpose it was designed. For example - a controller that only is responsible for resizing capacity of the volume, should ignore PVC updates that change other valid resources associated with PVC.

This is an alpha field and requires enabling RecoverVolumeExpansionFailure feature.

- **allocatedResources** (map[string][Quantity](#))

allocatedResources tracks the resources allocated to a PVC including its capacity. Key names follow standard Kubernetes label syntax. Valid values are either: * Un-prefixed keys: - storage - the capacity of the volume. * Custom resources must use implementation-defined prefixed names such as "example.com/my-custom-resource" Apart from above values - keys that are unprefixed or have kubernetes.io prefix are considered reserved and hence may not be used.

Capacity reported here may be larger than the actual capacity when a volume expansion operation is requested. For storage quota, the larger value from allocatedResources and PVC.spec.resources is used. If allocatedResources is not set, PVC.spec.resources alone is used for quota calculation. If a volume expansion capacity request is lowered, allocatedResources is only lowered if there are no

expansion operations in progress and if the actual volume capacity is equal or lower than the requested capacity.

A controller that receives PVC update with previously unknown resourceName should ignore the update for the purpose it was designed. For example - a controller that only is responsible for resizing capacity of the volume, should ignore PVC updates that change other valid resources associated with PVC.

This is an alpha field and requires enabling RecoverVolumeExpansionFailure feature.

- **capacity** (map[string][Quantity](#))

capacity represents the actual resources of the underlying volume.

- **conditions** ([]PersistentVolumeClaimCondition)

Patch strategy: merge on key type

conditions is the current Condition of persistent volume claim. If underlying persistent volume is being resized then the Condition will be set to 'ResizeStarted'.

PersistentVolumeClaimCondition contains details about state of pvc

- **conditions.status** (string), required
- **conditions.type** (string), required
- **conditions.lastProbeTime** (Time)

lastProbeTime is the time we probed the condition.

Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

- **conditions.lastTransitionTime** (Time)

lastTransitionTime is the time the condition transitioned from one status to another.

Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

- **conditions.message** (string)

message is the human-readable message indicating details about last transition.

- **conditions.reason** (string)

reason is a unique, this should be a short, machine understandable string that gives the reason for condition's last transition. If it reports "ResizeStarted" that means the underlying persistent volume is being resized.

- **phase** (string)

phase represents the current phase of PersistentVolumeClaim.

PersistentVolumeClaimList

PersistentVolumeClaimList is a list of PersistentVolumeClaim items.

- **apiVersion**: v1
- **kind**: PersistentVolumeClaimList
- **metadata** ([ListMeta](#))

Standard list metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds>

- **items** ([] [PersistentVolumeClaim](#)), required

items is a list of persistent volume claims. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims>

Operations

get read the specified PersistentVolumeClaim

HTTP Request

GET /api/v1/namespaces/{namespace}/persistentvolumeclaims/{name}

Parameters

- **name** (*in path*): string, required
name of the PersistentVolumeClaim
- **namespace** (*in path*): string, required
[namespace](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([PersistentVolumeClaim](#)): OK

401: Unauthorized

get read status of the specified PersistentVolumeClaim

HTTP Request

GET /api/v1/namespaces/{namespace}/persistentvolumeclaims/{name}/status

Parameters

- **name** (*in path*): string, required
name of the PersistentVolumeClaim
- **namespace** (*in path*): string, required
[namespace](#)
- **pretty** (*in query*): string

[pretty](#)

Response

200 ([PersistentVolumeClaim](#)): OK

401: Unauthorized

list list or watch objects of kind PersistentVolumeClaim

HTTP Request

GET /api/v1/namespaces/{namespace}/persistentvolumeclaims

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([PersistentVolumeClaimList](#)): OK

401: Unauthorized

list list or watch objects of kind PersistentVolumeClaim

HTTP Request

GET /api/v1/persistentvolumeclaims

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([PersistentVolumeClaimList](#)): OK

401: Unauthorized

create create a PersistentVolumeClaim

HTTP Request

POST /api/v1/namespaces/{namespace}/persistentvolumeclaims

Parameters

- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [PersistentVolumeClaim](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([PersistentVolumeClaim](#)): OK
201 ([PersistentVolumeClaim](#)): Created
202 ([PersistentVolumeClaim](#)): Accepted
401: Unauthorized

update replace the specified PersistentVolumeClaim

HTTP Request

PUT /api/v1/namespaces/{namespace}/persistentvolumeclaims/{name}

Parameters

- **name** (*in path*): string, required
name of the PersistentVolumeClaim
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [PersistentVolumeClaim](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([PersistentVolumeClaim](#)): OK

201 ([PersistentVolumeClaim](#)): Created

401: Unauthorized

update replace status of the specified PersistentVolumeClaim

HTTP Request

PUT /api/v1/namespaces/{namespace}/persistentvolumeclaims/{name}/status

Parameters

- **name** (*in path*): string, required
name of the PersistentVolumeClaim
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [PersistentVolumeClaim](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([PersistentVolumeClaim](#)): OK

201 ([PersistentVolumeClaim](#)): Created

401: Unauthorized

patch partially update the specified PersistentVolumeClaim

HTTP Request

PATCH /api/v1/namespaces/{namespace}/persistentvolumeclaims/{name}

Parameters

- **name** (*in path*): string, required
name of the PersistentVolumeClaim
- **namespace** (*in path*): string, required
[namespace](#)

- **body**: [Patch](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **force** (*in query*): boolean
[force](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([PersistentVolumeClaim](#)): OK
201 ([PersistentVolumeClaim](#)): Created
401: Unauthorized

patch partially update status of the specified PersistentVolumeClaim

HTTP Request

PATCH /api/v1/namespaces/{namespace}/persistentvolumeclaims/{name}/status

Parameters

- **name** (*in path*): string, required
name of the PersistentVolumeClaim
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [Patch](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **force** (*in query*): boolean
[force](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([PersistentVolumeClaim](#)): OK

201 ([PersistentVolumeClaim](#)): Created

401: Unauthorized

delete delete a PersistentVolumeClaim

HTTP Request

DELETE /api/v1/namespaces/{namespace}/persistentvolumeclaims/{name}

Parameters

- **name** (*in path*): string, required
[name](#)
name of the PersistentVolumeClaim
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [DeleteOptions](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)

Response

200 ([PersistentVolumeClaim](#)): OK

202 ([PersistentVolumeClaim](#)): Accepted

401: Unauthorized

deletecollection delete collection of PersistentVolumeClaim

HTTP Request

DELETE /api/v1/namespaces/{namespace}/persistentvolumeclaims

Parameters

- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [DeleteOptions](#)

- **continue** (*in query*): string

[continue](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.3.5 - PersistentVolume

PersistentVolume (PV) is a storage resource provisioned by an administrator.

```
apiVersion: v1

import "k8s.io/api/core/v1"
```

PersistentVolume

PersistentVolume (PV) is a storage resource provisioned by an administrator. It is analogous to a node. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes>

- **apiVersion**: v1
- **kind**: PersistentVolume
- **metadata** ([ObjectMeta](#))

Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **spec** ([PersistentVolumeSpec](#))

spec defines a specification of a persistent volume owned by the cluster. Provisioned by an administrator. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistent-volumes>

- **status** ([PersistentVolumeStatus](#))

status represents the current information/status for the persistent volume. Populated by the system. Read-only. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistent-volumes>

PersistentVolumeSpec

PersistentVolumeSpec is the specification of a persistent volume.

- **accessModes** ([]string)

accessModes contains all ways the volume can be mounted. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes>

- **capacity** (map[string][Quantity](#))

capacity is the description of the persistent volume's resources and capacity. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#capacity>

- **claimRef** ([ObjectReference](#))

claimRef is part of a bi-directional binding between PersistentVolume and PersistentVolumeClaim. Expected to be non-nil when bound. claim.VolumeName is the authoritative bind between PV and PVC. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#binding>

- **mountOptions** ([]string)

mountOptions is the list of mount options, e.g. ["ro", "soft"]. Not validated - mount will simply fail if one is invalid. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes/#mount-options>

- **nodeAffinity** (VolumeNodeAffinity)

nodeAffinity defines constraints that limit what nodes this volume can be accessed from. This field influences the scheduling of pods that use this volume.

VolumeNodeAffinity defines constraints that limit what nodes this volume can be accessed from.

- **nodeAffinity.required** (NodeSelector)

required specifies hard node constraints that must be met.

A node selector represents the union of the results of one or more label queries over a set of nodes; that is, it represents the OR of the selectors represented by the node selector terms.

- **nodeAffinity.required.nodeSelectorTerms**

([]NodeSelectorTerm), required

Required. A list of node selector terms. The terms are ORed.

A null or empty node selector term matches no objects. The requirements of them are ANDed. The TopologySelectorTerm type implements a subset of the NodeSelectorTerm.

- **nodeAffinity.required.nodeSelectorTerms.matchExpressions**

([]NodeSelectorRequirement)

A list of node selector requirements by node's labels.

- **nodeAffinity.required.nodeSelectorTerms.matchFields**

([]NodeSelectorRequirement)

A list of node selector requirements by node's fields.

- **persistentVolumeReclaimPolicy** (string)

persistentVolumeReclaimPolicy defines what happens to a persistent volume when released from its claim. Valid options are Retain (default for manually created PersistentVolumes), Delete (default for dynamically provisioned PersistentVolumes), and Recycle (deprecated). Recycle must be supported by the volume plugin underlying this PersistentVolume. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#reclaiming>

- **storageClassName** (string)

storageClassName is the name of StorageClass to which this persistent volume belongs. Empty value means that this volume does not belong to any StorageClass.

- **volumeMode** (string)

volumeMode defines if a volume is intended to be used with a formatted filesystem or to remain in raw block state. Value of Filesystem is implied when not included in spec.

Local

- **hostPath** (HostPathVolumeSource)

hostPath represents a directory on the host. Provisioned by a developer or tester. This is useful for single-node development and testing only! On-host storage is not supported in any way and WILL NOT WORK in a multi-node cluster. More info: <https://kubernetes.io/docs/concepts/storage/volumes#hostpath>

Represents a host path mapped into a pod. Host path volumes do not support ownership management or SELinux relabeling.

- **hostPath.path** (string), required

path of the directory on the host. If the path is a symlink, it will follow the link to the real path. More info: <https://kubernetes.io/docs/concepts/storage/volumes#hostpath>

- **hostPath.type** (string)

type for HostPath Volume Defaults to "" More info: <https://kubernetes.io/docs/concepts/storage/volumes#hostpath>

- **local** (LocalVolumeSource)

local represents directly-attached storage with node affinity

Local represents directly-attached storage with node affinity (Beta feature)

- **local.path** (string), required

path of the full path to the volume on the node. It can be either a directory or block device (disk, partition, ...).

- **local.fsType** (string)

fsType is the filesystem type to mount. It applies only when the Path is a block device. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". The default value is to auto-select a filesystem if unspecified.

Persistent volumes

- **awsElasticBlockStore** (AWSElasticBlockStoreVolumeSource)

awsElasticBlockStore represents an AWS Disk resource that is

attached to a kubelet's host machine and then exposed to the pod.

More info: <https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore>

*Represents a Persistent Disk resource in AWS.

An AWS EBS disk must exist before mounting to a container. The disk must also be in the same AWS zone as the kubelet. An AWS EBS disk can only be mounted as read/write once. AWS EBS volumes support ownership management and SELinux relabeling.*

- **awsElasticBlockStore.volumeID** (string), required

volumeID is unique ID of the persistent disk resource in AWS (Amazon EBS volume). More info: <https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore>

- **awsElasticBlockStore.fsType** (string)

fsType is the filesystem type of the volume that you want to

mount. Tip: Ensure that the filesystem type is supported by

the host operating system. Examples: "ext4", "xfs", "ntfs".

Implicitly inferred to be "ext4" if unspecified. More info:

<https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore>

- **awsElasticBlockStore.partition** (int32)

partition is the partition in the volume that you want to mount.

If omitted, the default is to mount by volume name. Examples:

For volume /dev/sda1, you specify the partition as "1".

Similarly, the volume partition for /dev/sda is "0" (or you can leave the property empty).

- **awsElasticBlockStore.readOnly** (boolean)

readOnly value true will force the readOnly setting in

VolumeMounts. More info: <https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore>

- **azureDisk** (AzureDiskVolumeSource)

azureDisk represents an Azure Data Disk mount on the host and bind mount to the pod.

AzureDisk represents an Azure Data Disk mount on the host and bind mount to the pod.

- **azureDisk.diskName** (string), required

diskName is the Name of the data disk in the blob storage

- **azureDisk.diskURI** (string), required

diskURI is the URI of data disk in the blob storage

- **azureDisk.cachingMode** (string)

cachingMode is the Host Caching mode: None, Read Only, Read Write.

- **azureDisk.fsType** (string)

fsType is Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.

- **azureDisk.kind** (string)

kind expected values are Shared: multiple blob disks per storage account Dedicated: single blob disk per storage account Managed: azure managed data disk (only in managed availability set). defaults to shared

- **azureDisk.readOnly** (boolean)

readOnly Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

- **azureFile** (AzureFilePersistentVolumeSource)

azureFile represents an Azure File Service mount on the host and bind mount to the pod.

AzureFile represents an Azure File Service mount on the host and bind mount to the pod.

- **azureFile.secretName** (string), required

secretName is the name of secret that contains Azure Storage

Account Name and Key

- **azureFile.shareName** (string), required

shareName is the azure Share Name

- **azureFile.readOnly** (boolean)

readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

- **azureFile.secretNamespace** (string)

secretNamespace is the namespace of the secret that contains Azure Storage Account Name and Key default is the same as the Pod

- **cephfs** (CephFSPersistentVolumeSource)

cephFS represents a Ceph FS mount on the host that shares a pod's lifetime

*Represents a Ceph Filesystem mount that lasts the lifetime of a pod
Cephfs volumes do not support ownership management or SELinux
relabeling.*

- **cephfs.monitors** ([]string), required

monitors is Required: Monitors is a collection of Ceph monitors More info: <https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it>

- **cephfs.path** (string)

path is Optional: Used as the mounted root, rather than the full Ceph tree, default is /

- **cephfs.readOnly** (boolean)

readOnly is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts. More info: <https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it>

- **cephfs.secretFile** (string)

secretFile is Optional: SecretFile is the path to key ring for User, default is /etc/ceph/user.secret More info: <https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it>

- **cephfs.secretRef** (SecretReference)

secretRef is Optional: SecretRef is reference to the authentication secret for User, default is empty. More info: <https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it>

SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace

- **cephfs.secretRef.name** (string)

name is unique within a namespace to reference a secret resource.

- **cephfs.secretRef.namespace** (string)

namespace defines the space within which the secret name must be unique.

- **cephfs.user** (string)

user is Optional: User is the rados user name, default is admin

More info: <https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it>

- **cinder** (CinderPersistentVolumeSource)

cinder represents a cinder volume attached and mounted on kubelets host machine. More info: <https://examples.k8s.io/mysql-cinder-pd/README.md>

Represents a cinder volume resource in Openstack. A Cinder volume must exist before mounting to a container. The volume must also be in the same region as the kubelet. Cinder volumes support ownership management and SELinux relabeling.

- **cinder.volumeID** (string), required

volumeID used to identify the volume in cinder. More info:

<https://examples.k8s.io/mysql-cinder-pd/README.md>

- **cinder.fsType** (string)

fsType Filesystem type to mount. Must be a filesystem type supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: <https://examples.k8s.io/mysql-cinder-pd/README.md>

- **cinder.readOnly** (boolean)

readOnly is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts. More info: <https://examples.k8s.io/mysql-cinder-pd/README.md>

- **cinder.secretRef** (SecretReference)

secretRef is Optional: points to a secret object containing parameters used to connect to OpenStack.

SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace

- **cinder.secretRef.name** (string)

name is unique within a namespace to reference a secret resource.

- **cinder.secretRef.namespace** (string)

namespace defines the space within which the secret name must be unique.

- **csi** (CSIPersistentVolumeSource)

csi represents storage that is handled by an external CSI driver (Beta feature).

Represents storage that is managed by an external CSI volume driver (Beta feature)

- **csi.driver** (string), required

driver is the name of the driver to use for this volume.

Required.

- **csi.volumeHandle** (string), required

volumeHandle is the unique volume name returned by the CSI volume plugin's CreateVolume to refer to the volume on all

subsequent calls. Required.

- **csi.controllerExpandSecretRef** (SecretReference)

controllerExpandSecretRef is a reference to the secret object containing sensitive information to pass to the CSI driver to complete the CSI ControllerExpandVolume call. This field is optional, and may be empty if no secret is required. If the secret object contains more than one secret, all secrets are passed.

SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace

- **csi.controllerExpandSecretRef.name** (string)

name is unique within a namespace to reference a secret resource.

- **csi.controllerExpandSecretRef.namespace** (string)

namespace defines the space within which the secret name must be unique.

- **csi.controllerPublishSecretRef** (SecretReference)

controllerPublishSecretRef is a reference to the secret object containing sensitive information to pass to the CSI driver to complete the CSI ControllerPublishVolume and ControllerUnpublishVolume calls. This field is optional, and may be empty if no secret is required. If the secret object contains more than one secret, all secrets are passed.

SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace

- **csi.controllerPublishSecretRef.name** (string)

name is unique within a namespace to reference a secret resource.

- **csi.controllerPublishSecretRef.namespace** (string)

namespace defines the space within which the secret name must be unique.

- **csi.fsType** (string)

fsType to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs".

- **csi.nodeExpandSecretRef** (SecretReference)

nodeExpandSecretRef is a reference to the secret object containing sensitive information to pass to the CSI driver to complete the CSI NodeExpandVolume call. This is a beta field which is enabled default by CSINodeExpandSecret feature gate. This field is optional, may be omitted if no secret is required. If the secret object contains more than one secret, all secrets are passed.

SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace

- **csi.nodeExpandSecretRef.name** (string)

name is unique within a namespace to reference a secret resource.

- **csi.nodeExpandSecretRef.namespace** (string)

namespace defines the space within which the secret name must be unique.

- **csi.nodePublishSecretRef** (SecretReference)

nodePublishSecretRef is a reference to the secret object containing sensitive information to pass to the CSI driver to complete the CSI NodePublishVolume and NodeUnpublishVolume calls. This field is optional, and may be empty if no secret is required. If the secret object contains more than one secret, all secrets are passed.

SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace

- **csi.nodePublishSecretRef.name** (string)

name is unique within a namespace to reference a secret resource.

- **csi.nodePublishSecretRef.namespace** (string)

namespace defines the space within which the secret name must be unique.

- **csi.nodeStageSecretRef** (SecretReference)

nodeStageSecretRef is a reference to the secret object containing sensitive information to pass to the CSI driver to complete the CSI NodeStageVolume and NodeStageVolume and NodeUnstageVolume calls. This field is optional, and may be empty if no secret is required. If the secret object contains more than one secret, all secrets are passed.

SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace

- **csi.nodeStageSecretRef.name** (string)

name is unique within a namespace to reference a secret resource.

- **csi.nodeStageSecretRef.namespace** (string)

namespace defines the space within which the secret name must be unique.

- **csi.readOnly** (boolean)

readOnly value to pass to ControllerPublishVolumeRequest. Defaults to false (read/write).

- **csi.volumeAttributes** (map[string]string)

volumeAttributes of the volume to publish.

- **fc** (FCVolumeSource)

fc represents a Fibre Channel resource that is attached to a kubelet's host machine and then exposed to the pod.

Represents a Fibre Channel volume. Fibre Channel volumes can only be mounted as read/write once. Fibre Channel volumes support ownership management and SELinux relabeling.

- **fc.fsType** (string)

fsType is the filesystem type to mount. Must be a filesystem

type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.

- o **fc.lun** (int32)

lun is Optional: FC target lun number

- o **fc.readOnly** (boolean)

readOnly is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

- o **fc.targetWWNs** ([]string)

targetWWNs is Optional: FC target worldwide names (WWNs)

- o **fc.wwids** ([]string)

wwids Optional: FC volume world wide identifiers (wwids)

Either wwid or combination of targetWWNs and lun must be set, but not both simultaneously.

- **flexVolume** (FlexPersistentVolumeSource)

flexVolume represents a generic volume resource that is provisioned/attached using an exec based plugin.

FlexPersistentVolumeSource represents a generic persistent volume resource that is provisioned/attached using an exec based plugin.

- o **flexVolume.driver** (string), required

driver is the name of the driver to use for this volume.

- o **flexVolume.fsType** (string)

fsType is the Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". The default filesystem depends on FlexVolume script.

- o **flexVolume.options** (map[string]string)

options is Optional: this field holds extra command options if any.

- o **flexVolume.readOnly** (boolean)

readOnly is Optional: defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

- o **flexVolume.secretRef** (SecretReference)

secretRef is Optional: SecretRef is reference to the secret object containing sensitive information to pass to the plugin scripts. This may be empty if no secret object is specified. If the secret object contains more than one secret, all secrets are passed to the plugin scripts.

SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace

- **flexVolume.secretRef.name** (string)

name is unique within a namespace to reference a secret resource.

- **flexVolume.secretRef.namespace** (string)

namespace defines the space within which the secret name must be unique.

- **flocker** (FlockerVolumeSource)

flocker represents a Flocker volume attached to a kubelet's host machine and exposed to the pod for its usage. This depends on the Flocker control service being running

Represents a Flocker volume mounted by the Flocker agent. One and only one of datasetName and datasetUUID should be set. Flocker volumes do not support ownership management or SELinux relabeling.

- **flocker.datasetName** (string)

datasetName is Name of the dataset stored as metadata -> name on the dataset for Flocker should be considered as deprecated

- **flocker.datasetUUID** (string)

datasetUUID is the UUID of the dataset. This is unique identifier of a Flocker dataset

- **gcePersistentDisk** (GCEPersistentDiskVolumeSource)

gcePersistentDisk represents a GCE Disk resource that is attached to a kubelet's host machine and then exposed to the pod.

Provisioned by an admin. More info: <https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk>

*Represents a Persistent Disk resource in Google Compute Engine.

A GCE PD must exist before mounting to a container. The disk must also be in the same GCE project and zone as the kubelet. A GCE PD can only be mounted as read/write once or read-only many times. GCE PDs support ownership management and SELinux relabeling.*

- **gcePersistentDisk.pdName** (string), required

pdName is unique name of the PD resource in GCE. Used to identify the disk in GCE. More info: <https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk>

- **gcePersistentDisk.fsType** (string)

fsType is filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs".

Implicitly inferred to be "ext4" if unspecified. More info:

<https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk>

- **gcePersistentDisk.partition** (int32)

partition is the partition in the volume that you want to mount. If omitted, the default is to mount by volume name. Examples:

For volume /dev/sda1, you specify the partition as "1".

Similarly, the volume partition for /dev/sda is "0" (or you can leave the property empty). More info: <https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk>

- **gcePersistentDisk.readOnly** (boolean)

readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false. More info: <https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk>

- **glusterfs** (GlusterfsPersistentVolumeSource)

glusterfs represents a Glusterfs volume that is attached to a host and exposed to the pod. Provisioned by an admin. More info: <https://examples.k8s.io/volumes/glusterfs/README.md>

Represents a Glusterfs mount that lasts the lifetime of a pod. Glusterfs volumes do not support ownership management or SELinux relabeling.

- **glusterfs.endpoints** (string), required

endpoints is the endpoint name that details Glusterfs topology. More info: <https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod>

- **glusterfs.path** (string), required

path is the Glusterfs volume path. More info: <https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod>

- **glusterfs.endpointsNamespace** (string)

endpointsNamespace is the namespace that contains Glusterfs endpoint. If this field is empty, the EndpointNamespace defaults to the same namespace as the bound PVC. More info: <https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod>

- **glusterfs.readOnly** (boolean)

readOnly here will force the Glusterfs volume to be mounted with read-only permissions. Defaults to false. More info: <https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod>

- **iscsi** (ISCSIPersistentVolumeSource)

iscsi represents an iSCSI Disk resource that is attached to a kubelet's host machine and then exposed to the pod. Provisioned by an admin.

ISCSIPersistentVolumeSource represents an iSCSI disk. iSCSI volumes can only be mounted as read/write once. iSCSI volumes support ownership management and SELinux relabeling.

- **iscsi.iqn** (string), required

iqn is Target iSCSI Qualified Name.

- **iscsi.lun** (int32), required

lun is iSCSI Target Lun number.

- **iscsi.targetPortal** (string), required

targetPortal is iSCSI Target Portal. The Portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

- **iscsi.chapAuthDiscovery** (boolean)

chapAuthDiscovery defines whether support iSCSI Discovery CHAP authentication

- **iscsi.chapAuthSession** (boolean)

chapAuthSession defines whether support iSCSI Session CHAP authentication

- **iscsi.fsType** (string)

fsType is the filesystem type of the volume that you want to

mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs".

Implicitly inferred to be "ext4" if unspecified. More info:

<https://kubernetes.io/docs/concepts/storage/volumes#iscsi>

- **iscsi.initiatorName** (string)

initiatorName is the custom iSCSI Initiator Name. If initiatorName is specified with iscsiInterface simultaneously, new iSCSI interface <target portal>:<volume name> will be created for the connection.

- **iscsi.iscsiInterface** (string)

iscsiInterface is the interface Name that uses an iSCSI transport. Defaults to 'default' (tcp).

- **iscsi.portals** ([]string)

portals is the iSCSI Target Portal List. The Portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

- **iscsi.readOnly** (boolean)

readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false.

- **iscsi.secretRef** (SecretReference)

secretRef is the CHAP Secret for iSCSI target and initiator authentication

SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace

- **iscsi.secretRef.name** (string)

name is unique within a namespace to reference a secret resource.

- **iscsi.secretRef.namespace** (string)

namespace defines the space within which the secret name must be unique.

- **nfs** (NFSVolumeSource)

nfs represents an NFS mount on the host. Provisioned by an admin.

More info: <https://kubernetes.io/docs/concepts/storage/volumes#nfs>

Represents an NFS mount that lasts the lifetime of a pod. NFS volumes do not support ownership management or SELinux relabeling.

- **nfs.path** (string), required

path that is exported by the NFS server. More info: <https://kubernetes.io/docs/concepts/storage/volumes#nfs>

- **nfs.server** (string), required

server is the hostname or IP address of the NFS server. More info: <https://kubernetes.io/docs/concepts/storage/volumes#nfs>

- **nfs.readOnly** (boolean)

readOnly here will force the NFS export to be mounted with read-only permissions. Defaults to false. More info: <https://kubernetes.io/docs/concepts/storage/volumes#nfs>

kubernetes.io/docs/concepts/storage/volumes#nfs

- **photonPersistentDisk** (PhotonPersistentDiskVolumeSource)

photonPersistentDisk represents a PhotonController persistent disk attached and mounted on kubelets host machine

Represents a Photon Controller persistent disk resource.

- **photonPersistentDisk.pdID** (string), required

pdID is the ID that identifies Photon Controller persistent disk

- **photonPersistentDisk.fsType** (string)

fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.

- **portworxVolume** (PortworxVolumeSource)

portworxVolume represents a portworx volume attached and mounted on kubelets host machine

PortworxVolumeSource represents a Portworx volume resource.

- **portworxVolume.volumeID** (string), required

volumeID uniquely identifies a Portworx volume

- **portworxVolume.fsType** (string)

fSType represents the filesystem type to mount Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs". Implicitly inferred to be "ext4" if unspecified.

- **portworxVolume.readOnly** (boolean)

readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

- **quobyte** (QuobyteVolumeSource)

quobyte represents a Quobyte mount on the host that shares a pod's lifetime

Represents a Quobyte mount that lasts the lifetime of a pod. Quobyte volumes do not support ownership management or SELinux relabeling.

- **quobyte.registry** (string), required

registry represents a single or multiple Quobyte Registry services specified as a string as host:port pair (multiple entries are separated with commas) which acts as the central registry for volumes

- **quobyte.volume** (string), required

volume is a string that references an already created Quobyte volume by name.

- **quobyte.group** (string)

group to map volume access to Default is no group

- **quobyte.readOnly** (boolean)

readOnly here will force the Quobyte volume to be mounted with read-only permissions. Defaults to false.

- **quobyte.tenant** (string)

tenant owning the given Quobyte volume in the Backend Used with dynamically provisioned Quobyte volumes, value is set by the plugin

- **quobyte.user** (string)

user to map volume access to Defaults to serviceaccount user

- **rbd** (RBDPersistentVolumeSource)

rbd represents a Rados Block Device mount on the host that shares a pod's lifetime. More info: <https://examples.k8s.io/volumes/rbd/README.md>

Represents a Rados Block Device mount that lasts the lifetime of a pod. RBD volumes support ownership management and SELinux relabeling.

- **rbd.image** (string), required

image is the rados image name. More info: <https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it>

- **rbd.monitors** ([]string), required

monitors is a collection of Ceph monitors. More info: <https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it>

- **rbd.fsType** (string)

fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs".

Implicitly inferred to be "ext4" if unspecified. More info: <https://kubernetes.io/docs/concepts/storage/volumes#rbd>

- **rbd.keyring** (string)

keyring is the path to key ring for RBDUser. Default is /etc/ceph/keyring. More info: <https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it>

- **rbd.pool** (string)

pool is the rados pool name. Default is rbd. More info: <https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it>

- **rbd.readOnly** (boolean)

readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false. More info: <https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it>

- **rbd.secretRef** (SecretReference)

secretRef is name of the authentication secret for RBDUser. If provided overrides keyring. Default is nil. More info: <https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it>

SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace

- **rbd.secretRef.name** (string)

name is unique within a namespace to reference a secret resource.

- **rbd.secretRef.namespace** (string)

namespace defines the space within which the secret name must be unique.

- **rbd.user** (string)

user is the rados user name. Default is admin. More info:
<https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it>

- **scaleIO** (ScaleIOPersistentVolumeSource)

scaleIO represents a ScaleIO persistent volume attached and mounted on Kubernetes nodes.

ScaleIOPersistentVolumeSource represents a persistent ScaleIO volume

- **scaleIO.gateway** (string), required

gateway is the host address of the ScaleIO API Gateway.

- **scaleIO.secretRef** (SecretReference), required

secretRef references to the secret for ScaleIO user and other sensitive information. If this is not provided, Login operation will fail.

SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace

- **scaleIO.secretRef.name** (string)

name is unique within a namespace to reference a secret resource.

- **scaleIO.secretRef.namespace** (string)

namespace defines the space within which the secret name must be unique.

- **scaleIO.system** (string), required

system is the name of the storage system as configured in ScaleIO.

- **scaleIO.fsType** (string)

fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Default is "xfs"

- **scaleIO.protectionDomain** (string)

protectionDomain is the name of the ScaleIO Protection Domain for the configured storage.

- **scaleIO.readOnly** (boolean)

readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

- **scaleIO.sslEnabled** (boolean)

sslEnabled is the flag to enable/disable SSL communication with Gateway, default false

- **scaleIO.storageMode** (string)

storageMode indicates whether the storage for a volume should be ThickProvisioned or ThinProvisioned. Default is ThinProvisioned.

- **scaleIO.storagePool** (string)

storagePool is the ScaleIO Storage Pool associated with the

protection domain.

- **scaleIO.volumeName** (string)

volumeName is the name of a volume already created in the ScaleIO system that is associated with this volume source.

- **storageos** (StorageOSPersistentVolumeSource)

storageOS represents a StorageOS volume that is attached to the kubelet's host machine and mounted into the pod More info:

<https://examples.k8s.io/volumes/storageos/README.md>

Represents a StorageOS persistent volume resource.

- **storageos.fsType** (string)

fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.

- **storageos.readOnly** (boolean)

readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

- **storageos.secretRef** ([ObjectReference](#))

secretRef specifies the secret to use for obtaining the StorageOS API credentials. If not specified, default values will be attempted.

- **storageos.volumeName** (string)

volumeName is the human-readable name of the StorageOS volume. Volume names are only unique within a namespace.

- **storageos.volumeNamespace** (string)

volumeNamespace specifies the scope of the volume within StorageOS. If no namespace is specified then the Pod's namespace will be used. This allows the Kubernetes name scoping to be mirrored within StorageOS for tighter integration. Set VolumeName to any name to override the default behaviour. Set to "default" if you are not using namespaces within StorageOS. Namespaces that do not pre-exist within StorageOS will be created.

- **vsphereVolume** (VsphereVirtualDiskVolumeSource)

vsphereVolume represents a vSphere volume attached and mounted on kubelets host machine

Represents a vSphere volume resource.

- **vsphereVolume.volumePath** (string), required

volumePath is the path that identifies vSphere volume vmdk

- **vsphereVolume.fsType** (string)

fsType is filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.

- **vsphereVolume.storagePolicyID** (string)

storagePolicyID is the storage Policy Based Management (SPBM) profile ID associated with the StoragePolicyName.

- **vsphereVolume.storagePolicyName** (string)
storagePolicyName is the storage Policy Based Management (SPBM) profile name.

PersistentVolumeStatus

PersistentVolumeStatus is the current status of a persistent volume.

- **lastPhaseTransitionTime** (Time)

lastPhaseTransitionTime is the time the phase transitioned from one to another and automatically resets to current time everytime a volume phase transitions. This is an alpha field and requires enabling PersistentVolumeLastPhaseTransitionTime feature.

Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

- **message** (string)

message is a human-readable message indicating details about why the volume is in this state.

- **phase** (string)

phase indicates if a volume is available, bound to a claim, or released by a claim. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#phase>

- **reason** (string)

reason is a brief CamelCase string that describes any failure and is meant for machine parsing and tidy display in the CLI.

PersistentVolumeList

PersistentVolumeList is a list of PersistentVolume items.

- **apiVersion**: v1
- **kind**: PersistentVolumeList
- **metadata** ([ListMeta](#))

Standard list metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds>

- **items** ([\[\]PersistentVolume](#)), required

items is a list of persistent volumes. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes>

Operations

get read the specified PersistentVolume

HTTP Request

GET /api/v1/persistentvolumes/{name}

Parameters

- **name** (*in path*): string, required

name of the PersistentVolume

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([PersistentVolume](#)): OK

401: Unauthorized

get read status of the specified PersistentVolume

HTTP Request

GET /api/v1/persistentvolumes/{name}/status

Parameters

- **name** (*in path*): string, required

name of the PersistentVolume

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([PersistentVolume](#)): OK

401: Unauthorized

list list or watch objects of kind PersistentVolume

HTTP Request

GET /api/v1/persistentvolumes

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string
[pretty](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)
- **watch** (*in query*): boolean
[watch](#)

Response

200 ([PersistentVolumeList](#)): OK

401: Unauthorized

create create a PersistentVolume

HTTP Request

POST /api/v1/persistentvolumes

Parameters

- **body**: [PersistentVolume](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([PersistentVolume](#)): OK

201 ([PersistentVolume](#)): Created

202 ([PersistentVolume](#)): Accepted

401: Unauthorized

update replace the specified PersistentVolume

HTTP Request

PUT /api/v1/persistentvolumes/{name}

Parameters

- **name** (*in path*): string, required
name of the PersistentVolume
- **body**: [PersistentVolume](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([PersistentVolume](#)): OK

201 ([PersistentVolume](#)): Created

401: Unauthorized

update replace status of the specified PersistentVolume

HTTP Request

PUT /api/v1/persistentvolumes/{name}/status

Parameters

- **name** (*in path*): string, required
name of the PersistentVolume
- **body**: [PersistentVolume](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([PersistentVolume](#)): OK

201 ([PersistentVolume](#)): Created

401: Unauthorized

patch partially update the specified PersistentVolume

HTTP Request

PATCH /api/v1/persistentvolumes/{name}

Parameters

- **name** (*in path*): string, required

name of the PersistentVolume

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([PersistentVolume](#)): OK

201 ([PersistentVolume](#)): Created

401: Unauthorized

patch partially update status of the specified PersistentVolume

HTTP Request

PATCH /api/v1/persistentvolumes/{name}/status

Parameters

- **name** (*in path*): string, required

name of the PersistentVolume

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([PersistentVolume](#)): OK201 ([PersistentVolume](#)): Created

401: Unauthorized

delete delete a PersistentVolume

HTTP Request

DELETE /api/v1/persistentvolumes/{name}

Parameters

- **name** (*in path*): string, required

name of the PersistentVolume

- **body**: [DeleteOptions](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

Response

200 ([PersistentVolume](#)): OK202 ([PersistentVolume](#)): Accepted

401: Unauthorized

deletecollection delete collection of PersistentVolume

HTTP Request

DELETE /api/v1/persistentvolumes

Parameters

- **body**: [DeleteOptions](#)
- **continue** (*in query*): string
[continue](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.3.6 - StorageClass

StorageClass describes the parameters for a class of storage for which PersistentVolumes can be dynamically provisioned.

```
apiVersion: storage.k8s.io/v1
import "k8s.io/api/storage/v1"
```

StorageClass

StorageClass describes the parameters for a class of storage for which PersistentVolumes can be dynamically provisioned.

StorageClasses are non-namespaced; the name of the storage class according to etcd is in ObjectMeta.Name.

- **apiVersion**: storage.k8s.io/v1
- **kind**: StorageClass
- **metadata** ([ObjectMeta](#))

Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **provisioner** (string), required

provisioner indicates the type of the provisioner.

- **allowVolumeExpansion** (boolean)

allowVolumeExpansion shows whether the storage class allow volume expand.

- **allowedTopologies** ([]TopologySelectorTerm)

Atomic: will be replaced during a merge

allowedTopologies restrict the node topologies where volumes can be dynamically provisioned. Each volume plugin defines its own supported topology specifications. An empty TopologySelectorTerm list means there is no topology restriction. This field is only honored by servers that enable the VolumeScheduling feature.

A topology selector term represents the result of label queries. A null or empty topology selector term matches no objects. The requirements of them are ANDed. It provides a subset of functionality as NodeSelectorTerm. This is an alpha feature and may change in the future.

- **allowedTopologies.matchLabelExpressions** ([]TopologySelectorLabelRequirement)

A list of topology selector requirements by labels.

A topology selector requirement is a selector that matches given label. This is an alpha feature and may change in the future.

- **allowedTopologies.matchLabelExpressions.key** (string), required

The label key that the selector applies to.

▪ `allowedTopologies.matchLabelExpressions.values`

([]string), required

An array of string values. One value must match the label to be selected. Each entry in Values is ORed.

• `mountOptions` ([]string)

mountOptions controls the mountOptions for dynamically provisioned PersistentVolumes of this storage class. e.g. ["ro", "soft"]. Not validated - mount of the PVs will simply fail if one is invalid.

• `parameters` (map[string]string)

parameters holds the parameters for the provisioner that should create volumes of this storage class.

• `reclaimPolicy` (string)

reclaimPolicy controls the reclaimPolicy for dynamically provisioned PersistentVolumes of this storage class. Defaults to Delete.

• `volumeBindingMode` (string)

volumeBindingMode indicates how PersistentVolumeClaims should be provisioned and bound. When unset, VolumeBindingImmediate is used. This field is only honored by servers that enable the VolumeScheduling feature.

StorageClassList

StorageClassList is a collection of storage classes.

• `apiVersion`: storage.k8s.io/v1**• `kind`: StorageClassList****• `metadata` ([ListMeta](#))**

Standard list metadata More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

• `items` ([][StorageClass](#)), required

items is the list of StorageClasses

Operations

get read the specified StorageClass

HTTP Request

GET /apis/storage.k8s.io/v1/storageclasses/{name}

Parameters

• `name` (*in path*): string, required

name of the StorageClass

• `pretty` (*in query*): string

[pretty](#)

Response

200 ([StorageClass](#)): OK

401: Unauthorized

list list or watch objects of kind StorageClass

HTTP Request

GET /apis/storage.k8s.io/v1/storageclasses

Parameters

- **allowWatchBookmarks** (*in query*): boolean
[allowWatchBookmarks](#)
- **continue** (*in query*): string
[continue](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)
- **watch** (*in query*): boolean
[watch](#)

Response

200 ([StorageClassList](#)): OK

401: Unauthorized

create create a StorageClass

HTTP Request

POST /apis/storage.k8s.io/v1/storageclasses

Parameters

- **body**: [StorageClass](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([StorageClass](#)): OK

201 ([StorageClass](#)): Created

202 ([StorageClass](#)): Accepted

401: Unauthorized

update replace the specified StorageClass

HTTP Request

PUT /apis/storage.k8s.io/v1/storageclasses/{name}

Parameters

- **name** (*in path*): string, required
name of the StorageClass
- **body**: [StorageClass](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([StorageClass](#)): OK

201 ([StorageClass](#)): Created

401: Unauthorized

patch partially update the specified StorageClass

HTTP Request

PATCH /apis/storage.k8s.io/v1/storageclasses/{name}

Parameters

- **name** (*in path*): string, required

name of the StorageClass

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([StorageClass](#)): OK

201 ([StorageClass](#)): Created

401: Unauthorized

delete delete a StorageClass

HTTP Request

DELETE /apis/storage.k8s.io/v1/storageclasses/{name}

Parameters

- **name** (*in path*): string, required

name of the StorageClass

- **body**: [DeleteOptions](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)

Response

200 ([StorageClass](#)): OK

202 ([StorageClass](#)): Accepted

401: Unauthorized

deletecollection delete collection of StorageClass

HTTP Request

DELETE /apis/storage.k8s.io/v1/storageclasses

Parameters

- **body**: [DeleteOptions](#)
- **continue** (*in query*): string
[continue](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.3.7 - VolumeAttachment

VolumeAttachment captures the intent to attach or detach the specified volume to/from the specified node.

```
apiVersion: storage.k8s.io/v1
import "k8s.io/api/storage/v1"
```

VolumeAttachment

VolumeAttachment captures the intent to attach or detach the specified volume to/from the specified node.

VolumeAttachment objects are non-namespaced.

- **apiVersion**: storage.k8s.io/v1
- **kind**: VolumeAttachment
- **metadata** ([ObjectMeta](#))

Standard object metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **spec** ([VolumeAttachmentSpec](#)), required

spec represents specification of the desired attach/detach volume behavior. Populated by the Kubernetes system.

- **status** ([VolumeAttachmentStatus](#))

status represents status of the VolumeAttachment request.

Populated by the entity completing the attach or detach operation, i.e. the external-attacher.

VolumeAttachmentSpec

VolumeAttachmentSpec is the specification of a VolumeAttachment request.

- **attacher** (string), required

attacher indicates the name of the volume driver that MUST handle this request. This is the name returned by GetPluginName().

- **nodeName** (string), required

nodeName represents the node that the volume should be attached to.

- **source** ([VolumeAttachmentSource](#)), required

source represents the volume that should be attached.

VolumeAttachmentSource represents a volume that should be attached. Right now only PersistentVolumes can be attached via external attacher, in future we may allow also inline volumes in pods. Exactly one member can be set.

- **source.inlineVolumeSpec** ([PersistentVolumeSpec](#))

inlineVolumeSpec contains all the information necessary to

attach a persistent volume defined by a pod's inline VolumeSource. This field is populated only for the CSIMigration feature. It contains translated fields from a pod's inline VolumeSource to a PersistentVolumeSpec. This field is beta-level and is only honored by servers that enabled the CSIMigration feature.

- **source.persistentVolumeName** (string)

persistentVolumeName represents the name of the persistent volume to attach.

VolumeAttachmentStatus

VolumeAttachmentStatus is the status of a VolumeAttachment request.

- **attached** (boolean), required

attached indicates the volume is successfully attached. This field must only be set by the entity completing the attach operation, i.e. the external-attacher.

- **attachError** (VolumeError)

attachError represents the last error encountered during attach operation, if any. This field must only be set by the entity completing the attach operation, i.e. the external-attacher.

VolumeError captures an error encountered during a volume operation.

- **attachError.message** (string)

message represents the error encountered during Attach or Detach operation. This string may be logged, so it should not contain sensitive information.

- **attachError.time** (Time)

time represents the time the error was encountered.

Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

- **attachmentMetadata** (map[string]string)

attachmentMetadata is populated with any information returned by the attach operation, upon successful attach, that must be passed into subsequent WaitForAttach or Mount calls. This field must only be set by the entity completing the attach operation, i.e. the external-attacher.

- **detachError** (VolumeError)

detachError represents the last error encountered during detach operation, if any. This field must only be set by the entity completing the detach operation, i.e. the external-attacher.

VolumeError captures an error encountered during a volume operation.

- **detachError.message** (string)

message represents the error encountered during Attach or Detach operation. This string may be logged, so it should not contain sensitive information.

- **detachError.time** (Time)

time represents the time the error was encountered.

Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

VolumeAttachmentList

VolumeAttachmentList is a collection of VolumeAttachment objects.

- **apiVersion**: storage.k8s.io/v1

- **kind**: VolumeAttachmentList

- **metadata** ([ListMeta](#))

Standard list metadata More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **items** ([]VolumeAttachment), required

items is the list of VolumeAttachments

Operations

get read the specified VolumeAttachment

HTTP Request

GET /apis/storage.k8s.io/v1/volumeattachments/{name}

Parameters

- **name** (*in path*): string, required

name of the VolumeAttachment

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([VolumeAttachment](#)): OK

401: Unauthorized

get read status of the specified VolumeAttachment

HTTP Request

GET /apis/storage.k8s.io/v1/volumeattachments/{name}/status

Parameters

- **name** (*in path*): string, required

name of the VolumeAttachment

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([VolumeAttachment](#)): OK

401: Unauthorized

list list or watch objects of kind VolumeAttachment

HTTP Request

GET /apis/storage.k8s.io/v1/volumeattachments

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([VolumeAttachmentList](#)): OK

401: Unauthorized

create create a VolumeAttachment

HTTP Request

POST /apis/storage.k8s.io/v1/volumeattachments

Parameters

- **body**: [VolumeAttachment](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([VolumeAttachment](#)): OK

201 ([VolumeAttachment](#)): Created

202 ([VolumeAttachment](#)): Accepted

401: Unauthorized

update replace the specified VolumeAttachment

HTTP Request

PUT /apis/storage.k8s.io/v1/volumeattachments/{name}

Parameters

- **name** (*in path*): string, required

name of the VolumeAttachment

- **body**: [VolumeAttachment](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([VolumeAttachment](#)): OK

201 ([VolumeAttachment](#)): Created

401: Unauthorized

update replace status of the specified VolumeAttachment

HTTP Request

PUT /apis/storage.k8s.io/v1/volumeattachments/{name}/status

Parameters

- **name** (*in path*): string, required

name of the VolumeAttachment

- **body**: [VolumeAttachment](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([VolumeAttachment](#)): OK

201 ([VolumeAttachment](#)): Created

401: Unauthorized

patch partially update the specified VolumeAttachment

HTTP Request

PATCH /apis/storage.k8s.io/v1/volumeattachments/{name}

Parameters

- **name** (*in path*): string, required

name of the VolumeAttachment

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([VolumeAttachment](#)): OK

201 ([VolumeAttachment](#)): Created

401: Unauthorized

patch partially update status of the specified VolumeAttachment

HTTP Request

PATCH /apis/storage.k8s.io/v1/volumeattachments/{name}/status

Parameters

- **name** (*in path*): string, required

name of the VolumeAttachment

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([VolumeAttachment](#)): OK

201 ([VolumeAttachment](#)): Created

401: Unauthorized

delete delete a VolumeAttachment

HTTP Request

DELETE /apis/storage.k8s.io/v1/volumeattachments/{name}

Parameters

- **name** (*in path*): string, required
name of the VolumeAttachment
- **body**: [DeleteOptions](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)

Response

200 ([VolumeAttachment](#)): OK

202 ([VolumeAttachment](#)): Accepted

401: Unauthorized

deletecollection delete collection of VolumeAttachment

HTTP Request

DELETE /apis/storage.k8s.io/v1/volumeattachments

Parameters

- **body**: [DeleteOptions](#)
- **continue** (*in query*): string
[continue](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)

- **propagationPolicy** (*in query*): string
[propagationPolicy](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.3.8 - CSIDriver

CSIDriver captures information about a Container Storage Interface (CSI) volume driver deployed on the cluster.

```
apiVersion: storage.k8s.io/v1
import "k8s.io/api/storage/v1"
```

CSIDriver

CSIDriver captures information about a Container Storage Interface (CSI) volume driver deployed on the cluster. Kubernetes attach detach controller uses this object to determine whether attach is required. Kubelet uses this object to determine whether pod information needs to be passed on mount. CSIDriver objects are non-namespaced.

- **apiVersion**: storage.k8s.io/v1
- **kind**: CSIDriver
- **metadata** ([ObjectMeta](#))

Standard object metadata. metadata.Name indicates the name of the CSI driver that this object refers to; it MUST be the same name returned by the CSI GetPluginName() call for that driver. The driver name must be 63 characters or less, beginning and ending with an alphanumeric character ([a-z0-9A-Z]) with dashes (-), dots (.), and alphanumerics between. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **spec** ([CSIDriverSpec](#)), required

spec represents the specification of the CSI Driver.

CSIDriverSpec

CSIDriverSpec is the specification of a CSIDriver.

- **attachRequired** (boolean)

attachRequired indicates this CSI volume driver requires an attach operation (because it implements the CSI ControllerPublishVolume() method), and that the Kubernetes attach detach controller should call the attach volume interface which checks the volumeattachment status and waits until the volume is attached before proceeding to mounting. The CSI external-attacher coordinates with CSI volume driver and updates the volumeattachment status when the attach operation is complete. If the CSIDriverRegistry feature gate is enabled and the value is specified to false, the attach operation will be skipped. Otherwise the attach operation will be called.

This field is immutable.

- **fsGroupPolicy** (string)

fsGroupPolicy defines if the underlying volume supports changing ownership and permission of the volume before being mounted. Refer to the specific FSGroupPolicy values for additional details.

This field is immutable.

Defaults to `ReadWriteOnceWithFSType`, which will examine each volume to determine if Kubernetes should modify ownership and permissions of the volume. With the default policy the defined `fsGroup` will only be applied if a `fstype` is defined and the volume's access mode contains `ReadWriteOnce`.

- **podInfoOnMount** (boolean)

`podInfoOnMount` indicates this CSI volume driver requires additional pod information (like `podName`, `podUID`, etc.) during mount operations, if set to true. If set to false, pod information will not be passed on mount. Default is false.

The CSI driver specifies `podInfoOnMount` as part of driver deployment. If true, Kubelet will pass pod information as `VolumeContext` in the CSI `NodePublishVolume()` calls. The CSI driver is responsible for parsing and validating the information passed in as `VolumeContext`.

The following `VolumeConext` will be passed if `podInfoOnMount` is set to true. This list might grow, but the prefix will be used.

```
"csi.storage.k8s.io/pod.name": pod.Name "csi.storage.k8s.io/pod.namespace": pod.Namespace "csi.storage.k8s.io/pod.uid": string(pod.UID) "csi.storage.k8s.io/ephemeral": "true" if the volume is an ephemeral inline volume defined by a CSIVolumeSource, otherwise "false"
```

`"csi.storage.k8s.io/ephemeral"` is a new feature in Kubernetes 1.16. It is only required for drivers which support both the "Persistent" and "Ephemeral" `VolumeLifecycleMode`. Other drivers can leave pod info disabled and/or ignore this field. As Kubernetes 1.15 doesn't support this field, drivers can only support one mode when deployed on such a cluster and the deployment determines which mode that is, for example via a command line parameter of the driver.

This field is immutable.

- **requiresRepublish** (boolean)

`requiresRepublish` indicates the CSI driver wants `NodePublishVolume` being periodically called to reflect any possible change in the mounted volume. This field defaults to false.

Note: After a successful initial `NodePublishVolume` call, subsequent calls to `NodePublishVolume` should only update the contents of the volume. New mount points will not be seen by a running container.

- **seLinuxMount** (boolean)

`seLinuxMount` specifies if the CSI driver supports "`-o context`" mount option.

When "true", the CSI driver must ensure that all volumes provided by this CSI driver can be mounted separately with different `-o context` options. This is typical for storage backends that provide volumes as filesystems on block devices or as independent shared volumes. Kubernetes will call `NodeStage` / `NodePublish` with "`-o context=xyz`" mount option when mounting a `ReadWriteOncePod` volume used in Pod that has explicitly set SELinux context. In the future, it may be expanded to other volume AccessModes. In any case, Kubernetes will ensure that the volume is mounted only with a single SELinux context.

When "false", Kubernetes won't pass any special SELinux mount options to the driver. This is typical for volumes that represent subdirectories of a bigger shared filesystem.

Default is "false".

- **storageCapacity** (boolean)

storageCapacity indicates that the CSI volume driver wants pod scheduling to consider the storage capacity that the driver deployment will report by creating CSIStrageCapacity objects with capacity information, if set to true.

The check can be enabled immediately when deploying a driver. In that case, provisioning new volumes with late binding will pause until the driver deployment has published some suitable CSIStrageCapacity object.

Alternatively, the driver can be deployed with the field unset or false and it can be flipped later when storage capacity information has been published.

This field was immutable in Kubernetes <= 1.22 and now is mutable.

- **tokenRequests** ([]TokenRequest)

Atomic: will be replaced during a merge

tokenRequests indicates the CSI driver needs pods' service account tokens it is mounting volume for to do necessary authentication.

Kubelet will pass the tokens in VolumeContext in the CSI NodePublishVolume calls. The CSI driver should parse and validate the following VolumeContext: "csi.storage.k8s.io/serviceAccount.tokens": { "<audience>": { "token": <token>, "expirationTimestamp": <expiration timestamp in RFC3339>, }, ... }

Note: Audience in each TokenRequest should be different and at most one token is empty string. To receive a new token after expiry, RequiresRepublish can be used to trigger NodePublishVolume periodically.

TokenRequest contains parameters of a service account token.

- **tokenRequests.audience** (string), required

audience is the intended audience of the token in "TokenRequestSpec". It will default to the audiences of kube apiserver.

- **tokenRequests.expirationSeconds** (int64)

expirationSeconds is the duration of validity of the token in "TokenRequestSpec". It has the same default value of "ExpirationSeconds" in "TokenRequestSpec".

- **volumeLifecycleModes** ([]string)

Set: unique values will be kept during a merge

volumeLifecycleModes defines what kind of volumes this CSI volume driver supports. The default if the list is empty is "Persistent", which is the usage defined by the CSI specification and implemented in Kubernetes via the usual PV/PVC mechanism.

The other mode is "Ephemeral". In this mode, volumes are defined inline inside the pod spec with CSIVolumeSource and their lifecycle is tied to the lifecycle of that pod. A driver has to be aware of this because it is only going to get a NodePublishVolume call for such a

volume.

For more information about implementing this mode, see <https://kubernetes-csi.github.io/docs/ephemeral-local-volumes.html> A driver can support one or more of these modes and more modes may be added in the future.

This field is beta. This field is immutable.

CSIDriverList

CSIDriverList is a collection of CSIDriver objects.

- **apiVersion**: storage.k8s.io/v1
- **kind**: CSIDriverList
- **metadata** ([ListMeta](#))

Standard list metadata More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **items** ([][CSIDriver](#)), required

items is the list of CSIDriver

Operations

get read the specified CSIDriver

HTTP Request

GET /apis/storage.k8s.io/v1/csidrivers/{name}

Parameters

- **name** (*in path*): string, required

name of the CSIDriver

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([CSIDriver](#)): OK

401: Unauthorized

list list or watch objects of kind CSIDriver

HTTP Request

GET /apis/storage.k8s.io/v1/csidrivers

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([CSIDriverList](#)): OK

401: Unauthorized

create create a CSIDriver

HTTP Request

POST /apis/storage.k8s.io/v1/csidrivers

Parameters

- **body**: [CSIDriver](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([CSIDriver](#)): OK

201 ([CSIDriver](#)): Created

202 ([CSIDriver](#)): Accepted

401: Unauthorized

update replace the specified CSIDriver

HTTP Request

PUT /apis/storage.k8s.io/v1/csidrivers/{name}

Parameters

- **name** (*in path*): string, required

name of the CSIDriver

- **body**: [CSIDriver](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([CSIDriver](#)): OK

201 ([CSIDriver](#)): Created

401: Unauthorized

patch partially update the specified CSIDriver

HTTP Request

PATCH /apis/storage.k8s.io/v1/csidrivers/{name}

Parameters

- **name** (*in path*): string, required

name of the CSIDriver

- **body**: [Patch](#), required

- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **force** (*in query*): boolean
[force](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([CSIDriver](#)): OK

201 ([CSIDriver](#)): Created

401: Unauthorized

delete delete a CSIDriver

HTTP Request

DELETE /apis/storage.k8s.io/v1/csidrivers/{name}

Parameters

- **name** (*in path*): string, required
name of the CSIDriver
- **body**: [DeleteOptions](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)

Response

200 ([CSIDriver](#)): OK

202 ([CSIDriver](#)): Accepted

401: Unauthorized

deletecollection delete collection of CSIDriver

HTTP Request

DELETE /apis/storage.k8s.io/v1/csidrivers

Parameters

- **body**: [DeleteOptions](#)
- **continue** (*in query*): string
[continue](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.3.9 - CSINode

CSINode holds information about all CSI drivers installed on a node.

```
apiVersion: storage.k8s.io/v1
import "k8s.io/api/storage/v1"
```

CSINode

CSINode holds information about all CSI drivers installed on a node. CSI drivers do not need to create the CSINode object directly. As long as they use the node-driver-registrar sidecar container, the kubelet will automatically populate the CSINode object for the CSI driver as part of kubelet plugin registration. CSINode has the same name as a node. If the object is missing, it means either there are no CSI Drivers available on the node, or the Kubelet version is low enough that it doesn't create this object. CSINode has an OwnerReference that points to the corresponding node object.

- **apiVersion**: storage.k8s.io/v1
- **kind**: CSINode
- **metadata** ([ObjectMeta](#))

Standard object's metadata. metadata.name must be the Kubernetes node name.

- **spec** ([CSINodeSpec](#)), required

spec is the specification of CSINode

CSINodeSpec

CSINodeSpec holds information about the specification of all CSI drivers installed on a node

- **drivers** ([]CSINodeDriver), required

Patch strategy: merge on key name

drivers is a list of information of all CSI Drivers existing on a node. If all drivers in the list are uninstalled, this can become empty.

CSINodeDriver holds information about the specification of one CSI driver installed on a node

- **drivers.name** (string), required

name represents the name of the CSI driver that this object refers to. This MUST be the same name returned by the CSI GetPluginName() call for that driver.

- **drivers.nodeID** (string), required

nodeID of the node from the driver point of view. This field enables Kubernetes to communicate with storage systems that do not share the same nomenclature for nodes. For example, Kubernetes may refer to a given node as "node1", but the storage system may refer to the same node as

"nodeA". When Kubernetes issues a command to the storage system to attach a volume to a specific node, it can use this field to refer to the node name using the ID that the storage system will understand, e.g. "nodeA" instead of "node1". This field is required.

- o **drivers.allocatable** (VolumeNodeResources)

allocatable represents the volume resources of a node that are available for scheduling. This field is beta.

VolumeNodeResources is a set of resource limits for scheduling of volumes.

- **drivers.allocatable.count** (int32)

count indicates the maximum number of unique volumes managed by the CSI driver that can be used on a node. A volume that is both attached and mounted on a node is considered to be used once, not twice. The same rule applies for a unique volume that is shared among multiple pods on the same node. If this field is not specified, then the supported number of volumes on this node is unbounded.

- o **drivers.topologyKeys** ([]string)

topologyKeys is the list of keys supported by the driver. When a driver is initialized on a cluster, it provides a set of topology keys that it understands (e.g. "company.com/zone", "company.com/region"). When a driver is initialized on a node, it provides the same topology keys along with values. Kubelet will expose these topology keys as labels on its own node object. When Kubernetes does topology aware provisioning, it can use this list to determine which labels it should retrieve from the node object and pass back to the driver. It is possible for different nodes to use different topology keys. This can be empty if driver does not support topology.

CSINodeList

CSINodeList is a collection of CSINode objects.

- **apiVersion**: storage.k8s.io/v1

- **kind**: CSINodeList

- **metadata** ([ListMeta](#))

Standard list metadata More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **items** ([][CSINode](#)), required

items is the list of CSINode

Operations

get read the specified CSINode

HTTP Request

GET /apis/storage.k8s.io/v1/csinodes/{name}

Parameters

- **name** (*in path*): string, required

name of the CSINode

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([CSINode](#)): OK

401: Unauthorized

list list or watch objects of kind CSINode

HTTP Request

GET /apis/storage.k8s.io/v1/csinodes

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([CSINodeList](#)): OK

401: Unauthorized

create create a CSINode

HTTP Request

POST /apis/storage.k8s.io/v1/csinodes

Parameters

- **body**: [CSINode](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([CSINode](#)): OK

201 ([CSINode](#)): Created

202 ([CSINode](#)): Accepted

401: Unauthorized

update replace the specified CSINode

HTTP Request

PUT /apis/storage.k8s.io/v1/csinodes/{name}

Parameters

- **name** (*in path*): string, required
name of the CSINode
- **body**: [CSINode](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([CSINode](#)): OK

201 ([CSINode](#)): Created

401: Unauthorized

patch partially update the specified CSINode

HTTP Request

PATCH /apis/storage.k8s.io/v1/csinodes/{name}

Parameters

- **name** (*in path*): string, required

name of the CSINode

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([CSINode](#)): OK

201 ([CSINode](#)): Created

401: Unauthorized

delete delete a CSINode

HTTP Request

DELETE /apis/storage.k8s.io/v1/csinodes/{name}

Parameters

- **name** (*in path*): string, required

name of the CSINode

- **body**: [DeleteOptions](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)

Response

200 ([CSINode](#)): OK

202 ([CSINode](#)): Accepted

401: Unauthorized

deletecollection delete collection of CSINode

HTTP Request

DELETE /apis/storage.k8s.io/v1/csinodes

Parameters

- **body**: [DeleteOptions](#)
- **continue** (*in query*): string
[continue](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)

- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.3.10 - CSIStrageCapacity

CSIStrageCapacity stores the result of one CSI GetCapacity call.

```
apiVersion: storage.k8s.io/v1
import "k8s.io/api/storage/v1"
```

CSIStrageCapacity

CSIStrageCapacity stores the result of one CSI GetCapacity call. For a given StorageClass, this describes the available capacity in a particular topology segment. This can be used when considering where to instantiate new PersistentVolumes.

For example this can express things like: - StorageClass "standard" has "1234 GiB" available in "topology.kubernetes.io/zone=us-east1" - StorageClass "localssd" has "10 GiB" available in "kubernetes.io/hostname=knode-abc123"

The following three cases all imply that no capacity is available for a certain combination: - no object exists with suitable topology and storage class name - such an object exists, but the capacity is unset - such an object exists, but the capacity is zero

The producer of these objects can decide which approach is more suitable.

They are consumed by the kube-scheduler when a CSI driver opts into capacity-aware scheduling with CSIDriverSpec.StorageCapacity. The scheduler compares the MaximumVolumeSize against the requested size of pending volumes to filter out unsuitable nodes. If MaximumVolumeSize is unset, it falls back to a comparison against the less precise Capacity. If that is also unset, the scheduler assumes that capacity is insufficient and tries some other node.

- **apiVersion:** storage.k8s.io/v1
- **kind:** CSIStrageCapacity
- **metadata** ([ObjectMeta](#))

Standard object's metadata. The name has no particular meaning. It must be a DNS subdomain (dots allowed, 253 characters). To ensure that there are no conflicts with other CSI drivers on the cluster, the recommendation is to use csisc-`<uuid>`, a generated name, or a reverse-domain name which ends with the unique CSI driver name.

Objects are namespaced.

More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **storageClassName** (string), required

storageClassName represents the name of the StorageClass that the reported capacity applies to. It must meet the same requirements as the name of a StorageClass object (non-empty, DNS subdomain). If that object no longer exists, the CSIStrageCapacity object is obsolete and should be removed by its creator. This field is immutable.

- **capacity** ([Quantity](#))

capacity is the value reported by the CSI driver in its GetCapacityResponse for a GetCapacityRequest with topology and parameters that match the previous fields.

The semantic is currently (CSI spec 1.2) defined as: The available capacity, in bytes, of the storage that can be used to provision volumes. If not set, that information is currently unavailable.

- **maximumVolumeSize** ([Quantity](#))

maximumVolumeSize is the value reported by the CSI driver in its GetCapacityResponse for a GetCapacityRequest with topology and parameters that match the previous fields.

This is defined since CSI spec 1.4.0 as the largest size that may be used in a CreateVolumeRequest.capacity_range.required_bytes field to create a volume with the same parameters as those in GetCapacityRequest. The corresponding value in the Kubernetes API is ResourceRequirements.Requests in a volume claim.

- **nodeTopology** ([LabelSelector](#))

nodeTopology defines which nodes have access to the storage for which capacity was reported. If not set, the storage is not accessible from any node in the cluster. If empty, the storage is accessible from all nodes. This field is immutable.

CSIStorageCapacityList

CSIStorageCapacityList is a collection of CSIStorageCapacity objects.

- **apiVersion**: storage.k8s.io/v1

- **kind**: CSIStorageCapacityList

- **metadata** ([ListMeta](#))

Standard list metadata More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **items** ([][CSIStorageCapacity](#)), required

Map: unique values on key name will be kept during a merge

items is the list of CSIStorageCapacity objects.

Operations

get read the specified CSIStorageCapacity

HTTP Request

GET /apis/storage.k8s.io/v1/namespaces/{namespace}/csistoragecapacities/{name}

Parameters

- **name** (*in path*): string, required

name of the CSIStorageCapacity

- **namespace** (*in path*): string, required

[namespace](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([CSIStorageCapacity](#)): OK

401: Unauthorized

list list or watch objects of kind CSIStorageCapacity

HTTP Request

GET /apis/storage.k8s.io/v1/namespaces/{namespace}/csistoragecapacities

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([CSIStrageCapacityList](#)): OK

401: Unauthorized

list list or watch objects of kind CSIStrageCapacity

HTTP Request

GET /apis/storage.k8s.io/v1/csistoragecapacities

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([CSIStrageCapacityList](#)): OK

401: Unauthorized

create create a CSIStrageCapacity

HTTP Request

POST /apis/storage.k8s.io/v1/namespaces/{namespace}/csistoragecapacities

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [CSIStrageCapacity](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([CSIStrageCapacity](#)): OK

201 ([CSIStrageCapacity](#)): Created

202 ([CSIStrageCapacity](#)): Accepted

401: Unauthorized

update replace the specified CSIStrageCapacity

HTTP Request

PUT /apis/storage.k8s.io/v1/namespaces/{namespace}/csistoragecapacities/{name}

Parameters

- **name** (*in path*): string, required

name of the CSIStrageCapacity

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [CSIStrageCapacity](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([CSIStrageCapacity](#)): OK201 ([CSIStrageCapacity](#)): Created

401: Unauthorized

patch partially update the specified CSIStrageCapacity

HTTP Request

PATCH /apis/storage.k8s.io/v1/namespaces/{namespace}/csistoragecapacities/{name}

Parameters

- **name** (*in path*): string, required
name of the CSIStrageCapacity
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [Patch](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **force** (*in query*): boolean
[force](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([CSIStrageCapacity](#)): OK201 ([CSIStrageCapacity](#)): Created

401: Unauthorized

delete delete a CSIStrageCapacity

HTTP Request

DELETE /apis/storage.k8s.io/v1/namespaces/{namespace}/

csistoragecapacities/{name}

Parameters

- **name** (*in path*): string, required
name of the CSIStrageCapacity
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [DeleteOptions](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)

Response

200 ([Status](#)): OK

202 ([Status](#)): Accepted

401: Unauthorized

deletecollection delete collection of CSIStrageCapacity

HTTP Request

DELETE /apis/storage.k8s.io/v1/namespaces/{namespace}/csistoragecapacities

Parameters

- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [DeleteOptions](#)
- **continue** (*in query*): string
[continue](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.4 - Authentication Resources

5.4.1 - ServiceAccount

ServiceAccount binds together:

- * a name, understood by users, and perhaps by peripheral systems, for an identity
- * a principal that can be authenticated and authorized
- * a set of secrets.

```
apiVersion: v1

import "k8s.io/api/core/v1"
```

ServiceAccount

ServiceAccount binds together:

- * a name, understood by users, and perhaps by peripheral systems, for an identity
- * a principal that can be authenticated and authorized
- * a set of secrets

- **apiVersion**: v1
- **kind**: ServiceAccount
- **metadata** ([ObjectMeta](#))

Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **automountServiceAccountToken** (boolean)

AutomountServiceAccountToken indicates whether pods running as this service account should have an API token automatically mounted. Can be overridden at the pod level.

- **imagePullSecrets** ([\[\]LocalObjectReference](#))

ImagePullSecrets is a list of references to secrets in the same namespace to use for pulling any images in pods that reference this ServiceAccount. ImagePullSecrets are distinct from Secrets because Secrets can be mounted in the pod, but ImagePullSecrets are only accessed by the kubelet. More info: <https://kubernetes.io/docs/concepts/containers/images/#specifying-imagepullsecrets-on-a-pod>

- **secrets** ([\[\]ObjectReference](#))

Patch strategy: merge on key name

Secrets is a list of the secrets in the same namespace that pods running using this ServiceAccount are allowed to use. Pods are only limited to this list if this service account has a "kubernetes.io/enforce-mountable-secrets" annotation set to "true". This field should not be used to find auto-generated service account token secrets for use outside of pods. Instead, tokens can be requested directly using the TokenRequest API, or service account token secrets can be manually created. More info: <https://kubernetes.io/docs/concepts/configuration/secret>

ServiceAccountList

ServiceAccountList is a list of ServiceAccount objects

- **apiVersion**: v1
- **kind**: ServiceAccountList
- **metadata** ([ListMeta](#))

Standard list metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds>

- **items** ([] [ServiceAccount](#)), required

List of ServiceAccounts. More info: <https://kubernetes.io/docs/tasks/configure-pod-container/configure-service-account/>

Operations

get read the specified ServiceAccount

HTTP Request

GET /api/v1/namespaces/{namespace}/serviceaccounts/{name}

Parameters

- **name** (*in path*): string, required
name of the ServiceAccount
- **namespace** (*in path*): string, required
[namespace](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([ServiceAccount](#)): OK

401: Unauthorized

list list or watch objects of kind ServiceAccount

HTTP Request

GET /api/v1/namespaces/{namespace}/serviceaccounts

Parameters

- **namespace** (*in path*): string, required
[namespace](#)
- **allowWatchBookmarks** (*in query*): boolean
[allowWatchBookmarks](#)

- **continue** (*in query*): string
[continue](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)
- **watch** (*in query*): boolean
[watch](#)

Response

200 ([ServiceAccountList](#)): OK

401: Unauthorized

list list or watch objects of kind ServiceAccount

HTTP Request

GET /api/v1/serviceaccounts

Parameters

- **allowWatchBookmarks** (*in query*): boolean
[allowWatchBookmarks](#)
- **continue** (*in query*): string
[continue](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **labelSelector** (*in query*): string
[labelSelector](#)

- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)
- **watch** (*in query*): boolean
[watch](#)

Response

200 ([ServiceAccountList](#)): OK

401: Unauthorized

create create a ServiceAccount

HTTP Request

POST /api/v1/namespaces/{namespace}/serviceaccounts

Parameters

- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [ServiceAccount](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([ServiceAccount](#)): OK

201 ([ServiceAccount](#)): Created

202 ([ServiceAccount](#)): Accepted

401: Unauthorized

update replace the specified ServiceAccount

HTTP Request

PUT /api/v1/namespaces/{namespace}/serviceaccounts/{name}

Parameters

- **name** (*in path*): string, required
name of the ServiceAccount
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [ServiceAccount](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([ServiceAccount](#)): OK

201 ([ServiceAccount](#)): Created

401: Unauthorized

patch partially update the specified ServiceAccount

HTTP Request

PATCH /api/v1/namespaces/{namespace}/serviceaccounts/{name}

Parameters

- **name** (*in path*): string, required
name of the ServiceAccount
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [Patch](#), required
- **dryRun** (*in query*): string
[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ServiceAccount](#)): OK

201 ([ServiceAccount](#)): Created

401: Unauthorized

delete delete a ServiceAccount

HTTP Request

DELETE /api/v1/namespaces/{namespace}/serviceaccounts/{name}

Parameters

- **name** (*in path*): string, required

name of the ServiceAccount

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [DeleteOptions](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

Response

200 ([ServiceAccount](#)): OK

202 ([ServiceAccount](#)): Accepted

401: Unauthorized

deletecollection delete collection of ServiceAccount

HTTP Request

DELETE /api/v1/namespaces/{namespace}/serviceaccounts

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [DeleteOptions](#)

- **continue** (*in query*): string

[continue](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.4.2 - TokenRequest

TokenRequest requests a token for a given service account.

```
apiVersion: authentication.k8s.io/v1

import "k8s.io/api/authentication/v1"
```

TokenRequest

TokenRequest requests a token for a given service account.

- **apiVersion**: authentication.k8s.io/v1

- **kind**: TokenRequest

- **metadata** ([ObjectMeta](#))

Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **spec** ([TokenRequestSpec](#)), required

Spec holds information about the request being evaluated

- **status** ([TokenRequestStatus](#))

Status is filled in by the server and indicates whether the token can be authenticated.

TokenRequestSpec

TokenRequestSpec contains client provided parameters of a token request.

- **audiences** ([]string), required

Audiences are the intended audiences of the token. A recipient of a token must identify themselves with an identifier in the list of audiences of the token, and otherwise should reject the token. A token issued for multiple audiences may be used to authenticate against any of the audiences listed but implies a high degree of trust between the target audiences.

- **boundObjectRef** ([BoundObjectReference](#))

BoundObjectRef is a reference to an object that the token will be bound to. The token will only be valid for as long as the bound object exists. NOTE: The API server's TokenReview endpoint will validate the BoundObjectRef, but other audiences may not. Keep ExpirationSeconds small if you want prompt revocation.

BoundObjectReference is a reference to an object that a token is bound to.

- **boundObjectRef.apiVersion** (string)

API version of the referent.

- **boundObjectRef.kind** (string)

Kind of the referent. Valid kinds are 'Pod' and 'Secret'.

- **boundObjectRef.name** (string)

Name of the referent.

- **boundObjectRef.uid** (string)

UID of the referent.

- **expirationSeconds** (int64)

ExpirationSeconds is the requested duration of validity of the request. The token issuer may return a token with a different validity duration so a client needs to check the 'expiration' field in a response.

TokenRequestStatus

TokenRequestStatus is the result of a token request.

- **expirationTimestamp** (Time), required

ExpirationTimestamp is the time of expiration of the returned token.

Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

- **token** (string), required

Token is the opaque bearer token.

Operations

create create token of a ServiceAccount

HTTP Request

POST /api/v1/namespaces/{namespace}/serviceaccounts/{name}/token

Parameters

- **name** (*in path*): string, required

name of the TokenRequest

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [TokenRequest](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([TokenRequest](#)): OK

201 ([TokenRequest](#)): Created

202 ([TokenRequest](#)): Accepted

401: Unauthorized

5.4.3 - TokenReview

TokenReview attempts to authenticate a token to a known user.

```
apiVersion: authentication.k8s.io/v1

import "k8s.io/api/authentication/v1"
```

TokenReview

TokenReview attempts to authenticate a token to a known user. Note: TokenReview requests may be cached by the webhook token authenticator plugin in the kube-apiserver.

- **apiVersion**: authentication.k8s.io/v1
- **kind**: TokenReview
- **metadata** ([ObjectMeta](#))

Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **spec** ([TokenReviewSpec](#)), required

Spec holds information about the request being evaluated

- **status** ([TokenReviewStatus](#))

Status is filled in by the server and indicates whether the request can be authenticated.

TokenReviewSpec

TokenReviewSpec is a description of the token authentication request.

- **audiences** ([]string)

Audiences is a list of the identifiers that the resource server presented with the token identifies as. Audience-aware token authenticators will verify that the token was intended for at least one of the audiences in this list. If no audiences are provided, the audience will default to the audience of the Kubernetes apiserver.

- **token** (string)

Token is the opaque bearer token.

TokenReviewStatus

TokenReviewStatus is the result of the token authentication request.

- **audiences** ([]string)

Audiences are audience identifiers chosen by the authenticator that are compatible with both the TokenReview and token. An identifier is any identifier in the intersection of the TokenReviewSpec audiences and the token's audiences. A client of the TokenReview

API that sets the spec.audiences field should validate that a compatible audience identifier is returned in the status.audiences field to ensure that the TokenReview server is audience aware. If a TokenReview returns an empty status.audience field where status.authenticated is "true", the token is valid against the audience of the Kubernetes API server.

- **authenticated** (boolean)

Authenticated indicates that the token was associated with a known user.

- **error** (string)

Error indicates that the token couldn't be checked

- **user** (UserInfo)

User is the UserInfo associated with the provided token.

UserInfo holds the information about the user needed to implement the user.Info interface.

- **user.extra** (map[string][]string)

Any additional information provided by the authenticator.

- **user.groups** ([]string)

The names of groups this user is a part of.

- **user.uid** (string)

A unique value that identifies this user across time. If this user is deleted and another user by the same name is added, they will have different UIDs.

- **user.username** (string)

The name that uniquely identifies this user among all active users.

Operations

create create a TokenReview

HTTP Request

POST /apis/authentication.k8s.io/v1/tokenreviews

Parameters

- **body**: [TokenReview](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([TokenReview](#)): OK

201 ([TokenReview](#)): Created

202 ([TokenReview](#)): Accepted

401: Unauthorized

5.4.4 - CertificateSigningRequest

CertificateSigningRequest objects provide a mechanism to obtain x509 certificates by submitting a certificate signing request, and having it asynchronously approved and issued.

```
apiVersion: certificates.k8s.io/v1
import "k8s.io/api/certificates/v1"
```

CertificateSigningRequest

CertificateSigningRequest objects provide a mechanism to obtain x509 certificates by submitting a certificate signing request, and having it asynchronously approved and issued.

Kubelets use this API to obtain:

1. client certificates to authenticate to kube-apiserver (with the "kubernetes.io/kube-apiserver-client-kubelet" signerName).
2. serving certificates for TLS endpoints kube-apiserver can connect to securely (with the "kubernetes.io/kubelet-serving" signerName).

This API can be used to request client certificates to authenticate to kube-apiserver (with the "kubernetes.io/kube-apiserver-client" signerName), or to obtain certificates from custom non-Kubernetes signers.

- **apiVersion**: certificates.k8s.io/v1
- **kind**: CertificateSigningRequest
- **metadata** ([ObjectMeta](#))
- **spec** ([CertificateSigningRequestSpec](#)), required

spec contains the certificate request, and is immutable after creation. Only the request, signerName, expirationSeconds, and usages fields can be set on creation. Other fields are derived by Kubernetes and cannot be modified by users.

- **status** ([CertificateSigningRequestStatus](#))

status contains information about whether the request is approved or denied, and the certificate issued by the signer, or the failure condition indicating signer failure.

CertificateSigningRequestSpec

CertificateSigningRequestSpec contains the certificate request.

- **request** ([]byte), required

Atomic: will be replaced during a merge

request contains an x509 certificate signing request encoded in a "CERTIFICATE REQUEST" PEM block. When serialized as JSON or YAML, the data is additionally base64-encoded.

- **signerName** (string), required

signerName indicates the requested signer, and is a qualified name.

List/watch requests for CertificateSigningRequests can filter on this field using a "spec.signerName=NAME" fieldSelector.

Well-known Kubernetes signers are:

1. "kubernetes.io/kube-apiserver-client": issues client certificates that can be used to authenticate to kube-apiserver. Requests for this signer are never auto-approved by kube-controller-manager, can be issued by the "csrsigning" controller in kube-controller-manager.
2. "kubernetes.io/kube-apiserver-client-kubelet": issues client certificates that kubelets use to authenticate to kube-apiserver. Requests for this signer can be auto-approved by the "csrapproving" controller in kube-controller-manager, and can be issued by the "csrsigning" controller in kube-controller-manager.
3. "kubernetes.io/kubelet-serving" issues serving certificates that kubelets use to serve TLS endpoints, which kube-apiserver can connect to securely. Requests for this signer are never auto-approved by kube-controller-manager, and can be issued by the "csrsigning" controller in kube-controller-manager.

More details are available at <https://k8s.io/docs/reference/access-authn-authz/certificate-signing-requests/#kubernetes-signers>

Custom signerNames can also be specified. The signer defines:

1. Trust distribution: how trust (CA bundles) are distributed.
2. Permitted subjects: and behavior when a disallowed subject is requested.
3. Required, permitted, or forbidden x509 extensions in the request (including whether subjectAltNames are allowed, which types, restrictions on allowed values) and behavior when a disallowed extension is requested.
4. Required, permitted, or forbidden key usages / extended key usages.
5. Expiration/certificate lifetime: whether it is fixed by the signer, configurable by the admin.
6. Whether or not requests for CA certificates are allowed.

- **expirationSeconds** (int32)

expirationSeconds is the requested duration of validity of the issued certificate. The certificate signer may issue a certificate with a different validity duration so a client must check the delta between the notBefore and notAfter fields in the issued certificate to determine the actual duration.

The v1.22+ in-tree implementations of the well-known Kubernetes signers will honor this field as long as the requested duration is not greater than the maximum duration they will honor per the --cluster-signing-duration CLI flag to the Kubernetes controller manager.

Certificate signers may not honor this field for various reasons:

1. Old signer that is unaware of the field (such as the in-tree implementations prior to v1.22)
2. Signer whose configured maximum is shorter than the requested duration
3. Signer whose configured minimum is longer than the requested duration

The minimum valid value for expirationSeconds is 600, i.e. 10

minutes.

- **extra** (map[string][]string)

extra contains extra attributes of the user that created the CertificateSigningRequest. Populated by the API server on creation and immutable.

- **groups** ([]string)

Atomic: will be replaced during a merge

groups contains group membership of the user that created the CertificateSigningRequest. Populated by the API server on creation and immutable.

- **uid** (string)

uid contains the uid of the user that created the CertificateSigningRequest. Populated by the API server on creation and immutable.

- **usages** ([]string)

Atomic: will be replaced during a merge

usages specifies a set of key usages requested in the issued certificate.

Requests for TLS client certificates typically request: "digital signature", "key encipherment", "client auth".

Requests for TLS serving certificates typically request: "key encipherment", "digital signature", "server auth".

Valid values are: "signing", "digital signature", "content commitment", "key encipherment", "key agreement", "data encipherment", "cert sign", "crl sign", "encipher only", "decipher only", "any", "server auth", "client auth", "code signing", "email protection", "s/mime", "ipsec end system", "ipsec tunnel", "ipsec user", "timestamping", "ocsp signing", "microsoft sgc", "netscape sgc"

- **username** (string)

username contains the name of the user that created the CertificateSigningRequest. Populated by the API server on creation and immutable.

CertificateSigningRequestStatus

CertificateSigningRequestStatus contains conditions used to indicate approved/denied/failed status of the request, and the issued certificate.

- **certificate** ([]byte)

Atomic: will be replaced during a merge

certificate is populated with an issued certificate by the signer after an Approved condition is present. This field is set via the /status subresource. Once populated, this field is immutable.

If the certificate signing request is denied, a condition of type "Denied" is added and this field remains empty. If the signer cannot issue the certificate, a condition of type "Failed" is added and this field remains empty.

Validation requirements:

1. certificate must contain one or more PEM blocks.
2. All PEM blocks must have the "CERTIFICATE" label, contain no headers, and the encoded data must be a BER-encoded ASN.1 Certificate structure as described in section 4 of RFC5280.
3. Non-PEM content may appear before or after the "CERTIFICATE" PEM blocks and is unvalidated, to allow for explanatory text as described in section 5.2 of RFC7468.

If more than one PEM block is present, and the definition of the requested spec.signerName does not indicate otherwise, the first block is the issued certificate, and subsequent blocks should be treated as intermediate certificates and presented in TLS handshakes.

The certificate is encoded in PEM format.

When serialized as JSON or YAML, the data is additionally base64-encoded, so it consists of:

```
base64(  
  -----BEGIN CERTIFICATE-----  
  ...  
  -----END CERTIFICATE-----  
)
```

- **conditions** ([]CertificateSigningRequestCondition)

Map: unique values on key type will be kept during a merge

conditions applied to the request. Known conditions are "Approved", "Denied", and "Failed".

CertificateSigningRequestCondition describes a condition of a CertificateSigningRequest object

- **conditions.status** (string), required

status of the condition, one of True, False, Unknown.

Approved, Denied, and Failed conditions may not be "False" or "Unknown".

- **conditions.type** (string), required

type of the condition. Known conditions are "Approved", "Denied", and "Failed".

An "Approved" condition is added via the /approval subresource, indicating the request was approved and should be issued by the signer.

A "Denied" condition is added via the /approval subresource, indicating the request was denied and should not be issued by the signer.

A "Failed" condition is added via the /status subresource, indicating the signer failed to issue the certificate.

Approved and Denied conditions are mutually exclusive.

Approved, Denied, and Failed conditions cannot be removed once added.

Only one condition of a given type is allowed.

- **conditions.lastTransitionTime** (Time)

lastTransitionTime is the time the condition last transitioned

from one status to another. If unset, when a new condition type is added or an existing condition's status is changed, the server defaults this to the current time.

Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

- **conditions.lastUpdateTime** (Time)

lastUpdateTime is the time of the last update to this condition

Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

- **conditions.message** (string)

message contains a human readable message with details about the request state

- **conditions.reason** (string)

reason indicates a brief reason for the request state

CertificateSigningRequestList

CertificateSigningRequestList is a collection of CertificateSigningRequest objects

- **apiVersion**: certificates.k8s.io/v1
- **kind**: CertificateSigningRequestList
- **metadata** ([ListMeta](#))
- **items** ([\[\]CertificateSigningRequest](#)), required

items is a collection of CertificateSigningRequest objects

Operations

get read the specified CertificateSigningRequest

HTTP Request

GET /apis/certificates.k8s.io/v1/certificatesigningrequests/{name}

Parameters

- **name** (*in path*): string, required
name of the CertificateSigningRequest
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([CertificateSigningRequest](#)): OK

401: Unauthorized

get read approval of the specified CertificateSigningRequest

HTTP Request

GET /apis/certificates.k8s.io/v1/certificatesigningrequests/{name}/approval

Parameters

- **name** (*in path*): string, required
name of the CertificateSigningRequest
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([CertificateSigningRequest](#)): OK

401: Unauthorized

get read status of the specified CertificateSigningRequest

HTTP Request

GET /apis/certificates.k8s.io/v1/certificatesigningrequests/{name}/status

Parameters

- **name** (*in path*): string, required
name of the CertificateSigningRequest
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([CertificateSigningRequest](#)): OK

401: Unauthorized

list list or watch objects of kind CertificateSigningRequest

HTTP Request

GET /apis/certificates.k8s.io/v1/certificatesigningrequests

Parameters

- **allowWatchBookmarks** (*in query*): boolean
[allowWatchBookmarks](#)
- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([CertificateSigningRequestList](#)): OK

401: Unauthorized

create create a CertificateSigningRequest

HTTP Request

POST /apis/certificates.k8s.io/v1/certificatesigningrequests

Parameters

- **body**: [CertificateSigningRequest](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([CertificateSigningRequest](#)): OK

201 ([CertificateSigningRequest](#)): Created

202 ([CertificateSigningRequest](#)): Accepted

401: Unauthorized

update replace the specified CertificateSigningRequest

HTTP Request

PUT /apis/certificates.k8s.io/v1/certificatesigningrequests/{name}

Parameters

- **name** (*in path*): string, required

name of the CertificateSigningRequest

- **body**: [CertificateSigningRequest](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([CertificateSigningRequest](#)): OK

201 ([CertificateSigningRequest](#)): Created

401: Unauthorized

update replace approval of the specified CertificateSigningRequest

HTTP Request

PUT /apis/certificates.k8s.io/v1/certificatesigningrequests/{name}/approval

Parameters

- **name** (*in path*): string, required

name of the CertificateSigningRequest

- **body**: [CertificateSigningRequest](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([CertificateSigningRequest](#)): OK

201 ([CertificateSigningRequest](#)): Created

401: Unauthorized

update replace status of the specified CertificateSigningRequest

HTTP Request

PUT /apis/certificates.k8s.io/v1/certificatesigningrequests/{name}/status

Parameters

- **name** (*in path*): string, required

name of the CertificateSigningRequest

- **body**: [CertificateSigningRequest](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([CertificateSigningRequest](#)): OK

201 ([CertificateSigningRequest](#)): Created

401: Unauthorized

patch partially update the specified CertificateSigningRequest

HTTP Request

PATCH /apis/certificates.k8s.io/v1/certificatesigningrequests/{name}

Parameters

- **name** (*in path*): string, required
name of the CertificateSigningRequest
- **body**: [Patch](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **force** (*in query*): boolean
[force](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([CertificateSigningRequest](#)): OK

201 ([CertificateSigningRequest](#)): Created

401: Unauthorized

patch partially update approval of the specified CertificateSigningRequest

HTTP Request

PATCH /apis/certificates.k8s.io/v1/certificatesigningrequests/{name}/approval

Parameters

- **name** (*in path*): string, required
name of the CertificateSigningRequest
- **body**: [Patch](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([CertificateSigningRequest](#)): OK

201 ([CertificateSigningRequest](#)): Created

401: Unauthorized

patch partially update status of the specified CertificateSigningRequest

HTTP Request

PATCH /apis/certificates.k8s.io/v1/certificatesigningrequests/{name}/status

Parameters

- **name** (*in path*): string, required

name of the CertificateSigningRequest

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([CertificateSigningRequest](#)): OK

201 ([CertificateSigningRequest](#)): Created

401: Unauthorized

delete delete a CertificateSigningRequest

HTTP Request

DELETE /apis/certificates.k8s.io/v1/certificatesigningrequests/{name}

Parameters

- **name** (*in path*): string, required
name of the CertificateSigningRequest
- **body**: [DeleteOptions](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)

Response

200 ([Status](#)): OK

202 ([Status](#)): Accepted

401: Unauthorized

deletecollection delete collection of
CertificateSigningRequest

HTTP Request

DELETE /apis/certificates.k8s.io/v1/certificatesigningrequests

Parameters

- **body**: [DeleteOptions](#)
- **continue** (*in query*): string
[continue](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)

- **propagationPolicy** (*in query*): string
[propagationPolicy](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.4.5 - ClusterTrustBundle v1alpha1

ClusterTrustBundle is a cluster-scoped container for X.

```
apiVersion: certificates.k8s.io/v1alpha1

import "k8s.io/api/certificates/v1alpha1"
```

ClusterTrustBundle

ClusterTrustBundle is a cluster-scoped container for X.509 trust anchors (root certificates).

ClusterTrustBundle objects are considered to be readable by any authenticated user in the cluster, because they can be mounted by pods using the `clusterTrustBundle` projection. All service accounts have read access to ClusterTrustBundles by default. Users who only have namespace-level access to a cluster can read ClusterTrustBundles by impersonating a serviceaccount that they have access to.

It can be optionally associated with a particular assigner, in which case it contains one valid set of trust anchors for that signer. Signers may have multiple associated ClusterTrustBundles; each is an independent set of trust anchors for that signer. Admission control is used to enforce that only users with permissions on the signer can create or modify the corresponding bundle.

- **apiVersion**: certificates.k8s.io/v1alpha1
- **kind**: ClusterTrustBundle
- **metadata** ([ObjectMeta](#))

metadata contains the object metadata.

- **spec** ([ClusterTrustBundleSpec](#)), required

spec contains the signer (if any) and trust anchors.

ClusterTrustBundleSpec

ClusterTrustBundleSpec contains the signer and trust anchors.

- **trustBundle** (string), required

trustBundle contains the individual X.509 trust anchors for this bundle, as PEM bundle of PEM-wrapped, DER-formatted X.509 certificates.

The data must consist only of PEM certificate blocks that parse as valid X.509 certificates. Each certificate must include a basic constraints extension with the CA bit set. The API server will reject objects that contain duplicate certificates, or that use PEM block headers.

Users of ClusterTrustBundles, including Kubelet, are free to reorder and deduplicate certificate blocks in this file according to their own logic, as well as to drop PEM block headers and inter-block data.

- **signerName** (string)

signerName indicates the associated signer, if any.

In order to create or update a ClusterTrustBundle that sets signerName, you must have the following cluster-scoped permission: group=certificates.k8s.io resource=signers resourceName=<the signer name> verb=attest.

If signerName is not empty, then the ClusterTrustBundle object must be named with the signer name as a prefix (translating slashes to colons). For example, for the signer name `example.com/foo`, valid ClusterTrustBundle object names include `example.com:foo:abc` and `example.com:foo:v1`.

If signerName is empty, then the ClusterTrustBundle object's name must not have such a prefix.

List/watch requests for ClusterTrustBundles can filter on this field using a `spec.signerName=NAME` field selector.

ClusterTrustBundleList

ClusterTrustBundleList is a collection of ClusterTrustBundle objects

- **apiVersion**: certificates.k8s.io/v1alpha1
- **kind**: ClusterTrustBundleList
- **metadata** ([ListMeta](#))

metadata contains the list metadata.

- **items** ([][ClusterTrustBundle](#)), required

items is a collection of ClusterTrustBundle objects

Operations

get read the specified ClusterTrustBundle

HTTP Request

GET /apis/certificates.k8s.io/v1alpha1/clustertrustbundles/{name}

Parameters

- **name** (*in path*): string, required

name of the ClusterTrustBundle

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ClusterTrustBundle](#)): OK

401: Unauthorized

list list or watch objects of kind ClusterTrustBundle

HTTP Request

GET /apis/certificates.k8s.io/v1alpha1/clustertrustbundles

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([ClusterTrustBundleList](#)): OK

401: Unauthorized

create create a ClusterTrustBundle

HTTP Request

POST /apis/certificates.k8s.io/v1alpha1/clustertrustbundles

Parameters

- **body**: [ClusterTrustBundle](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ClusterTrustBundle](#)): OK

201 ([ClusterTrustBundle](#)): Created

202 ([ClusterTrustBundle](#)): Accepted

401: Unauthorized

update replace the specified ClusterTrustBundle

HTTP Request

PUT /apis/certificates.k8s.io/v1alpha1/clustertrustbundles/{name}

Parameters

- **name** (*in path*): string, required

name of the ClusterTrustBundle

- **body**: [ClusterTrustBundle](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ClusterTrustBundle](#)): OK

201 ([ClusterTrustBundle](#)): Created

401: Unauthorized

patch partially update the specified ClusterTrustBundle

HTTP Request

PATCH /apis/certificates.k8s.io/v1alpha1/clustertrustbundles/{name}

Parameters

- **name** (*in path*): string, required
name of the ClusterTrustBundle

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ClusterTrustBundle](#)): OK

201 ([ClusterTrustBundle](#)): Created

401: Unauthorized

delete delete a ClusterTrustBundle

HTTP Request

DELETE /apis/certificates.k8s.io/v1alpha1/clustertrustbundles/{name}

Parameters

- **name** (*in path*): string, required
name of the ClusterTrustBundle

- **body**: [DeleteOptions](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

Response

200 ([Status](#)): OK

202 ([Status](#)): Accepted

401: Unauthorized

deletecollection delete collection of ClusterTrustBundle

HTTP Request

DELETE /apis/certificates.k8s.io/v1alpha1/clustertrustbundles

Parameters

- **body**: [DeleteOptions](#)
- **continue** (*in query*): string
[continue](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.4.6 - SelfSubjectReview

SelfSubjectReview contains the user information that the kube-apiserver has about the user making this request.

```
apiVersion: authentication.k8s.io/v1

import "k8s.io/api/authentication/v1"
```

SelfSubjectReview

SelfSubjectReview contains the user information that the kube-apiserver has about the user making this request. When using impersonation, users will receive the user info of the user being impersonated. If impersonation or request header authentication is used, any extra keys will have their case ignored and returned as lowercase.

- **apiVersion**: authentication.k8s.io/v1
- **kind**: SelfSubjectReview
- **metadata** ([ObjectMeta](#))

Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **status** ([SelfSubjectReviewStatus](#))

Status is filled in by the server with the user attributes.

SelfSubjectReviewStatus

SelfSubjectReviewStatus is filled by the kube-apiserver and sent back to a user.

- **userInfo** ([UserInfo](#))

User attributes of the user making this request.

UserInfo holds the information about the user needed to implement the user.Info interface.

- **userInfo.extra** (map[string][]string)

Any additional information provided by the authenticator.

- **userInfo.groups** ([]string)

The names of groups this user is a part of.

- **userInfo.uid** (string)

A unique value that identifies this user across time. If this user is deleted and another user by the same name is added, they will have different UIDs.

- **userInfo.username** (string)

The name that uniquely identifies this user among all active users.

Operations

create create a SelfSubjectReview

HTTP Request

POST /apis/authentication.k8s.io/v1/selfsubjectreviews

Parameters

- **body**: [SelfSubjectReview](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([SelfSubjectReview](#)): OK

201 ([SelfSubjectReview](#)): Created

202 ([SelfSubjectReview](#)): Accepted

401: Unauthorized

5.5 - Authorization Resources

5.5.1 - LocalSubjectAccessReview

LocalSubjectAccessReview checks whether or not a user or group can perform an action in a given namespace.

```
apiVersion: authorization.k8s.io/v1

import "k8s.io/api/authorization/v1"
```

LocalSubjectAccessReview

LocalSubjectAccessReview checks whether or not a user or group can perform an action in a given namespace. Having a namespace scoped resource makes it much easier to grant namespace scoped policy that includes permissions checking.

- **apiVersion**: authorization.k8s.io/v1
- **kind**: LocalSubjectAccessReview
- **metadata** ([ObjectMeta](#))

Standard list metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **spec** ([SubjectAccessReviewSpec](#)), required

Spec holds information about the request being evaluated. spec.namespace must be equal to the namespace you made the request against. If empty, it is defaulted.

- **status** ([SubjectAccessReviewStatus](#))

Status is filled in by the server and indicates whether the request is allowed or not

Operations

create create a LocalSubjectAccessReview

HTTP Request

POST /apis/authorization.k8s.io/v1/namespaces/{namespace}/localsubjectaccessreviews

Parameters

- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [LocalSubjectAccessReview](#), required

- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

- 200 ([LocalSubjectAccessReview](#)): OK
201 ([LocalSubjectAccessReview](#)): Created
202 ([LocalSubjectAccessReview](#)): Accepted
401: Unauthorized

5.5.2 - SelfSubjectAccessReview

SelfSubjectAccessReview checks whether or the current user can perform an action.

```
apiVersion: authorization.k8s.io/v1  
import "k8s.io/api/authorization/v1"
```

SelfSubjectAccessReview

SelfSubjectAccessReview checks whether or the current user can perform an action. Not filling in a spec.namespace means "in all namespaces". Self is a special case, because users should always be able to check whether they can perform an action

- **apiVersion**: authorization.k8s.io/v1

- **kind**: SelfSubjectAccessReview

- **metadata** ([ObjectMeta](#))

Standard list metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **spec** ([SelfSubjectAccessReviewSpec](#)), required

Spec holds information about the request being evaluated. user and groups must be empty

- **status** ([SubjectAccessReviewStatus](#))

Status is filled in by the server and indicates whether the request is allowed or not

SelfSubjectAccessReviewSpec

SelfSubjectAccessReviewSpec is a description of the access request.

Exactly one of ResourceAuthorizationAttributes and

NonResourceAuthorizationAttributes must be set

- **nonResourceAttributes** (NonResourceAttributes)

NonResourceAttributes describes information for a non-resource access request

NonResourceAttributes includes the authorization attributes available for non-resource requests to the Authorizer interface

- **nonResourceAttributes.path** (string)

Path is the URL path of the request

- **nonResourceAttributes.verb** (string)

Verb is the standard HTTP verb

- **resourceAttributes** (ResourceAttributes)

ResourceAuthorizationAttributes describes information for a resource access request

ResourceAttributes includes the authorization attributes available for resource requests to the Authorizer interface

- **resourceAttributes.group** (string)

Group is the API Group of the Resource. "*" means all.

- **resourceAttributes.name** (string)

Name is the name of the resource being requested for a "get" or deleted for a "delete". "" (empty) means all.

- **resourceAttributes.namespace** (string)

Namespace is the namespace of the action being requested. Currently, there is no distinction between no namespace and all namespaces "" (empty) is defaulted for LocalSubjectAccessReviews "" (empty) is empty for cluster-scoped resources "" (empty) means "all" for namespace scoped resources from a SubjectAccessReview or SelfSubjectAccessReview

- **resourceAttributes.resource** (string)

Resource is one of the existing resource types. "*" means all.

- **resourceAttributes.subresource** (string)

Subresource is one of the existing resource types. "" means none.

- **resourceAttributes.verb** (string)

Verb is a kubernetes resource API verb, like: get, list, watch, create, update, delete, proxy. "*" means all.

- **resourceAttributes.version** (string)

Version is the API Version of the Resource. "*" means all.

Operations

create create a SelfSubjectAccessReview

HTTP Request

POST /apis/authorization.k8s.io/v1/selfsubjectaccessreviews

Parameters

- **body**: [SelfSubjectAccessReview](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([SelfSubjectAccessReview](#)): OK

201 ([SelfSubjectAccessReview](#)): Created

202 ([SelfSubjectAccessReview](#)): Accepted

401: Unauthorized

5.5.3 - SelfSubjectRulesReview

SelfSubjectRulesReview enumerates the set of actions the current user can perform within a namespace.

```
apiVersion: authorization.k8s.io/v1

import "k8s.io/api/authorization/v1"
```

SelfSubjectRulesReview

SelfSubjectRulesReview enumerates the set of actions the current user can perform within a namespace. The returned list of actions may be incomplete depending on the server's authorization mode, and any errors experienced during the evaluation. SelfSubjectRulesReview should be used by UIs to show/hide actions, or to quickly let an end user reason about their permissions. It should NOT Be used by external systems to drive authorization decisions as this raises confused deputy, cache lifetime/revocation, and correctness concerns. SubjectAccessReview, and LocalAccessReview are the correct way to defer authorization decisions to the API server.

- **apiVersion**: authorization.k8s.io/v1

- **kind**: SelfSubjectRulesReview

- **metadata** ([ObjectMeta](#))

Standard list metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **spec** ([SelfSubjectRulesReviewSpec](#)), required

Spec holds information about the request being evaluated.

- **status** ([SubjectRulesReviewStatus](#))

Status is filled in by the server and indicates the set of actions a user can perform.

SubjectRulesReviewStatus contains the result of a rules check. This check can be incomplete depending on the set of authorizers the server is configured with and any errors experienced during evaluation. Because authorization rules are additive, if a rule appears in a list it's safe to assume the subject has that permission, even if that list is incomplete.

- **status.incomplete** (boolean), required

Incomplete is true when the rules returned by this call are incomplete. This is most commonly encountered when an authorizer, such as an external authorizer, doesn't support rules evaluation.

- **status.nonResourceRules** ([]NonResourceRule), required

NonResourceRules is the list of actions the subject is allowed to perform on non-resources. The list ordering isn't significant, may contain duplicates, and possibly be incomplete.

NonResourceRule holds information that describes a rule for the non-resource

- **status.nonResourceRules.verbs** ([]string), required

Verb is a list of kubernetes non-resource API verbs, like: get, post, put, delete, patch, head, options. "*" means all.

- **status.nonResourceRules.nonResourceURLs** ([]string)

NonResourceURLs is a set of partial urls that a user should have access to. *s are allowed, but only as the full, final step in the path.* "" means all.

- **status.resourceRules** ([]ResourceRule), required

ResourceRules is the list of actions the subject is allowed to perform on resources. The list ordering isn't significant, may contain duplicates, and possibly be incomplete.

ResourceRule is the list of actions the subject is allowed to perform on resources. The list ordering isn't significant, may contain duplicates, and possibly be incomplete.

- **status.resourceRules.verbs** ([]string), required

Verb is a list of kubernetes resource API verbs, like: get, list, watch, create, update, delete, proxy. "*" means all.

- **status.resourceRules.apiGroups** ([]string)

APIGroups is the name of the APIGroup that contains the resources. If multiple API groups are specified, any action requested against one of the enumerated resources in any API group will be allowed. "*" means all.

- **status.resourceRules.resourceNames** ([]string)

ResourceNames is an optional white list of names that the rule applies to. An empty set means that everything is allowed. "*" means all.

- **status.resourceRules.resources** ([]string)

Resources is a list of resources this rule applies to. "" means all in the specified apiGroups. "/foo" represents the subresource 'foo' for all resources in the specified apiGroups.

- **status.evaluationError** (string)

EvaluationError can appear in combination with Rules. It indicates an error occurred during rule evaluation, such as an authorizer that doesn't support rule evaluation, and that ResourceRules and/or NonResourceRules may be incomplete.

SelfSubjectRulesReviewSpec

SelfSubjectRulesReviewSpec defines the specification for SelfSubjectRulesReview.

- **namespace** (string)

Namespace to evaluate rules for. Required.

Operations

create create a SelfSubjectRulesReview

HTTP Request

POST /apis/authorization.k8s.io/v1/selfsubjectrulesreviews

Parameters

- **body**: [SelfSubjectRulesReview](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([SelfSubjectRulesReview](#)): OK

201 ([SelfSubjectRulesReview](#)): Created

202 ([SelfSubjectRulesReview](#)): Accepted

401: Unauthorized

5.5.4 - SubjectAccessReview

SubjectAccessReview checks whether or not a user or group can perform an action.

```
apiVersion: authorization.k8s.io/v1
import "k8s.io/api/authorization/v1"
```

SubjectAccessReview

SubjectAccessReview checks whether or not a user or group can perform an action.

- **apiVersion**: authorization.k8s.io/v1
- **kind**: SubjectAccessReview
- **metadata** ([ObjectMeta](#))

Standard list metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **spec** ([SubjectAccessReviewSpec](#)), required

Spec holds information about the request being evaluated

- **status** ([SubjectAccessReviewStatus](#))

Status is filled in by the server and indicates whether the request is allowed or not

SubjectAccessReviewSpec

SubjectAccessReviewSpec is a description of the access request. Exactly one of ResourceAuthorizationAttributes and NonResourceAuthorizationAttributes must be set

- **extra** (map[string][]string)

Extra corresponds to the user.Info.GetExtra() method from the authenticator. Since that is input to the authorizer it needs a reflection here.

- **groups** ([]string)

Groups is the groups you're testing for.

- **nonResourceAttributes** (NonResourceAttributes)

NonResourceAttributes describes information for a non-resource access request

NonResourceAttributes includes the authorization attributes available for non-resource requests to the Authorizer interface

- **nonResourceAttributes.path** (string)

Path is the URL path of the request

- **nonResourceAttributes.verb** (string)

Verb is the standard HTTP verb

- **resourceAttributes** (ResourceAttributes)

ResourceAuthorizationAttributes describes information for a resource access request

ResourceAttributes includes the authorization attributes available for resource requests to the Authorizer interface

- **resourceAttributes.group** (string)

Group is the API Group of the Resource. "*" means all.

- **resourceAttributes.name** (string)

Name is the name of the resource being requested for a "get" or deleted for a "delete". "" (empty) means all.

- **resourceAttributes.namespace** (string)

Namespace is the namespace of the action being requested. Currently, there is no distinction between no namespace and all namespaces "" (empty) is defaulted for LocalSubjectAccessReviews "" (empty) is empty for cluster-scoped resources "" (empty) means "all" for namespace scoped resources from a SubjectAccessReview or SelfSubjectAccessReview

- **resourceAttributes.resource** (string)

Resource is one of the existing resource types. "*" means all.

- **resourceAttributes.subresource** (string)

Subresource is one of the existing resource types. "" means none.

- **resourceAttributes.verb** (string)

Verb is a kubernetes resource API verb, like: get, list, watch, create, update, delete, proxy. "*" means all.

- **resourceAttributes.version** (string)

Version is the API Version of the Resource. "*" means all.

- **uid** (string)

UID information about the requesting user.

- **user** (string)

User is the user you're testing for. If you specify "User" but not "Groups", then is it interpreted as "What if User were not a member of any groups

SubjectAccessReviewStatus

SubjectAccessReviewStatus

- **allowed** (boolean), required

Allowed is required. True if the action would be allowed, false otherwise.

- **denied** (boolean)

Denied is optional. True if the action would be denied, otherwise false. If both allowed is false and denied is false, then the authorizer

has no opinion on whether to authorize the action. Denied may not be true if Allowed is true.

- **evaluationError** (string)

EvaluationError is an indication that some error occurred during the authorization check. It is entirely possible to get an error and be able to continue determine authorization status in spite of it. For instance, RBAC can be missing a role, but enough roles are still present and bound to reason about the request.

- **reason** (string)

Reason is optional. It indicates why a request was allowed or denied.

Operations

create create a SubjectAccessReview

HTTP Request

POST /apis/authorization.k8s.io/v1/subjectaccessreviews

Parameters

- **body**: [SubjectAccessReview](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([SubjectAccessReview](#)): OK

201 ([SubjectAccessReview](#)): Created

202 ([SubjectAccessReview](#)): Accepted

401: Unauthorized

5.5.5 - ClusterRole

ClusterRole is a cluster level, logical grouping of PolicyRules that can be referenced as a unit by a RoleBinding or ClusterRoleBinding.

```
apiVersion: rbac.authorization.k8s.io/v1  
import "k8s.io/api/rbac/v1"
```

ClusterRole

ClusterRole is a cluster level, logical grouping of PolicyRules that can be referenced as a unit by a RoleBinding or ClusterRoleBinding.

- **apiVersion**: rbac.authorization.k8s.io/v1
- **kind**: ClusterRole
- **metadata** ([ObjectMeta](#))

Standard object's metadata.

- **aggregationRule** (AggregationRule)

AggregationRule is an optional field that describes how to build the Rules for this ClusterRole. If AggregationRule is set, then the Rules are controller managed and direct changes to Rules will be stomped by the controller.

AggregationRule describes how to locate ClusterRoles to aggregate into the ClusterRole

- **aggregationRule.clusterRoleSelectors** ([][LabelSelector](#))

ClusterRoleSelectors holds a list of selectors which will be used to find ClusterRoles and create the rules. If any of the selectors match, then the ClusterRole's permissions will be added

- **rules** ([][PolicyRule](#))

Rules holds all the PolicyRules for this ClusterRole

PolicyRule holds information that describes a policy rule, but does not contain information about who the rule applies to or which namespace the rule applies to.

- **rules.apiGroups** ([]string)

APIGroups is the name of the APIGroup that contains the resources. If multiple API groups are specified, any action requested against one of the enumerated resources in any API group will be allowed. "" represents the core API group and "*" represents all API groups.

- **rules.resources** ([]string)

Resources is a list of resources this rule applies to. '*' represents all resources.

- **rules.verbs** ([]string), required

Verbs is a list of Verbs that apply to ALL the ResourceKinds contained in this rule. '*' represents all verbs.

- **rules.resourceNames** ([]string)

ResourceNames is an optional white list of names that the rule applies to. An empty set means that everything is allowed.

- **rules.nonResourceURLs** ([]string)

NonResourceURLs is a set of partial urls that a user should have access to. *s are allowed, but only as the full, final step in the path. Since non-resource URLs are not namespaced, this field is only applicable for ClusterRoles referenced from a ClusterRoleBinding. Rules can either apply to API resources (such as "pods" or "secrets") or non-resource URL paths (such as "/api"), but not both.

ClusterRoleList

ClusterRoleList is a collection of ClusterRoles

- **apiVersion**: rbac.authorization.k8s.io/v1

- **kind**: ClusterRoleList

- **metadata** ([ListMeta](#))

Standard object's metadata.

- **items** ([][ClusterRole](#)), required

Items is a list of ClusterRoles

Operations

get read the specified ClusterRole

HTTP Request

GET /apis/rbac.authorization.k8s.io/v1/clusterroles/{name}

Parameters

- **name** (*in path*): string, required

name of the ClusterRole

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ClusterRole](#)): OK

401: Unauthorized

list list or watch objects of kind ClusterRole

HTTP Request

GET /apis/rbac.authorization.k8s.io/v1/clusterroles

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([ClusterRoleList](#)): OK

401: Unauthorized

create create a ClusterRole

HTTP Request

POST /apis/rbac.authorization.k8s.io/v1/clusterroles

Parameters

- **body**: [ClusterRole](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ClusterRole](#)): OK

201 ([ClusterRole](#)): Created

202 ([ClusterRole](#)): Accepted

401: Unauthorized

update replace the specified ClusterRole

HTTP Request

PUT /apis/rbac.authorization.k8s.io/v1/clusterroles/{name}

Parameters

- **name** (*in path*): string, required

name of the ClusterRole

- **body**: [ClusterRole](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ClusterRole](#)): OK

201 ([ClusterRole](#)): Created

401: Unauthorized

patch partially update the specified ClusterRole

HTTP Request

PATCH /apis/rbac.authorization.k8s.io/v1/clusterroles/{name}

Parameters

- **name** (*in path*): string, required

name of the ClusterRole

- **body**: [Patch](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **force** (*in query*): boolean
[force](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([ClusterRole](#)): OK

201 ([ClusterRole](#)): Created

401: Unauthorized

delete delete a ClusterRole

HTTP Request

DELETE /apis/rbac.authorization.k8s.io/v1/clusterroles/{name}

Parameters

- **name** (*in path*): string, required
name of the ClusterRole
- **body**: [DeleteOptions](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)

Response

200 ([Status](#)): OK

202 ([Status](#)): Accepted

401: Unauthorized

deletecollection delete collection of ClusterRole

HTTP Request

DELETE /apis/rbac.authorization.k8s.io/v1/clusterroles

Parameters

- **body**: [DeleteOptions](#)
- **continue** (*in query*): string
[continue](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.5.6 - ClusterRoleBinding

ClusterRoleBinding references a ClusterRole, but not contain it.

```
apiVersion: rbac.authorization.k8s.io/v1  
import "k8s.io/api/rbac/v1"
```

ClusterRoleBinding

ClusterRoleBinding references a ClusterRole, but not contain it. It can reference a ClusterRole in the global namespace, and adds who information via Subject.

- **apiVersion**: rbac.authorization.k8s.io/v1
- **kind**: ClusterRoleBinding
- **metadata** ([ObjectMeta](#))

Standard object's metadata.

- **roleRef** (RoleRef), required

RoleRef can only reference a ClusterRole in the global namespace. If the RoleRef cannot be resolved, the Authorizer must return an error. This field is immutable.

RoleRef contains information that points to the role being used

- **roleRef.apiGroup** (string), required

APIGroup is the group for the resource being referenced

- **roleRef.kind** (string), required

Kind is the type of resource being referenced

- **roleRef.name** (string), required

Name is the name of resource being referenced

- **subjects** ([]Subject)

Subjects holds references to the objects the role applies to.

Subject contains a reference to the object or user identities a role binding applies to. This can either hold a direct API object reference, or a value for non-objects such as user and group names.

- **subjects.kind** (string), required

Kind of object being referenced. Values defined by this API group are "User", "Group", and "ServiceAccount". If the Authorizer does not recognize the kind value, the Authorizer should report an error.

- **subjects.name** (string), required

Name of the object being referenced.

- **subjects.apiGroup** (string)

APIGroup holds the API group of the referenced subject. Defaults to "" for ServiceAccount subjects. Defaults to

"rbac.authorization.k8s.io" for User and Group subjects.

- **subjects.namespace** (string)

Namespace of the referenced object. If the object kind is non-namespace, such as "User" or "Group", and this value is not empty the Authorizer should report an error.

ClusterRoleBindingList

ClusterRoleBindingList is a collection of ClusterRoleBindings

- **apiVersion**: rbac.authorization.k8s.io/v1
- **kind**: ClusterRoleBindingList
- **metadata** ([ListMeta](#))

Standard object's metadata.

- **items** ([][ClusterRoleBinding](#)), required

Items is a list of ClusterRoleBindings

Operations

get read the specified ClusterRoleBinding

HTTP Request

GET /apis/rbac.authorization.k8s.io/v1/clusterrolebindings/{name}

Parameters

- **name** (*in path*): string, required

name of the ClusterRoleBinding

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ClusterRoleBinding](#)): OK

401: Unauthorized

list list or watch objects of kind ClusterRoleBinding

HTTP Request

GET /apis/rbac.authorization.k8s.io/v1/clusterrolebindings

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string
[continue](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)
- **watch** (*in query*): boolean
[watch](#)

Response

200 ([ClusterRoleBindingList](#)): OK

401: Unauthorized

create create a ClusterRoleBinding

HTTP Request

POST /apis/rbac.authorization.k8s.io/v1/clusterrolebindings

Parameters

- **body**: [ClusterRoleBinding](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

[pretty](#)

Response

200 ([ClusterRoleBinding](#)): OK

201 ([ClusterRoleBinding](#)): Created

202 ([ClusterRoleBinding](#)): Accepted

401: Unauthorized

update replace the specified ClusterRoleBinding

HTTP Request

PUT /apis/rbac.authorization.k8s.io/v1/clusterrolebindings/{name}

Parameters

- **name** (*in path*): string, required

name of the ClusterRoleBinding

- **body**: [ClusterRoleBinding](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ClusterRoleBinding](#)): OK

201 ([ClusterRoleBinding](#)): Created

401: Unauthorized

patch partially update the specified ClusterRoleBinding

HTTP Request

PATCH /apis/rbac.authorization.k8s.io/v1/clusterrolebindings/{name}

Parameters

- **name** (*in path*): string, required

name of the ClusterRoleBinding

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ClusterRoleBinding](#)): OK201 ([ClusterRoleBinding](#)): Created

401: Unauthorized

delete delete a ClusterRoleBinding

HTTP Request

DELETE /apis/rbac.authorization.k8s.io/v1/clusterrolebindings/{name}

Parameters

- **name** (*in path*): string, required

name of the ClusterRoleBinding

- **body**: [DeleteOptions](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

Response

200 ([Status](#)): OK202 ([Status](#)): Accepted

401: Unauthorized

deletecollection delete collection of ClusterRoleBinding

HTTP Request

DELETE /apis/rbac.authorization.k8s.io/v1/clusterrolebindings

Parameters

- **body**: [DeleteOptions](#)
- **continue** (*in query*): string
[continue](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.5.7 - Role

Role is a namespaced, logical grouping of PolicyRules that can be referenced as a unit by a RoleBinding.

```
apiVersion: rbac.authorization.k8s.io/v1
import "k8s.io/api/rbac/v1"
```

Role

Role is a namespaced, logical grouping of PolicyRules that can be referenced as a unit by a RoleBinding.

- **apiVersion**: rbac.authorization.k8s.io/v1

- **kind**: Role

- **metadata** ([ObjectMeta](#))

Standard object's metadata.

- **rules** ([]PolicyRule)

Rules holds all the PolicyRules for this Role

PolicyRule holds information that describes a policy rule, but does not contain information about who the rule applies to or which namespace the rule applies to.

- **rules.apiGroups** ([]string)

APIGroups is the name of the APIGroup that contains the resources. If multiple API groups are specified, any action requested against one of the enumerated resources in any API group will be allowed. "" represents the core API group and "*" represents all API groups.

- **rules.resources** ([]string)

Resources is a list of resources this rule applies to. '*' represents all resources.

- **rules.verbs** ([]string), required

Verbs is a list of Verbs that apply to ALL the ResourceKinds contained in this rule. '*' represents all verbs.

- **rules.resourceNames** ([]string)

ResourceNames is an optional white list of names that the rule applies to. An empty set means that everything is allowed.

- **rules.nonResourceURLs** ([]string)

NonResourceURLs is a set of partial urls that a user should have access to. *s are allowed, but only as the full, final step in the path. Since non-resource URLs are not namespaced, this field is only applicable for ClusterRoles referenced from a ClusterRoleBinding. Rules can either apply to API resources (such as "pods" or "secrets") or non-resource URL paths (such as "/api"), but not both.

RoleList

RoleList is a collection of Roles

- **apiVersion**: rbac.authorization.k8s.io/v1

- **kind**: RoleList

- **metadata** ([ListMeta](#))

Standard object's metadata.

- **items** ([][Role](#)), required

Items is a list of Roles

Operations

get read the specified Role

HTTP Request

```
GET /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles/{name}
```

Parameters

- **name** (*in path*): string, required

name of the Role

- **namespace** (*in path*): string, required

[namespace](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Role](#)): OK

401: Unauthorized

list list or watch objects of kind Role

HTTP Request

```
GET /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles
```

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([RoleList](#)): OK

401: Unauthorized

list list or watch objects of kind Role

HTTP Request

GET /apis/rbac.authorization.k8s.io/v1/roles

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string
[pretty](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)
- **watch** (*in query*): boolean
[watch](#)

Response

200 ([RoleList](#)): OK

401: Unauthorized

create create a Role

HTTP Request

POST /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles

Parameters

- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [Role](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([Role](#)): OK

201 ([Role](#)): Created

202 ([Role](#)): Accepted

401: Unauthorized

update replace the specified Role

HTTP Request

PUT /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles/{name}

Parameters

- **name** (*in path*): string, required
name of the Role
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [Role](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([Role](#)): OK

201 ([Role](#)): Created

401: Unauthorized

patch partially update the specified Role

HTTP Request

PATCH /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles/{name}

Parameters

- **name** (*in path*): string, required
name of the Role
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [Patch](#), required
- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Role](#)): OK

201 ([Role](#)): Created

401: Unauthorized

delete delete a Role

HTTP Request

DELETE /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles/{name}

Parameters

- **name** (*in path*): string, required

name of the Role

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [DeleteOptions](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

Response

200 ([Status](#)): OK

202 ([Status](#)): Accepted

401: Unauthorized

deletecollection delete collection of Role

HTTP Request

DELETE /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [DeleteOptions](#)

- **continue** (*in query*): string

[continue](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.5.8 - RoleBinding

RoleBinding references a role, but does not contain it.

```
apiVersion: rbac.authorization.k8s.io/v1

import "k8s.io/api/rbac/v1"
```

RoleBinding

RoleBinding references a role, but does not contain it. It can reference a Role in the same namespace or a ClusterRole in the global namespace. It adds who information via Subjects and namespace information by which namespace it exists in. RoleBindings in a given namespace only have effect in that namespace.

- **apiVersion**: rbac.authorization.k8s.io/v1

- **kind**: RoleBinding

- **metadata** ([ObjectMeta](#))

Standard object's metadata.

- **roleRef** (RoleRef), required

RoleRef can reference a Role in the current namespace or a ClusterRole in the global namespace. If the RoleRef cannot be resolved, the Authorizer must return an error. This field is immutable.

RoleRef contains information that points to the role being used

- **roleRef.apiGroup** (string), required

APIGroup is the group for the resource being referenced

- **roleRef.kind** (string), required

Kind is the type of resource being referenced

- **roleRef.name** (string), required

Name is the name of resource being referenced

- **subjects** ([]Subject)

Subjects holds references to the objects the role applies to.

Subject contains a reference to the object or user identities a role binding applies to. This can either hold a direct API object reference, or a value for non-objects such as user and group names.

- **subjects.kind** (string), required

Kind of object being referenced. Values defined by this API group are "User", "Group", and "ServiceAccount". If the Authorizer does not recognize the kind value, the Authorizer should report an error.

- **subjects.name** (string), required

Name of the object being referenced.

- **subjects.apiGroup** (string)

APIGroup holds the API group of the referenced subject.

Defaults to "" for ServiceAccount subjects. Defaults to "rbac.authorization.k8s.io" for User and Group subjects.

- **subjects.namespace** (string)

Namespace of the referenced object. If the object kind is non-namespace, such as "User" or "Group", and this value is not empty the Authorizer should report an error.

RoleBindingList

RoleBindingList is a collection of RoleBindings

- **apiVersion**: rbac.authorization.k8s.io/v1

- **kind**: RoleBindingList

- **metadata** ([ListMeta](#))

Standard object's metadata.

- **items** ([][RoleBinding](#)), required

Items is a list of RoleBindings

Operations

get read the specified RoleBinding

HTTP Request

GET /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/rolebindings/{name}

Parameters

- **name** (*in path*): string, required

name of the RoleBinding

- **namespace** (*in path*): string, required

[namespace](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([RoleBinding](#)): OK

401: Unauthorized

list list or watch objects of kind RoleBinding

HTTP Request

GET /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/rolebindings

Parameters

- **namespace** (*in path*): string, required
[namespace](#)
- **allowWatchBookmarks** (*in query*): boolean
[allowWatchBookmarks](#)
- **continue** (*in query*): string
[continue](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)
- **watch** (*in query*): boolean
[watch](#)

Response

200 ([RoleBindingList](#)): OK

401: Unauthorized

list list or watch objects of kind RoleBinding

HTTP Request

GET /apis/rbac.authorization.k8s.io/v1/rolebindings

Parameters

- **allowWatchBookmarks** (*in query*): boolean
[allowWatchBookmarks](#)
- **continue** (*in query*): string
[continue](#)

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([RoleBindingList](#)): OK

401: Unauthorized

create create a RoleBinding

HTTP Request

POST /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/rolebindings

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [RoleBinding](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([RoleBinding](#)): OK

201 ([RoleBinding](#)): Created

202 ([RoleBinding](#)): Accepted

401: Unauthorized

update replace the specified RoleBinding

HTTP Request

PUT /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/rolebindings/{name}

Parameters

- **name** (*in path*): string, required
 - name of the RoleBinding
- **namespace** (*in path*): string, required
 - [namespace](#)
- **body**: [RoleBinding](#), required
- **dryRun** (*in query*): string
 - [dryRun](#)
- **fieldManager** (*in query*): string
 - [fieldManager](#)
- **fieldValidation** (*in query*): string
 - [fieldValidation](#)
- **pretty** (*in query*): string
 - [pretty](#)

Response

200 ([RoleBinding](#)): OK

201 ([RoleBinding](#)): Created

401: Unauthorized

patch partially update the specified RoleBinding

HTTP Request

PATCH /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/rolebindings/{name}

Parameters

- **name** (*in path*): string, required
name of the RoleBinding
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [Patch](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **force** (*in query*): boolean
[force](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([RoleBinding](#)): OK

201 ([RoleBinding](#)): Created

401: Unauthorized

delete delete a RoleBinding

HTTP Request

DELETE /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/rolebindings/{name}

Parameters

- **name** (*in path*): string, required
name of the RoleBinding
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [DeleteOptions](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

Response

200 ([Status](#)): OK

202 ([Status](#)): Accepted

401: Unauthorized

deletecollection delete collection of RoleBinding

HTTP Request

DELETE /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/rolebindings

Parameters

- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [DeleteOptions](#)
- **continue** (*in query*): string
[continue](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.6 - Policy Resources

5.6.1 - LimitRange

LimitRange sets resource usage limits for each kind of resource in a Namespace.

```
apiVersion: v1

import "k8s.io/api/core/v1"
```

LimitRange

LimitRange sets resource usage limits for each kind of resource in a Namespace.

- **apiVersion**: v1
- **kind**: LimitRange
- **metadata** ([ObjectMeta](#))
Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>
- **spec** ([LimitRangeSpec](#))
Spec defines the limits enforced. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status>

LimitRangeSpec

LimitRangeSpec defines a min/max usage limit for resources that match on kind.

- **limits** ([]LimitRangeItem), required
Limits is the list of LimitRangeItem objects that are enforced.
LimitRangeItem defines a min/max usage limit for any resource that matches on kind.
 - **limits.type** (string), required
Type of resource that this limit applies to.
 - **limits.default** (map[string][Quantity](#))
Default resource requirement limit value by resource name if resource limit is omitted.
 - **limits.defaultRequest** (map[string][Quantity](#))
DefaultRequest is the default resource requirement request value by resource name if resource request is omitted.
 - **limits.max** (map[string][Quantity](#))
Max usage constraints on this kind by resource name.
 - **limits.maxLimitRequestRatio** (map[string][Quantity](#))

MaxLimitRequestRatio if specified, the named resource must have a request and limit that are both non-zero where limit divided by request is less than or equal to the enumerated value; this represents the max burst for the named resource.

- **limits.min** (map[string][Quantity](#))

Min usage constraints on this kind by resource name.

LimitRangeList

LimitRangeList is a list of LimitRange items.

- **apiVersion**: v1
- **kind**: LimitRangeList
- **metadata** ([ListMeta](#))
Standard list metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds>
- **items** ([][LimitRange](#)), required
Items is a list of LimitRange objects. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Operations

get read the specified LimitRange

HTTP Request

GET /api/v1/namespaces/{namespace}/limitranges/{name}

Parameters

- **name** (*in path*): string, required
name of the LimitRange
- **namespace** (*in path*): string, required
[namespace](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([LimitRange](#)): OK

401: Unauthorized

list list or watch objects of kind LimitRange

HTTP Request

GET /api/v1/namespaces/{namespace}/limitranges

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([LimitRangeList](#)): OK

401: Unauthorized

list list or watch objects of kind LimitRange

HTTP Request

GET /api/v1/limitranges

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string
[continue](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)
- **watch** (*in query*): boolean
[watch](#)

Response

200 ([LimitRangeList](#)): OK

401: Unauthorized

create create a LimitRange

HTTP Request

POST /api/v1/namespaces/{namespace}/limitranges

Parameters

- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [LimitRange](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([LimitRange](#)): OK201 ([LimitRange](#)): Created202 ([LimitRange](#)): Accepted

401: Unauthorized

update replace the specified LimitRange

HTTP Request

PUT /api/v1/namespaces/{namespace}/limitranges/{name}

Parameters

- **name** (*in path*): string, required

name of the LimitRange

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [LimitRange](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([LimitRange](#)): OK201 ([LimitRange](#)): Created

401: Unauthorized

patch partially update the specified LimitRange

HTTP Request

PATCH /api/v1/namespaces/{namespace}/limitranges/{name}

Parameters

- **name** (*in path*): string, required
name of the LimitRange
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [Patch](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **force** (*in query*): boolean
[force](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([LimitRange](#)): OK

201 ([LimitRange](#)): Created

401: Unauthorized

delete delete a LimitRange

HTTP Request

DELETE /api/v1/namespaces/{namespace}/limitranges/{name}

Parameters

- **name** (*in path*): string, required
name of the LimitRange
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [DeleteOptions](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

Response

200 ([Status](#)): OK

202 ([Status](#)): Accepted

401: Unauthorized

deletecollection delete collection of LimitRange

HTTP Request

DELETE /api/v1/namespaces/{namespace}/limitranges

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [DeleteOptions](#)

- **continue** (*in query*): string

[continue](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.6.2 - ResourceQuota

ResourceQuota sets aggregate quota restrictions enforced per namespace.

```
apiVersion: v1

import "k8s.io/api/core/v1"
```

ResourceQuota

ResourceQuota sets aggregate quota restrictions enforced per namespace

- **apiVersion**: v1

- **kind**: ResourceQuota

- **metadata** ([ObjectMeta](#))

Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **spec** ([ResourceQuotaSpec](#))

Spec defines the desired quota. <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status>

- **status** ([ResourceQuotaStatus](#))

Status defines the actual enforced quota and its current usage. <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status>

ResourceQuotaSpec

ResourceQuotaSpec defines the desired hard limits to enforce for Quota.

- **hard** ([map\[string\]Quantity](#))

hard is the set of desired hard limits for each named resource.

More info: <https://kubernetes.io/docs/concepts/policy/resource-quotas/>

- **scopeSelector** ([ScopeSelector](#))

scopeSelector is also a collection of filters like scopes that must match each object tracked by a quota but expressed using ScopeSelectorOperator in combination with possible values. For a resource to match, both scopes AND scopeSelector (if specified in spec), must be matched.

A scope selector represents the AND of the selectors represented by the scoped-resource selector requirements.

- **scopeSelector.matchExpressions**
([][ScopedResourceSelectorRequirement](#))

A list of scope selector requirements by scope of the resources.

A scoped-resource selector requirement is a selector that contains values, a scope name, and an operator that relates the scope name and values.

- **scopeSelector.matchExpressions.operator** (string), required

Represents a scope's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist.

- **scopeSelector.matchExpressions.scopeName** (string), required

The name of the scope that the selector applies to.

- **scopeSelector.matchExpressions.values** ([]string)

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.

- **scopes** ([]string)

A collection of filters that must match each object tracked by a quota. If not specified, the quota matches all objects.

ResourceQuotaStatus

ResourceQuotaStatus defines the enforced hard limits and observed use.

- **hard** (map[string][Quantity](#))

Hard is the set of enforced hard limits for each named resource.

More info: <https://kubernetes.io/docs/concepts/policy/resource-quotas/>

- **used** (map[string][Quantity](#))

Used is the current observed total usage of the resource in the namespace.

ResourceQuotaList

ResourceQuotaList is a list of ResourceQuota items.

- **apiVersion**: v1

- **kind**: ResourceQuotaList

- **metadata** ([ListMeta](#))

Standard list metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds>

- **items** ([][ResourceQuota](#)), required

Items is a list of ResourceQuota objects. More info: <https://kubernetes.io/docs/concepts/policy/resource-quotas/>

Operations

get read the specified ResourceQuota

HTTP Request

GET /api/v1/namespaces/{namespace}/resourcequotas/{name}

Parameters

- **name** (*in path*): string, required
name of the ResourceQuota
- **namespace** (*in path*): string, required
[namespace](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([ResourceQuota](#)): OK

401: Unauthorized

get read status of the specified ResourceQuota

HTTP Request

GET /api/v1/namespaces/{namespace}/resourcequotas/{name}/status

Parameters

- **name** (*in path*): string, required
name of the ResourceQuota
- **namespace** (*in path*): string, required
[namespace](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([ResourceQuota](#)): OK

401: Unauthorized

list list or watch objects of kind ResourceQuota

HTTP Request

GET /api/v1/namespaces/{namespace}/resourcequotas

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([ResourceQuotaList](#)): OK

401: Unauthorized

list list or watch objects of kind ResourceQuota

HTTP Request

GET /api/v1/resourcequotas

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([ResourceQuotaList](#)): OK

401: Unauthorized

create create a ResourceQuota

HTTP Request

POST /api/v1/namespaces/{namespace}/resourcequotas

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [ResourceQuota](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ResourceQuota](#)): OK

201 ([ResourceQuota](#)): Created

202 ([ResourceQuota](#)): Accepted

401: Unauthorized

update replace the specified ResourceQuota

HTTP Request

PUT /api/v1/namespaces/{namespace}/resourcequotas/{name}

Parameters

- **name** (*in path*): string, required

name of the ResourceQuota

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [ResourceQuota](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ResourceQuota](#)): OK

201 ([ResourceQuota](#)): Created

401: Unauthorized

update replace status of the specified ResourceQuota

HTTP Request

PUT /api/v1/namespaces/{namespace}/resourcequotas/{name}/status

Parameters

- **name** (*in path*): string, required

name of the ResourceQuota

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [ResourceQuota](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ResourceQuota](#)): OK

201 ([ResourceQuota](#)): Created

401: Unauthorized

patch partially update the specified ResourceQuota

HTTP Request

PATCH /api/v1/namespaces/{namespace}/resourcequotas/{name}

Parameters

- **name** (*in path*): string, required

name of the ResourceQuota

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ResourceQuota](#)): OK

201 ([ResourceQuota](#)): Created

401: Unauthorized

patch partially update status of the specified ResourceQuota

HTTP Request

PATCH /api/v1/namespaces/{namespace}/resourcequotas/{name}/status

Parameters

- **name** (*in path*): string, required
name of the ResourceQuota
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [Patch](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **force** (*in query*): boolean
[force](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([ResourceQuota](#)): OK

201 ([ResourceQuota](#)): Created

401: Unauthorized

delete delete a ResourceQuota

HTTP Request

DELETE /api/v1/namespaces/{namespace}/resourcequotas/{name}

Parameters

- **name** (*in path*): string, required
name of the ResourceQuota
- **namespace** (*in path*): string, required
[namespace](#)

- **body**: [DeleteOptions](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)

Response

200 ([ResourceQuota](#)): OK

202 ([ResourceQuota](#)): Accepted

401: Unauthorized

deletecollection delete collection of ResourceQuota

HTTP Request

DELETE /api/v1/namespaces/{namespace}/resourcequotas

Parameters

- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [DeleteOptions](#)
- **continue** (*in query*): string
[continue](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)

[propagationPolicy](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.6.3 - NetworkPolicy

NetworkPolicy describes what network traffic is allowed for a set of Pods.

```
apiVersion: networking.k8s.io/v1
import "k8s.io/api/networking/v1"
```

NetworkPolicy

NetworkPolicy describes what network traffic is allowed for a set of Pods

- **apiVersion**: networking.k8s.io/v1
- **kind**: NetworkPolicy
- **metadata** ([ObjectMeta](#))

Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **spec** ([NetworkPolicySpec](#))

spec represents the specification of the desired behavior for this NetworkPolicy.

NetworkPolicySpec

NetworkPolicySpec provides the specification of a NetworkPolicy

- **podSelector** ([LabelSelector](#)), required

podSelector selects the pods to which this NetworkPolicy object applies. The array of ingress rules is applied to any pods selected by this field. Multiple network policies can select the same set of pods. In this case, the ingress rules for each are combined additively. This field is NOT optional and follows standard label selector semantics. An empty podSelector matches all pods in this namespace.

- **policyTypes** ([]string)

policyTypes is a list of rule types that the NetworkPolicy relates to. Valid options are ["Ingress"], ["Egress"], or ["Ingress", "Egress"]. If this field is not specified, it will default based on the existence of ingress or egress rules; policies that contain an egress section are assumed to affect egress, and all policies (whether or not they contain an ingress section) are assumed to affect ingress. If you want to write an egress-only policy, you must explicitly specify policyTypes ["Egress"]. Likewise, if you want to write a policy that specifies that no egress is allowed, you must specify a policyTypes value that include "Egress" (since such a policy would not include an egress section and would otherwise default to just ["Ingress"]). This field is beta-level in 1.8

- **ingress** ([]NetworkPolicyIngressRule)

ingress is a list of ingress rules to be applied to the selected pods. Traffic is allowed to a pod if there are no NetworkPolicies selecting the pod (and cluster policy otherwise allows the traffic), OR if the

traffic source is the pod's local node, OR if the traffic matches at least one ingress rule across all of the NetworkPolicy objects whose podSelector matches the pod. If this field is empty then this NetworkPolicy does not allow any traffic (and serves solely to ensure that the pods it selects are isolated by default)

NetworkPolicyIngressRule describes a particular set of traffic that is allowed to the pods matched by a NetworkPolicySpec's podSelector. The traffic must match both ports and from.

- **ingress.from** ([]NetworkPolicyPeer)

from is a list of sources which should be able to access the pods selected for this rule. Items in this list are combined using a logical OR operation. If this field is empty or missing, this rule matches all sources (traffic not restricted by source). If this field is present and contains at least one item, this rule allows traffic only if the traffic matches at least one item in the from list.

NetworkPolicyPeer describes a peer to allow traffic to/from. Only certain combinations of fields are allowed

- **ingress.from.ipBlock** (IPBlock)

ipBlock defines policy on a particular IPBlock. If this field is set then neither of the other fields can be.

IPBlock describes a particular CIDR (Ex. "192.168.1.0/24", "2001:db8::/64") that is allowed to the pods matched by a NetworkPolicySpec's podSelector. The except entry describes CIDRs that should not be included within this rule.

- **ingress.from.ipBlock.cidr** (string), required

cidr is a string representing the IPBlock Valid examples are "192.168.1.0/24" or "2001:db8::/64"

- **ingress.from.ipBlock.exception** ([]string)

exception is a slice of CIDRs that should not be included within an IPBlock Valid examples are "192.168.1.0/24" or "2001:db8::/64" Except values will be rejected if they are outside the cidr range

- **ingress.from.namespaceSelector** ([LabelSelector](#))

namespaceSelector selects namespaces using cluster-scoped labels. This field follows standard label selector semantics; if present but empty, it selects all namespaces.

If podSelector is also set, then the NetworkPolicyPeer as a whole selects the pods matching podSelector in the namespaces selected by namespaceSelector. Otherwise it selects all pods in the namespaces selected by namespaceSelector.

- **ingress.from.podSelector** ([LabelSelector](#))

podSelector is a label selector which selects pods. This field follows standard label selector semantics; if present but empty, it selects all pods.

If namespaceSelector is also set, then the NetworkPolicyPeer as a whole selects the pods matching

podSelector in the Namespaces selected by NamespaceSelector. Otherwise it selects the pods matching podSelector in the policy's own namespace.

- **ingress.ports** ([]NetworkPolicyPort)

ports is a list of ports which should be made accessible on the pods selected for this rule. Each item in this list is combined using a logical OR. If this field is empty or missing, this rule matches all ports (traffic not restricted by port). If this field is present and contains at least one item, then this rule allows traffic only if the traffic matches at least one port in the list.

NetworkPolicyPort describes a port to allow traffic on

- **ingress.ports.port** (IntOrString)

port represents the port on the given protocol. This can either be a numerical or named port on a pod. If this field is not provided, this matches all port names and numbers. If present, only traffic on the specified protocol AND port will be matched.

IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

- **ingress.ports.endPort** (int32)

endPort indicates that the range of ports from port to endPort if set, inclusive, should be allowed by the policy. This field cannot be defined if the port field is not defined or if the port field is defined as a named (string) port. The endPort must be equal or greater than port.

- **ingress.ports.protocol** (string)

protocol represents the protocol (TCP, UDP, or SCTP) which traffic must match. If not specified, this field defaults to TCP.

- **egress** ([]NetworkPolicyEgressRule)

egress is a list of egress rules to be applied to the selected pods. Outgoing traffic is allowed if there are no NetworkPolicies selecting the pod (and cluster policy otherwise allows the traffic), OR if the traffic matches at least one egress rule across all of the NetworkPolicy objects whose podSelector matches the pod. If this field is empty then this NetworkPolicy limits all outgoing traffic (and serves solely to ensure that the pods it selects are isolated by default). This field is beta-level in 1.8

NetworkPolicyEgressRule describes a particular set of traffic that is allowed out of pods matched by a NetworkPolicySpec's podSelector. The traffic must match both ports and to. This type is beta-level in 1.8

- **egress.to** ([]NetworkPolicyPeer)

to is a list of destinations for outgoing traffic of pods selected for this rule. Items in this list are combined using a logical OR operation. If this field is empty or missing, this rule matches all destinations (traffic not restricted by destination). If this field is present and contains at least one item, this rule allows traffic only if the traffic matches at least one item in the to list.

NetworkPolicyPeer describes a peer to allow traffic to/from. Only certain combinations of fields are allowed

- **egress.to.ipBlock** (IPBlock)

ipBlock defines policy on a particular IPBlock. If this field is set then neither of the other fields can be.

IPBlock describes a particular CIDR (Ex. "192.168.1.0/24", "2001:db8::/64") that is allowed to the pods matched by a NetworkPolicySpec's podSelector. The except entry describes CIDRs that should not be included within this rule.

- **egress.to.ipBlock.cidr** (string), required

cidr is a string representing the IPBlock Valid examples are "192.168.1.0/24" or "2001:db8::/64"

- **egress.to.ipBlock.exception** ([]string)

except is a slice of CIDRs that should not be included within an IPBlock Valid examples are "192.168.1.0/24" or "2001:db8::/64" Except values will be rejected if they are outside the cidr range

- **egress.to.namespaceSelector** ([LabelSelector](#))

namespaceSelector selects namespaces using cluster-scoped labels. This field follows standard label selector semantics; if present but empty, it selects all namespaces.

If podSelector is also set, then the NetworkPolicyPeer as a whole selects the pods matching podSelector in the namespaces selected by namespaceSelector. Otherwise it selects all pods in the namespaces selected by namespaceSelector.

- **egress.to.podSelector** ([LabelSelector](#))

podSelector is a label selector which selects pods. This field follows standard label selector semantics; if present but empty, it selects all pods.

If namespaceSelector is also set, then the NetworkPolicyPeer as a whole selects the pods matching podSelector in the Namespaces selected by NamespaceSelector. Otherwise it selects the pods matching podSelector in the policy's own namespace.

- **egress.ports** ([]NetworkPolicyPort)

ports is a list of destination ports for outgoing traffic. Each item in this list is combined using a logical OR. If this field is empty or missing, this rule matches all ports (traffic not restricted by port). If this field is present and contains at least one item, then this rule allows traffic only if the traffic matches at least one port in the list.

NetworkPolicyPort describes a port to allow traffic on

- **egress.ports.port** (IntOrString)

port represents the port on the given protocol. This can either be a numerical or named port on a pod. If this field is not provided, this matches all port names and numbers. If present, only traffic on the specified protocol

AND port will be matched.

IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

- **egress.ports.endPort** (int32)

endPort indicates that the range of ports from port to endPort if set, inclusive, should be allowed by the policy. This field cannot be defined if the port field is not defined or if the port field is defined as a named (string) port. The endPort must be equal or greater than port.

- **egress.ports.protocol** (string)

protocol represents the protocol (TCP, UDP, or SCTP) which traffic must match. If not specified, this field defaults to TCP.

NetworkPolicyList

NetworkPolicyList is a list of NetworkPolicy objects.

- **apiVersion**: networking.k8s.io/v1

- **kind**: NetworkPolicyList

- **metadata** ([ListMeta](#))

Standard list metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **items** ([][NetworkPolicy](#)), required

items is a list of schema objects.

Operations

get read the specified NetworkPolicy

HTTP Request

```
GET /apis/networking.k8s.io/v1/namespaces/{namespace}/  
networkpolicies/{name}
```

Parameters

- **name** (*in path*): string, required

name of the NetworkPolicy

- **namespace** (*in path*): string, required

[namespace](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([NetworkPolicy](#)): OK

401: Unauthorized

list list or watch objects of kind NetworkPolicy

HTTP Request

GET /apis/networking.k8s.io/v1/namespaces/{namespace}/
networkpolicies

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([NetworkPolicyList](#)): OK

401: Unauthorized

list list or watch objects of kind NetworkPolicy

HTTP Request

GET /apis/networking.k8s.io/v1/networkpolicies

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([NetworkPolicyList](#)): OK

401: Unauthorized

create create a NetworkPolicy

HTTP Request

POST /apis/networking.k8s.io/v1/namespaces/{namespace}/networkpolicies

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [NetworkPolicy](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([NetworkPolicy](#)): OK

201 ([NetworkPolicy](#)): Created

202 ([NetworkPolicy](#)): Accepted

401: Unauthorized

update replace the specified NetworkPolicy

HTTP Request

PUT /apis/networking.k8s.io/v1/namespaces/{namespace}/
networkpolicies/{name}

Parameters

- **name** (*in path*): string, required

name of the NetworkPolicy

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [NetworkPolicy](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([NetworkPolicy](#)): OK

201 ([NetworkPolicy](#)): Created

401: Unauthorized

patch partially update the specified NetworkPolicy

HTTP Request

PATCH /apis/networking.k8s.io/v1/namespaces/{namespace}/networkpolicies/{name}

Parameters

- **name** (*in path*): string, required
name of the NetworkPolicy
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [Patch](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **force** (*in query*): boolean
[force](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([NetworkPolicy](#)): OK

201 ([NetworkPolicy](#)): Created

401: Unauthorized

delete delete a NetworkPolicy

HTTP Request

DELETE /apis/networking.k8s.io/v1/namespaces/{namespace}/networkpolicies/{name}

Parameters

- **name** (*in path*): string, required
name of the NetworkPolicy

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [DeleteOptions](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

Response

200 ([Status](#)): OK

202 ([Status](#)): Accepted

401: Unauthorized

deletecollection delete collection of NetworkPolicy

HTTP Request

DELETE /apis/networking.k8s.io/v1/namespaces/{namespace}/
networkpolicies

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [DeleteOptions](#)

- **continue** (*in query*): string

[continue](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.6.4 - PodDisruptionBudget

PodDisruptionBudget is an object to define the max disruption that can be caused to a collection of pods.

```
apiVersion: policy/v1

import "k8s.io/api/policy/v1"
```

PodDisruptionBudget

PodDisruptionBudget is an object to define the max disruption that can be caused to a collection of pods

- **apiVersion**: policy/v1
- **kind**: PodDisruptionBudget
- **metadata** ([ObjectMeta](#))

Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **spec** ([PodDisruptionBudgetSpec](#))

Specification of the desired behavior of the PodDisruptionBudget.

- **status** ([PodDisruptionBudgetStatus](#))

Most recently observed status of the PodDisruptionBudget.

PodDisruptionBudgetSpec

PodDisruptionBudgetSpec is a description of a PodDisruptionBudget.

- **maxUnavailable** ([IntOrString](#))

An eviction is allowed if at most "maxUnavailable" pods selected by "selector" are unavailable after the eviction, i.e. even in absence of the evicted pod. For example, one can prevent all voluntary evictions by specifying 0. This is a mutually exclusive setting with "minAvailable".

IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

- **minAvailable** ([IntOrString](#))

An eviction is allowed if at least "minAvailable" pods selected by "selector" will still be available after the eviction, i.e. even in the absence of the evicted pod. So for example you can prevent all voluntary evictions by specifying "100%".

IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

- **selector** ([LabelSelector](#))

Label query over pods whose evictions are managed by the disruption budget. A null selector will match no pods, while an empty ({}) selector will select all pods within the namespace.

- **unhealthyPodEvictionPolicy** (string)

UnhealthyPodEvictionPolicy defines the criteria for when unhealthy pods should be considered for eviction. Current implementation considers healthy pods, as pods that have status.conditions item with type="Ready",status="True".

Valid policies are IfHealthyBudget and AlwaysAllow. If no policy is specified, the default behavior will be used, which corresponds to the IfHealthyBudget policy.

IfHealthyBudget policy means that running pods (status.phase="Running"), but not yet healthy can be evicted only if the guarded application is not disrupted (status.currentHealthy is at least equal to status.desiredHealthy). Healthy pods will be subject to the PDB for eviction.

AlwaysAllow policy means that all running pods (status.phase="Running"), but not yet healthy are considered disrupted and can be evicted regardless of whether the criteria in a PDB is met. This means perspective running pods of a disrupted application might not get a chance to become healthy. Healthy pods will be subject to the PDB for eviction.

Additional policies may be added in the future. Clients making eviction decisions should disallow eviction of unhealthy pods if they encounter an unrecognized policy in this field.

This field is beta-level. The eviction API uses this field when the feature gate PDBUnhealthyPodEvictionPolicy is enabled (enabled by default).

PodDisruptionBudgetStatus

PodDisruptionBudgetStatus represents information about the status of a PodDisruptionBudget. Status may trail the actual state of a system.

- **currentHealthy** (int32), required

current number of healthy pods

- **desiredHealthy** (int32), required

minimum desired number of healthy pods

- **disruptionsAllowed** (int32), required

Number of pod disruptions that are currently allowed.

- **expectedPods** (int32), required

total number of pods counted by this disruption budget

- **conditions** ([]Condition)

Patch strategy: merge on key type

Map: unique values on key type will be kept during a merge

Conditions contain conditions for PDB. The disruption controller sets the DisruptionAllowed condition. The following are known values for the reason field (additional reasons could be added in the

future): - SyncFailed: The controller encountered an error and wasn't able to compute the number of allowed disruptions.

Therefore no disruptions are allowed and the status of the condition will be False.

- InsufficientPods: The number of pods are either at or below the number required by the PodDisruptionBudget. No disruptions are allowed and the status of the condition will be False.
- SufficientPods: There are more pods than required by the PodDisruptionBudget. The condition will be True, and the number of allowed disruptions are provided by the disruptionsAllowed property.

Condition contains details for one aspect of the current state of this API Resource.

- **conditions.lastTransitionTime** (Time), required

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.

Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

- **conditions.message** (string), required

message is a human readable message indicating details about the transition. This may be an empty string.

- **conditions.reason** (string), required

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.

- **conditions.status** (string), required

status of the condition, one of True, False, Unknown.

- **conditions.type** (string), required

type of condition in CamelCase or in foo.example.com/CamelCase.

- **conditions.observedGeneration** (int64)

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.

- **disruptedPods** (map[string]Time)

DisruptedPods contains information about pods whose eviction was processed by the API server eviction subresource handler but has not yet been observed by the PodDisruptionBudget controller. A pod will be in this map from the time when the API server processed the eviction request to the time when the pod is seen by PDB controller as having been marked for deletion (or after a timeout). The key in the map is the name of the pod and the value is

the time when the API server processed the eviction request. If the deletion didn't occur and a pod is still there it will be removed from the list automatically by PodDisruptionBudget controller after some time. If everything goes smooth this map should be empty for the most of the time. Large number of entries in the map may indicate problems with pod deletions.

Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

- **observedGeneration** (int64)

Most recent generation observed when updating this PDB status. DisruptionsAllowed and other status information is valid only if observedGeneration equals to PDB's object generation.

PodDisruptionBudgetList

PodDisruptionBudgetList is a collection of PodDisruptionBudgets.

- **apiVersion**: policy/v1
- **kind**: PodDisruptionBudgetList
- **metadata** ([ListMeta](#))

Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **items** ([\[\]PodDisruptionBudget](#)), required

Items is a list of PodDisruptionBudgets

Operations

get read the specified PodDisruptionBudget

HTTP Request

GET /apis/policy/v1/namespaces/{namespace}/poddisruptionbudgets/{name}

Parameters

- **name** (*in path*): string, required
name of the PodDisruptionBudget
- **namespace** (*in path*): string, required
[namespace](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([PodDisruptionBudget](#)): OK

401: Unauthorized

get read status of the specified PodDisruptionBudget

HTTP Request

GET /apis/policy/v1/namespaces/{namespace}/poddisruptionbudgets/{name}/status

Parameters

- **name** (*in path*): string, required
name of the PodDisruptionBudget
- **namespace** (*in path*): string, required
[namespace](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([PodDisruptionBudget](#)): OK

401: Unauthorized

list list or watch objects of kind PodDisruptionBudget

HTTP Request

GET /apis/policy/v1/namespaces/{namespace}/poddisruptionbudgets

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([PodDisruptionBudgetList](#)): OK

401: Unauthorized

list list or watch objects of kind PodDisruptionBudget

HTTP Request

GET /apis/policy/v1/poddisruptionbudgets

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([PodDisruptionBudgetList](#)): OK

401: Unauthorized

create create a PodDisruptionBudget

HTTP Request

POST /apis/policy/v1/namespaces/{namespace}/poddisruptionbudgets

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [PodDisruptionBudget](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([PodDisruptionBudget](#)): OK

201 ([PodDisruptionBudget](#)): Created

202 ([PodDisruptionBudget](#)): Accepted

401: Unauthorized

update replace the specified PodDisruptionBudget

HTTP Request

PUT /apis/policy/v1/namespaces/{namespace}/poddisruptionbudgets/{name}

Parameters

- **name** (*in path*): string, required

name of the PodDisruptionBudget

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [PodDisruptionBudget](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([PodDisruptionBudget](#)): OK

201 ([PodDisruptionBudget](#)): Created

401: Unauthorized

update replace status of the specified PodDisruptionBudget

HTTP Request

PUT /apis/policy/v1/namespaces/{namespace}/poddisruptionbudgets/{name}/status

Parameters

- **name** (*in path*): string, required

name of the PodDisruptionBudget

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [PodDisruptionBudget](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([PodDisruptionBudget](#)): OK

201 ([PodDisruptionBudget](#)): Created

401: Unauthorized

patch partially update the specified PodDisruptionBudget

HTTP Request

PATCH /apis/policy/v1/namespaces/{namespace}/poddisruptionbudgets/{name}

Parameters

- **name** (*in path*): string, required
name of the PodDisruptionBudget
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [Patch](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **force** (*in query*): boolean
[force](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([PodDisruptionBudget](#)): OK

201 ([PodDisruptionBudget](#)): Created

401: Unauthorized

patch partially update status of the specified PodDisruptionBudget

HTTP Request

PATCH /apis/policy/v1/namespaces/{namespace}/poddisruptionbudgets/{name}/status

Parameters

- **name** (*in path*): string, required
name of the PodDisruptionBudget
- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [Patch](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **force** (*in query*): boolean
[force](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([PodDisruptionBudget](#)): OK

201 ([PodDisruptionBudget](#)): Created

401: Unauthorized

delete delete a PodDisruptionBudget

HTTP Request

DELETE /apis/policy/v1/namespaces/{namespace}/
poddisruptionbudgets/{name}

Parameters

- **name** (*in path*): string, required
name of the PodDisruptionBudget
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [DeleteOptions](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)

Response

200 ([Status](#)): OK

202 ([Status](#)): Accepted

401: Unauthorized

deletecollection delete collection of PodDisruptionBudget

HTTP Request

DELETE /apis/policy/v1/namespaces/{namespace}/poddisruptionbudgets

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [DeleteOptions](#)

- **continue** (*in query*): string

[continue](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.6.5 - IPAddress v1alpha1

IPAddress represents a single IP of a single IP Family.

```
apiVersion: networking.k8s.io/v1alpha1

import "k8s.io/api/networking/v1alpha1"
```

IPAddress

IPAddress represents a single IP of a single IP Family. The object is designed to be used by APIs that operate on IP addresses. The object is used by the Service core API for allocation of IP addresses. An IP address can be represented in different formats, to guarantee the uniqueness of the IP, the name of the object is the IP address in canonical format, four decimal digits separated by dots suppressing leading zeros for IPv4 and the representation defined by RFC 5952 for IPv6. Valid: 192.168.1.5 or 2001:db8::1 or 2001:db8:aaaa:bbbb:cccc:dddd:eeee:1 Invalid: 10.01.2.3 or 2001:db8:0:0:0::1

- **apiVersion**: networking.k8s.io/v1alpha1

- **kind**: IPAddress

- **metadata** ([ObjectMeta](#))

Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **spec** ([IPAddressSpec](#))

spec is the desired state of the IPAddress. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status>

IPAddressSpec

IPAddressSpec describe the attributes in an IP Address.

- **parentRef** ([ParentReference](#))

ParentRef references the resource that an IPAddress is attached to. An IPAddress must reference a parent object.

ParentReference describes a reference to a parent object.

- **parentRef.group** (string)

Group is the group of the object being referenced.

- **parentRef.name** (string)

Name is the name of the object being referenced.

- **parentRef.namespace** (string)

Namespace is the namespace of the object being referenced.

- **parentRef.resource** (string)

Resource is the resource of the object being referenced.

- **parentRef.uid** (string)

UID is the uid of the object being referenced.

IPAddressList

IPAddressList contains a list of IPAddress.

- **apiVersion**: networking.k8s.io/v1alpha1
- **kind**: IPAddressList
- **metadata** ([ListMeta](#))

Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **items** ([\[\]IPAddress](#)), required

items is the list of IPAddresses.

Operations

get read the specified IPAddress

HTTP Request

GET /apis/networking.k8s.io/v1alpha1/ipaddresses/{name}

Parameters

- **name** (*in path*): string, required

name of the IPAddress

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([IPAddress](#)): OK

401: Unauthorized

list list or watch objects of kind IPAddress

HTTP Request

GET /apis/networking.k8s.io/v1alpha1/ipaddresses

Parameters

- **allowWatchBookmarks** (*in query*): boolean
[allowWatchBookmarks](#)
- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([IPAddressList](#)): OK

401: Unauthorized

create create an IPAddress

HTTP Request

POST /apis/networking.k8s.io/v1alpha1/ipaddresses

Parameters

- **body**: [IPAddress](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([IPAddress](#)): OK

201 ([IPAddress](#)): Created

202 ([IPAddress](#)): Accepted

401: Unauthorized

update replace the specified IPAddress

HTTP Request

PUT /apis/networking.k8s.io/v1alpha1/ipaddresses/{name}

Parameters

- **name** (*in path*): string, required

name of the IPAddress

- **body**: [IPAddress](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([IPAddress](#)): OK

201 ([IPAddress](#)): Created

401: Unauthorized

patch partially update the specified IPAddress

HTTP Request

PATCH /apis/networking.k8s.io/v1alpha1/ipaddresses/{name}

Parameters

- **name** (*in path*): string, required

name of the IPAddress

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([IPAddress](#)): OK

201 ([IPAddress](#)): Created

401: Unauthorized

delete delete an IPAddress

HTTP Request

DELETE /apis/networking.k8s.io/v1alpha1/ipaddresses/{name}

Parameters

- **name** (*in path*): string, required

name of the IPAddress

- **body**: [DeleteOptions](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

Response

200 ([Status](#)): OK

202 ([Status](#)): Accepted

401: Unauthorized

deletecollection delete collection of IPAddress

HTTP Request

DELETE /apis/networking.k8s.io/v1alpha1/ipaddresses

Parameters

- **body**: [DeleteOptions](#)
- **continue** (*in query*): string
[continue](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.7 - Extend Resources

5.7.1 - CustomResourceDefinition

CustomResourceDefinition represents a resource that should be exposed on the API server.

```
apiVersion: apiextensions.k8s.io/v1
import "k8s.io/apiextensions-apiserver/pkg/apis/apiextensions/v1"
```

CustomResourceDefinition

CustomResourceDefinition represents a resource that should be exposed on the API server. Its name MUST be in the format `<.spec.name>.<.spec.group>`.

- **apiVersion**: apiextensions.k8s.io/v1

- **kind**: CustomResourceDefinition

- **metadata** ([ObjectMeta](#))

Standard object's metadata More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **spec** ([CustomResourceDefinitionSpec](#)), required

spec describes how the user wants the resources to appear

- **status** ([CustomResourceDefinitionStatus](#))

status indicates the actual state of the CustomResourceDefinition

CustomResourceDefinitionSpec

CustomResourceDefinitionSpec describes how a user wants their resource to appear

- **group** (string), required

group is the API group of the defined custom resource. The custom resources are served under `/apis/\/...`. Must match the name of the CustomResourceDefinition (in the form `\<names.plural\>.\`).

- **names** ([CustomResourceDefinitionNames](#)), required

names specify the resource and kind names for the custom resource.

CustomResourceDefinitionNames indicates the names to serve this CustomResourceDefinition

- **names.kind** (string), required

kind is the serialized kind of the resource. It is normally CamelCase and singular. Custom resource instances will use this value as the `kind` attribute in API calls.

- **names.plural** (string), required

plural is the plural name of the resource to serve. The custom resources are served under `/apis/\<group>/\<version>/.../\<plural>`. Must match the name of the CustomResourceDefinition (in the form `\<names.plural>.\<group>`). Must be all lowercase.

- **names.categories** ([]string)

categories is a list of grouped resources this custom resource belongs to (e.g. 'all'). This is published in API discovery documents, and used by clients to support invocations like `kubectl get all`.

- **names.listKind** (string)

listKind is the serialized kind of the list for this resource. Defaults to " kind List".

- **names.shortNames** ([]string)

shortNames are short names for the resource, exposed in API discovery documents, and used by clients to support invocations like `kubectl get \<shortname>`. It must be all lowercase.

- **names.singular** (string)

singular is the singular name of the resource. It must be all lowercase. Defaults to lowercased `kind`.

- **scope** (string), required

scope indicates whether the defined custom resource is cluster- or namespace-scoped. Allowed values are `Cluster` and `Namespaced`.

- **versions** ([]CustomResourceDefinitionVersion), required

versions is the list of all API versions of the defined custom resource. Version names are used to compute the order in which served versions are listed in API discovery. If the version string is "kube-like", it will sort above non "kube-like" version strings, which are ordered lexicographically. "Kube-like" versions start with a "v", then are followed by a number (the major version), then optionally the string "alpha" or "beta" and another number (the minor version). These are sorted first by GA > beta > alpha (where GA is a version with no suffix such as beta or alpha), and then by comparing major version, then minor version. An example sorted list of versions: v10, v2, v1, v11beta2, v10beta3, v3beta1, v12alpha1, v11alpha2, foo1, foo10.

CustomResourceDefinitionVersion describes a version for CRD.

- **versions.name** (string), required

name is the version name, e.g. "v1", "v2beta1", etc. The custom resources are served under this version at `/apis/\<group>/\<version>/...` if `served` is true.

- **versions.served** (boolean), required

served is a flag enabling/disabling this version from being served via REST APIs

- **versions.storage** (boolean), required

storage indicates this version should be used when persisting

custom resources to storage. There must be exactly one version with storage=true.

- **versions.additionalPrinterColumns**

([]CustomResourceColumnDefinition)

additionalPrinterColumns specifies additional columns returned in Table output. See <https://kubernetes.io/docs/reference/using-api/api-concepts/#receiving-resources-as-tables> for details. If no columns are specified, a single column displaying the age of the custom resource is used.

CustomResourceColumnDefinition specifies a column for server side printing.

- **versions.additionalPrinterColumns.jsonPath** (string), required

jsonPath is a simple JSON path (i.e. with array notation) which is evaluated against each custom resource to produce the value for this column.

- **versions.additionalPrinterColumns.name** (string), required

name is a human readable name for the column.

- **versions.additionalPrinterColumns.type** (string), required

type is an OpenAPI type definition for this column. See <https://github.com/OAI/OpenAPI-Specification/blob/master/versions/2.0.md#data-types> for details.

- **versions.additionalPrinterColumns.description** (string)

description is a human readable description of this column.

- **versions.additionalPrinterColumns.format** (string)

format is an optional OpenAPI type definition for this column. The 'name' format is applied to the primary identifier column to assist in clients identifying column is the resource name. See <https://github.com/OAI/OpenAPI-Specification/blob/master/versions/2.0.md#data-types> for details.

- **versions.additionalPrinterColumns.priority** (int32)

priority is an integer defining the relative importance of this column compared to others. Lower numbers are considered higher priority. Columns that may be omitted in limited space scenarios should be given a priority greater than 0.

- **versions.deprecated** (boolean)

deprecated indicates this version of the custom resource API is deprecated. When set to true, API requests to this version receive a warning header in the server response. Defaults to false.

- **versions.deprecationWarning** (string)

deprecationWarning overrides the default warning returned to API clients. May only be set when `deprecated` is true. The

default warning indicates this version is deprecated and recommends use of the newest served version of equal or greater stability, if one exists.

- **versions.schema** (CustomResourceValidation)

schema describes the schema used for validation, pruning, and defaulting of this version of the custom resource.

CustomResourceValidation is a list of validation methods for CustomResources.

- **versions.schema.openAPIV3Schema**

(JSONSchemaProps)

openAPIV3Schema is the OpenAPI v3 schema to use for validation and pruning.

- **versions.subresources** (CustomResourceSubresources)

subresources specify what subresources this version of the defined custom resource have.

CustomResourceSubresources defines the status and scale subresources for CustomResources.

- **versions.subresources.scale**

(CustomResourceSubresourceScale)

scale indicates the custom resource should serve a `/scale` subresource that returns an `autoscaling/v1` Scale object.

CustomResourceSubresourceScale defines how to serve the scale subresource for CustomResources.

- **versions.subresources.scale.specReplicasPath**

(string), required

specReplicasPath defines the JSON path inside of a custom resource that corresponds to Scale `spec.replicas`. Only JSON paths without the array notation are allowed. Must be a JSON Path under `.spec`. If there is no value under the given path in the custom resource, the `/scale` subresource will return an error on GET.

- **versions.subresources.scale.statusReplicasPath**

(string), required

statusReplicasPath defines the JSON path inside of a custom resource that corresponds to Scale `status.replicas`. Only JSON paths without the array notation are allowed. Must be a JSON Path under `.status`. If there is no value under the given path in the custom resource, the `status.replicas` value in the `/scale` subresource will default to 0.

- **versions.subresources.scale.labelSelectorPath**

(string)

labelSelectorPath defines the JSON path inside of a custom resource that corresponds to Scale `status.selector`. Only JSON paths without the array notation are allowed. Must be a JSON Path under `.status` or `.spec`. Must be set to work with HorizontalPodAutoscaler. The field pointed by this

JSON path must be a string field (not a complex selector struct) which contains a serialized label selector in string form. More info: <https://kubernetes.io/docs/tasks/access-kubernetes-api/custom-resources/custom-resource-definitions#scale-subresource> If there is no value under the given path in the custom resource, the `status.selector` value in the `/scale` subresource will default to the empty string.

- **versions.subresources.status**

(CustomResourceSubresourceStatus)

status indicates the custom resource should serve a `/status` subresource. When enabled: 1. requests to the custom resource primary endpoint ignore changes to the `status` stanza of the object. 2. requests to the custom resource `/status` subresource ignore changes to anything other than the `status` stanza of the object.

*CustomResourceSubresourceStatus defines how to serve the status subresource for CustomResources. Status is represented by the `.status` JSON path inside of a CustomResource. When set, * exposes a `/status` subresource for the custom resource * `PUT` requests to the `/status` subresource take a custom resource object, and ignore changes to anything except the `status` stanza * `PUT/POST/PATCH` requests to the custom resource ignore changes to the `status` stanza*

- **conversion** (CustomResourceConversion)

conversion defines conversion settings for the CRD.

CustomResourceConversion describes how to convert different versions of a CR.

- **conversion.strategy** (string), required

strategy specifies how custom resources are converted between versions. Allowed values are: - "None" : The converter only change the `apiVersion` and would not touch any other field in the custom resource. - "Webhook" : API Server will call to an external webhook to do the conversion. Additional information is needed for this option. This requires `spec.preserveUnknownFields` to be false, and `spec.conversion.webhook` to be set.

- **conversion.webhook** (WebhookConversion)

webhook describes how to call the conversion webhook. Required when `strategy` is set to "Webhook".

WebhookConversion describes how to call a conversion webhook

- **conversion.webhook.conversionReviewVersions**

([]string), required

`conversionReviewVersions` is an ordered list of preferred `ConversionReview` versions the Webhook expects. The API server will use the first version in the list which it supports. If none of the versions specified in this list are supported by API server, conversion will fail for the custom resource. If a persisted Webhook configuration specifies allowed versions and does not include any versions known to the API Server, calls to the webhook

will fail.

- **conversion.webhook.clientConfig**

(WebhookClientConfig)

clientConfig is the instructions for how to call the webhook if strategy is `Webhook`.

WebhookClientConfig contains the information to make a TLS connection with the webhook.

- **conversion.webhook.clientConfig.caBundle**

([]byte)

caBundle is a PEM encoded CA bundle which will be used to validate the webhook's server certificate. If unspecified, system trust roots on the apiserver are used.

- **conversion.webhook.clientConfig.service**

(ServiceReference)

service is a reference to the service for this webhook. Either service or url must be specified.

If the webhook is running within the cluster, then you should use `service`.

ServiceReference holds a reference to Service.legacy.k8s.io

- **conversion.webhook.clientConfig.service.name**

(string), required

name is the name of the service. Required

- **conversion.webhook.clientConfig.service.namespace**

(string), required

namespace is the namespace of the service.

Required

- **conversion.webhook.clientConfig.service.path**

(string)

path is an optional URL path at which the webhook will be contacted.

- **conversion.webhook.clientConfig.service.port**

(int32)

port is an optional service port at which the webhook will be contacted. `port` should be a valid port number (1-65535, inclusive).

Defaults to 443 for backward compatibility.

- **conversion.webhook.clientConfig.url** (string)

url gives the location of the webhook, in standard URL form (`scheme://host:port/path`). Exactly one of `url` or `service` must be specified.

The `host` should not refer to a service running in the cluster; use the `service` field instead. The host might be resolved via external DNS in some apiservers (e.g., `kube-apiserver` cannot resolve in-cluster DNS as that would be a layering violation). `host` may also be an IP address.

Please note that using `localhost` or `127.0.0.1` as a host is risky unless you take great care to run this webhook on all hosts which run an apiserver which might need to make calls to this webhook. Such installs are likely to be non-portable, i.e., not easy to turn up in a new cluster.

The scheme must be "https"; the URL must begin with "https://".

A path is optional, and if present may be any string permissible in a URL. You may use the path to pass an arbitrary string to the webhook, for example, a cluster identifier.

Attempting to use a user or basic auth e.g. "user:password@" is not allowed. Fragments ("#...") and query parameters ("?...") are not allowed, either.

- **preserveUnknownFields** (boolean)

`preserveUnknownFields` indicates that object fields which are not specified in the OpenAPI schema should be preserved when persisting to storage. `apiVersion`, `kind`, `metadata` and `knownFields` inside `metadata` are always preserved. This field is deprecated in favor of setting `x-preserve-unknown-fields` to true in `spec.versions[*].schema.openAPIV3Schema`. See <https://kubernetes.io/docs/tasks/extend-kubernetes/custom-resources/custom-resource-definitions/#field-pruning> for details.

JSONSchemaProps

`JSONSchemaProps` is a JSON-Schema following Specification Draft 4 (<http://json-schema.org/>).

- **\$ref** (string)
- **\$schema** (string)
- **additionalItems** (JSONSchemaPropsOrBool)

JSONSchemaPropsOrBool represents JSONSchemaProps or a boolean value. Defaults to true for the boolean property.

- **additionalProperties** (JSONSchemaPropsOrBool)

JSONSchemaPropsOrBool represents JSONSchemaProps or a boolean value. Defaults to true for the boolean property.

- **allOf** ([][JSONSchemaProps](#))
- **anyOf** ([][JSONSchemaProps](#))
- **default** (JSON)

`default` is a default value for undefined object fields. Defaulting is a beta feature under the `CustomResourceDefaulting` feature gate. Defaulting requires `spec.preserveUnknownFields` to be false.

JSON represents any valid JSON value. These types are supported: bool, int64, float64, string, []interface{}, map[string]interface{} and nil.

- **definitions** (map[string][JSONSchemaProps](#))
- **dependencies** (map[string][JSONSchemaPropsOrStringArray](#))

JSONSchemaPropsOrStringArray represents a *JSONSchemaProps* or a *string* array.

- **description** (string)
- **enum** ([]JSON)

JSON represents any valid *JSON* value. These types are supported: *bool*, *int64*, *float64*, *string*, *[]interface{}*, *map[string]interface{}* and *nil*.

- **example** (JSON)

JSON represents any valid *JSON* value. These types are supported: *bool*, *int64*, *float64*, *string*, *[]interface{}*, *map[string]interface{}* and *nil*.

- **exclusiveMaximum** (boolean)
- **exclusiveMinimum** (boolean)
- **externalDocs** (ExternalDocumentation)

ExternalDocumentation allows referencing an external resource for extended documentation.

- **externalDocs.description** (string)
- **externalDocs.url** (string)

- **format** (string)

format is an OpenAPI v3 format string. Unknown formats are ignored. The following formats are validated:

- **bsonobjectid**: a bson object ID, i.e. a 24 characters hex string -
uri: an URI as parsed by Golang `net/url.ParseRequestURI` -
email: an email address as parsed by Golang `net/mail.ParseAddress` -
hostname: a valid representation for an Internet host name, as defined by RFC 1034, section 3.1 [RFC1034]. -
ipv4: an IPv4 IP as parsed by Golang `net.ParseIP` -
ipv6: an IPv6 IP as parsed by Golang `net.ParseIP` -
cidr: a CIDR as parsed by Golang `net.ParseCIDR` -
mac: a MAC address as parsed by Golang `net.ParseMAC` -
uuid: an UUID that allows uppercase defined by the regex `(?i)^[0-9a-f]{8}-?[0-9a-f]{4}-?[0-9a-f]{4}-?[0-9a-f]{4}-?[0-9a-f]{12}$` -
uuid3: an UUID3 that allows uppercase defined by the regex `(?i)^[0-9a-f]{8}-?[0-9a-f]{4}-?[0-9a-f]{3}-?[0-9a-f]{4}-?[0-9a-f]{12}$` -
uuid4: an UUID4 that allows uppercase defined by the regex `(?i)^[0-9a-f]{8}-?[0-9a-f]{4}-?[0-9a-f]{4}-?[0-9a-f]{4}-?[89ab][0-9a-f]{3}-?[0-9a-f]{12}$` -
uuid5: an UUID5 that allows uppercase defined by the regex `(?i)^[0-9a-f]{8}-?[0-9a-f]{4}-?[50-9a-f]{3}-?[89ab][0-9a-f]{3}-?[0-9a-f]{12}$` -
isbn: an ISBN10 or ISBN13 number string like "0321751043" or "978-0321751041" -
isbn10: an ISBN10 number string like "0321751043" -
isbn13: an ISBN13 number string like "978-0321751041" -
creditcard: a credit card number defined by the regex `^(?:4[0-9]{12}(?:[0-9]{3})?|5[1-5][0-9]{14}|6(?:011|5[0-9][0-9])[0-9]{12}|3[47][0-9]{13}|3(?:0[0-5]|68)[0-9]{11}|(?:2131|1800|35\d{3})\d{11})$` with any non digit characters mixed in -
ssn: a U.S. social security number following the regex `^\d{3}[-]?\d{2}[-]?\d{4}$` -
hexcolor: an hexadecimal color code like "#FFFFFF" following the regex `^#([0-9a-fA-F]{3}|[0-9a-fA-F]{6})$` -
rgbcolor: an RGB color code like `rgb(255,255,255)` -
byte: base64 encoded binary data -
password: any kind of string -
date: a date string like "2006-01-02" as defined by full-date in RFC3339 -
duration: a duration string like "22 ns" as parsed by Golang `time.ParseDuration` or compatible with Scala duration format -

datetime: a date time string like "2014-12-15T19:30:20.000Z" as defined by date-time in RFC3339.

- **id** (string)
- **items** (JSONSchemaPropsOrArray)

JSONSchemaPropsOrArray represents a value that can either be a JSONSchemaProps or an array of JSONSchemaProps. Mainly here for serialization purposes.

- **maxItems** (int64)
- **maxLength** (int64)
- **maxProperties** (int64)
- **maximum** (double)
- **minItems** (int64)
- **minLength** (int64)
- **minProperties** (int64)
- **minimum** (double)
- **multipleOf** (double)
- **not** (JSONSchemaProps)
- **nullable** (boolean)
- **oneOf** ([]JSONSchemaProps)
- **pattern** (string)
- **patternProperties** (map[string]JSONSchemaProps)
- **properties** (map[string]JSONSchemaProps)
- **required** ([]string)
- **title** (string)
- **type** (string)
- **uniqueItems** (boolean)
- **x-kubernetes-embedded-resource** (boolean)

x-kubernetes-embedded-resource defines that the value is an embedded Kubernetes runtime.Object, with TypeMeta and ObjectMeta. The type must be object. It is allowed to further restrict the embedded object. kind, apiVersion and metadata are validated automatically. x-kubernetes-preserve-unknown-fields is allowed to be true, but does not have to be if the object is fully specified (up to kind, apiVersion, metadata).

- **x-kubernetes-int-or-string** (boolean)

x-kubernetes-int-or-string specifies that this value is either an integer or a string. If this is true, an empty type is allowed and type as child of anyOf is permitted if following one of the following patterns:

1. anyOf:
 - type: integer
 - type: string
2. allOf:
 - anyOf:

- type: integer

- type: string

- ... zero or more

- **x-kubernetes-list-map-keys** ([]string)

x-kubernetes-list-map-keys annotates an array with the `x-kubernetes-list-type map` by specifying the keys used as the index of the map.

This tag MUST only be used on lists that have the "x-kubernetes-list-type" extension set to "map". Also, the values specified for this attribute must be a scalar typed field of the child structure (no nesting is supported).

The properties specified must either be required or have a default value, to ensure those properties are present for all list items.

- **x-kubernetes-list-type** (string)

x-kubernetes-list-type annotates an array to further describe its topology. This extension must only be used on lists and may have 3 possible values:

1. `atomic` : the list is treated as a single entity, like a scalar.
Atomic lists will be entirely replaced when updated. This extension may be used on any type of list (struct, scalar, ...).
2. `set` : Sets are lists that must not have multiple items with the same value. Each value must be a scalar, an object with `x-kubernetes-map-type atomic` or an array with `x-kubernetes-list-type atomic`.
3. `map` : These lists are like maps in that their elements have a non-index key used to identify them. Order is preserved upon merge. The map tag must only be used on a list with elements of type object. Defaults to atomic for arrays.

- **x-kubernetes-map-type** (string)

x-kubernetes-map-type annotates an object to further describe its topology. This extension must only be used when type is object and may have 2 possible values:

1. `granular` : These maps are actual maps (key-value pairs) and each fields are independent from each other (they can each be manipulated by separate actors). This is the default behaviour for all maps.
2. `atomic` : the list is treated as a single entity, like a scalar.
Atomic maps will be entirely replaced when updated.

- **x-kubernetes-preserve-unknown-fields** (boolean)

x-kubernetes-preserve-unknown-fields stops the API server decoding step from pruning fields which are not specified in the validation schema. This affects fields recursively, but switches back to normal pruning behaviour if nested properties or `additionalProperties` are specified in the schema. This can either be true or undefined. False is forbidden.

- **x-kubernetes-validations** ([]ValidationRule)

Patch strategy: merge on key rule

Map: unique values on key rule will be kept during a merge

x-kubernetes-validations describes a list of validation rules written in the CEL expression language. This field is an alpha-level. Using this field requires the feature gate

`CustomResourceValidationExpressions` to be enabled.

ValidationRule describes a validation rule written in the CEL expression language.

- **x-kubernetes-validations.rule** (string), required

Rule represents the expression which will be evaluated by CEL.
ref: <https://github.com/google/ce1-spec> The Rule is scoped to the location of the x-kubernetes-validations extension in the schema. The `self` variable in the CEL expression is bound to the scoped value. Example: - Rule scoped to the root of a resource with a status subresource: {"rule": "self.status.actual <= self.spec.maxDesired"}

If the Rule is scoped to an object with properties, the accessible properties of the object are field selectable via `self.field` and field presence can be checked via `has(self.field)`. Null valued fields are treated as absent fields in CEL expressions. If the Rule is scoped to an object with `additionalProperties` (i.e. a map) the value of the map are accessible via `self[mapKey]`, map containment can be checked via `mapKey in self` and all entries of the map are accessible via CEL macros and functions such as `self.all(...)`. If the Rule is scoped to an array, the elements of the array are accessible via `self[i]` and also by macros and functions. If the Rule is scoped to a scalar, `self` is bound to the scalar value. Examples: - Rule scoped to a map of objects: {"rule": "self.components['Widget'].priority < 10"} - Rule scoped to a list of integers: {"rule": "self.values.all(value, value >= 0 && value < 100)"} - Rule scoped to a string value: {"rule": "self.startsWith('kube')"}

The `apiVersion`, `kind`, `metadata.name` and `metadata.generateName` are always accessible from the root of the object and from any x-kubernetes-embedded-resource annotated objects. No other metadata properties are accessible.

Unknown data preserved in custom resources via x-kubernetes-preserve-unknown-fields is not accessible in CEL expressions. This includes: - Unknown field values that are preserved by object schemas with x-kubernetes-preserve-unknown-fields. - Object properties where the property schema is of an "unknown type". An "unknown type" is recursively defined as:

- A schema with no type and x-kubernetes-preserve-unknown-fields set to true
- An array where the items schema is of an "unknown type"
- An object where the `additionalProperties` schema is of an "unknown type"

Only property names of the form `[a-zA-Z_.-/*][a-zA-Z0-9_.-/*]*` are accessible. Accessible property names are escaped according to the following rules when accessed in the expression: - **'escapes to 'underscores'** - **'escapes to 'dot'** - **'escapes to 'dash'** - **'escapes to 'slash'** - **Property names that exactly match a CEL RESERVED keyword escape to '{keyword}__'**. The keywords are: "true", "false", "null", "in", "as", "break", "const", "continue", "else", "for", "function", "if", "import", "let", "loop", "package", "namespace", "return". Examples:

- Rule accessing a property named "namespace": {"rule":

- "self.namespace > 0"}
- Rule accessing a property named "x-prop": {"rule": "self.x_dash_prop > 0"}
- Rule accessing a property named "redact_d": {"rule": "self.redact_underscores_d > 0"}

Equality on arrays with x-kubernetes-list-type of 'set' or 'map' ignores element order, i.e. [1, 2] == [2, 1]. Concatenation on arrays with x-kubernetes-list-type use the semantics of the list type:

- 'set': $x + y$ performs a union where the array positions of all elements in x are preserved and non-intersecting elements in y are appended, retaining their partial order.
 - 'map': $x + y$ performs a merge where the array positions of all keys in x are preserved but the values are overwritten by values in y when the key sets of x and y intersect. Elements in y with non-intersecting keys are appended, retaining their partial order.
- **x-kubernetes-validations.fieldPath** (string)

fieldPath represents the field path returned when the validation fails. It must be a relative JSON path (i.e. with array notation) scoped to the location of this x-kubernetes-validations extension in the schema and refer to an existing field. e.g. when validation checks if a specific attribute `foo` under a map `testMap`, the fieldPath could be set to `.testMap.foo`. If the validation checks two lists must have unique attributes, the fieldPath could be set to either of the list: e.g. `.testList`. It does not support list numeric index. It supports child operation to refer to an existing field currently. Refer to [JSONPath support in Kubernetes](#) for more info. Numeric index of array is not supported. For field name which contains special characters, use `['specialName']` to refer the field name. e.g. for attribute `foo.34$` appears in a list `testList`, the fieldPath could be set to `.testList['foo.34$']`

- **x-kubernetes-validations.message** (string)

Message represents the message displayed when validation fails. The message is required if the Rule contains line breaks. The message must not contain line breaks. If unset, the message is "failed rule: {Rule}". e.g. "must be a URL with the host matching spec.host"

- **x-kubernetes-validations.messageExpression** (string)

MessageExpression declares a CEL expression that evaluates to the validation failure message that is returned when this rule fails. Since messageExpression is used as a failure message, it must evaluate to a string. If both message and messageExpression are present on a rule, then messageExpression will be used if validation fails. If messageExpression results in a runtime error, the runtime error is logged, and the validation failure message is produced as if the messageExpression field were unset. If messageExpression evaluates to an empty string, a string with only spaces, or a string that contains line breaks, then the validation failure message will also be produced as if the messageExpression field were unset, and the fact that messageExpression produced an empty string/string with only spaces/string with line breaks will be logged. messageExpression has access to all the same variables as the

rule; the only difference is the return type. Example: "x must be less than max ("+string(self.max)+")"

- **x-kubernetes-validations.reason** (string)

reason provides a machine-readable validation failure reason that is returned to the caller when a request fails this validation rule. The HTTP status code returned to the caller will match the reason of the reason of the first failed validation rule. The currently supported reasons are: "FieldValueInvalid", "FieldValueForbidden", "FieldValueRequired", "FieldValueDuplicate". If not set, default to use "FieldValueInvalid". All future added reasons must be accepted by clients when reading this value and unknown reasons should be treated as FieldValueInvalid.

CustomResourceDefinitionStatus

CustomResourceDefinitionStatus indicates the state of the CustomResourceDefinition

- **acceptedNames** (CustomResourceDefinitionNames)

acceptedNames are the names that are actually being used to serve discovery. They may be different than the names in spec.

CustomResourceDefinitionNames indicates the names to serve this CustomResourceDefinition

- **acceptedNames.kind** (string), required

kind is the serialized kind of the resource. It is normally CamelCase and singular. Custom resource instances will use this value as the `kind` attribute in API calls.

- **acceptedNames.plural** (string), required

plural is the plural name of the resource to serve. The custom resources are served under `/apis/\<group>/\<version>/.../\<plural>`. Must match the name of the CustomResourceDefinition (in the form `\<names.plural>.\<group>`). Must be all lowercase.

- **acceptedNames.categories** ([]string)

categories is a list of grouped resources this custom resource belongs to (e.g. 'all'). This is published in API discovery documents, and used by clients to support invocations like `kubectl get all`.

- **acceptedNames.listKind** (string)

listKind is the serialized kind of the list for this resource. Defaults to " kind List".

- **acceptedNames.shortNames** ([]string)

shortNames are short names for the resource, exposed in API discovery documents, and used by clients to support invocations like `kubectl get \<shortname>`. It must be all lowercase.

- **acceptedNames.singular** (string)

singular is the singular name of the resource. It must be all

lowercase. Defaults to lowercased `kind`.

- **conditions** (`[]CustomResourceDefinitionCondition`)

Map: unique values on key type will be kept during a merge

conditions indicate state for particular aspects of a `CustomResourceDefinition`

CustomResourceDefinitionCondition contains details for the current condition of this pod.

- **conditions.status** (`string`), required

`status` is the status of the condition. Can be `True`, `False`, `Unknown`.

- **conditions.type** (`string`), required

`type` is the type of the condition. Types include `Established`, `NamesAccepted` and `Terminating`.

- **conditions.lastTransitionTime** (`Time`)

`lastTransitionTime` last time the condition transitioned from one status to another.

Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

- **conditions.message** (`string`)

`message` is a human-readable message indicating details about last transition.

- **conditions.reason** (`string`)

`reason` is a unique, one-word, CamelCase reason for the condition's last transition.

- **storedVersions** (`[]string`)

`storedVersions` lists all versions of `CustomResources` that were ever persisted. Tracking these versions allows a migration path for stored versions in `etcd`. The field is mutable so a migration controller can finish a migration to another version (ensuring no old objects are left in storage), and then remove the rest of the versions from this list. Versions may not be removed from `spec.versions` while they exist in this list.

CustomResourceDefinitionList

`CustomResourceDefinitionList` is a list of `CustomResourceDefinition` objects.

- **items** (`[]CustomResourceDefinition`), required

`items` list individual `CustomResourceDefinition` objects

- **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** (ListMeta)

Standard object's metadata More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

Operations

get read the specified CustomResourceDefinition

HTTP Request

GET /apis/apiextensions.k8s.io/v1/customresourcedefinitions/{name}

Parameters

- **name** (*in path*): string, required

name of the CustomResourceDefinition

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([CustomResourceDefinition](#)): OK

401: Unauthorized

get read status of the specified CustomResourceDefinition

HTTP Request

GET /apis/apiextensions.k8s.io/v1/customresourcedefinitions/{name}/status

Parameters

- **name** (*in path*): string, required

name of the CustomResourceDefinition

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([CustomResourceDefinition](#)): OK

401: Unauthorized

list list or watch objects of kind CustomResourceDefinition

HTTP Request

GET /apis/apiextensions.k8s.io/v1/customresourcedefinitions

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([CustomResourceDefinitionList](#)): OK

401: Unauthorized

create create a CustomResourceDefinition

HTTP Request

POST /apis/apiextensions.k8s.io/v1/customresourcedefinitions

Parameters

- **body**: [CustomResourceDefinition](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([CustomResourceDefinition](#)): OK

201 ([CustomResourceDefinition](#)): Created

202 ([CustomResourceDefinition](#)): Accepted

401: Unauthorized

update replace the specified CustomResourceDefinition

HTTP Request

PUT /apis/apiextensions.k8s.io/v1/customresourcedefinitions/{name}

Parameters

- **name** (*in path*): string, required

name of the CustomResourceDefinition

- **body**: [CustomResourceDefinition](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([CustomResourceDefinition](#)): OK

201 ([CustomResourceDefinition](#)): Created

401: Unauthorized

update replace status of the specified

CustomResourceDefinition

HTTP Request

PUT /apis/apiextensions.k8s.io/v1/customresourcedefinitions/{name}/status

Parameters

- **name** (*in path*): string, required
name of the CustomResourceDefinition
- **body**: [CustomResourceDefinition](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([CustomResourceDefinition](#)): OK

201 ([CustomResourceDefinition](#)): Created

401: Unauthorized

patch partially update the specified
CustomResourceDefinition

HTTP Request

PATCH /apis/apiextensions.k8s.io/v1/customresourcedefinitions/{name}

Parameters

- **name** (*in path*): string, required
name of the CustomResourceDefinition
- **body**: [Patch](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([CustomResourceDefinition](#)): OK

201 ([CustomResourceDefinition](#)): Created

401: Unauthorized

patch partially update status of the specified CustomResourceDefinition

HTTP Request

PATCH /apis/apiextensions.k8s.io/v1/customresourcedefinitions/{name}/status

Parameters

- **name** (*in path*): string, required

name of the CustomResourceDefinition

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([CustomResourceDefinition](#)): OK

201 ([CustomResourceDefinition](#)): Created

401: Unauthorized

delete delete a CustomResourceDefinition

HTTP Request

DELETE /apis/apiextensions.k8s.io/v1/customresourcedefinitions/{name}

Parameters

- **name** (*in path*): string, required
name of the CustomResourceDefinition
- **body**: [DeleteOptions](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)

Response

200 ([Status](#)): OK

202 ([Status](#)): Accepted

401: Unauthorized

deletecollection delete collection of
CustomResourceDefinition

HTTP Request

DELETE /apis/apiextensions.k8s.io/v1/customresourcedefinitions

Parameters

- **body**: [DeleteOptions](#)
- **continue** (*in query*): string
[continue](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)

- **propagationPolicy** (*in query*): string
[propagationPolicy](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.7.2 - MutatingWebhookConfiguration

MutatingWebhookConfiguration describes the configuration of an admission webhook that accept or reject and may change the object.

```
apiVersion: admissionregistration.k8s.io/v1  
import "k8s.io/api/admissionregistration/v1"
```

MutatingWebhookConfiguration

MutatingWebhookConfiguration describes the configuration of an admission webhook that accept or reject and may change the object.

- **apiVersion**: admissionregistration.k8s.io/v1
- **kind**: MutatingWebhookConfiguration
- **metadata** ([ObjectMeta](#))

Standard object metadata; More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>.

- **webhooks** ([]MutatingWebhook)

Patch strategy: merge on key name

Webhooks is a list of webhooks and the affected resources and operations.

MutatingWebhook describes an admission webhook and the resources and operations it applies to.

- **webhooks.admissionReviewVersions** ([]string), required

AdmissionReviewVersions is an ordered list of preferred AdmissionReview versions the Webhook expects. API server will try to use first version in the list which it supports. If none of the versions specified in this list supported by API server, validation will fail for this object. If a persisted webhook configuration specifies allowed versions and does not include any versions known to the API Server, calls to the webhook will fail and be subject to the failure policy.

- **webhooks.clientConfig** (WebhookClientConfig), required

ClientConfig defines how to communicate with the hook.
Required

WebhookClientConfig contains the information to make a TLS connection with the webhook

- **webhooks.clientConfig.caBundle** ([]byte)

caBundle is a PEM encoded CA bundle which will be used to validate the webhook's server certificate. If unspecified, system trust roots on the apiserver are used.

- **webhooks.clientConfig.service** (ServiceReference)

`service` is a reference to the service for this webhook.
Either `service` or `url` must be specified.

If the webhook is running within the cluster, then you should use `service`.

ServiceReference holds a reference to Service.legacy.k8s.io

- **webhooks.clientConfig.service.name** (string), required

`name` is the name of the service. Required

- **webhooks.clientConfig.service.namespace** (string), required

`namespace` is the namespace of the service.
Required

- **webhooks.clientConfig.service.path** (string)

`path` is an optional URL path which will be sent in any request to this service.

- **webhooks.clientConfig.service.port** (int32)

If specified, the port on the service that hosting webhook. Default to 443 for backward compatibility. `port` should be a valid port number (1-65535, inclusive).

- **webhooks.clientConfig.url** (string)

`url` gives the location of the webhook, in standard URL form (`scheme://host:port/path`). Exactly one of `url` or `service` must be specified.

The `host` should not refer to a service running in the cluster; use the `service` field instead. The host might be resolved via external DNS in some apiservers (e.g., `kube-apiserver` cannot resolve in-cluster DNS as that would be a layering violation). `host` may also be an IP address.

Please note that using `localhost` or `127.0.0.1` as a `host` is risky unless you take great care to run this webhook on all hosts which run an apiserver which might need to make calls to this webhook. Such installs are likely to be non-portable, i.e., not easy to turn up in a new cluster.

The scheme must be "https"; the URL must begin with "https://".

A path is optional, and if present may be any string permissible in a URL. You may use the path to pass an arbitrary string to the webhook, for example, a cluster identifier.

Attempting to use a user or basic auth e.g. "user:password@" is not allowed. Fragments ("#...") and query parameters ("?...") are not allowed, either.

- **webhooks.name** (string), required

The name of the admission webhook. Name should be fully qualified, e.g., `imagepolicy.kubernetes.io`, where "imagepolicy" is the name of the webhook, and `kubernetes.io` is the name of the organization. Required.

- **webhooks.sideEffects** (string), required

SideEffects states whether this webhook has side effects. Acceptable values are: None, NoneOnDryRun (webhooks created via v1beta1 may also specify Some or Unknown). Webhooks with side effects MUST implement a reconciliation system, since a request may be rejected by a future step in the admission chain and the side effects therefore need to be undone. Requests with the dryRun attribute will be auto-rejected if they match a webhook with sideEffects == Unknown or Some.

- **webhooks.failurePolicy** (string)

FailurePolicy defines how unrecognized errors from the admission endpoint are handled - allowed values are Ignore or Fail. Defaults to Fail.

- **webhooks.matchConditions** ([]MatchCondition)

Patch strategy: merge on key name

Map: unique values on key name will be kept during a merge

MatchConditions is a list of conditions that must be met for a request to be sent to this webhook. Match conditions filter requests that have already been matched by the rules, namespaceSelector, and objectSelector. An empty list of matchConditions matches all requests. There are a maximum of 64 match conditions allowed.

The exact matching logic is (in order):

1. If ANY matchCondition evaluates to FALSE, the webhook is skipped.
2. If ALL matchConditions evaluate to TRUE, the webhook is called.
3. If any matchCondition evaluates to an error (but none are FALSE):
 - If failurePolicy=Fail, reject the request
 - If failurePolicy=Ignore, the error is ignored and the webhook is skipped

This is a beta feature and managed by the AdmissionWebhookMatchConditions feature gate.

MatchCondition represents a condition which must be fulfilled for a request to be sent to a webhook.

- **webhooks.matchConditions.expression** (string), required

Expression represents the expression which will be evaluated by CEL. Must evaluate to bool. CEL expressions have access to the contents of the AdmissionRequest and Authorizer, organized into CEL variables:

'object' - The object from the incoming request. The value is null for DELETE requests. 'oldObject' - The existing object. The value is null for CREATE requests. 'request' - Attributes of the admission request(/pkg/apis/admission/types.go#AdmissionRequest). 'authorizer' - A CEL Authorizer. May be used to perform authorization checks for the principal (user or service account) of the request.

See <https://pkg.go.dev/k8s.io/apiserver/pkg/cel/library#Authz> 'authorizer.requestResource' - A CEL

ResourceCheck constructed from the 'authorizer' and configured with the request resource. Documentation on CEL: <https://kubernetes.io/docs/reference/using-api/cel/>

Required.

- **webhooks.matchConditions.name** (string), required

Name is an identifier for this match condition, used for strategic merging of MatchConditions, as well as providing an identifier for logging purposes. A good name should be descriptive of the associated expression. Name must be a qualified name consisting of alphanumeric characters, '-', '' or '.', and must start and end with an alphanumeric character (e.g. 'MyName', or 'my.name', or '123-abc', regex used for validation is '([A-Za-z0-9][A-Za-z0-9.]*[A-Za-z0-9])' with an optional DNS subdomain prefix and '/' (e.g. 'example.com/MyName')

Required.

- **webhooks.matchPolicy** (string)

matchPolicy defines how the "rules" list is used to match incoming requests. Allowed values are "Exact" or "Equivalent".

- Exact: match a request only if it exactly matches a specified rule. For example, if deployments can be modified via apps/v1, apps/v1beta1, and extensions/v1beta1, but "rules" only included `apiGroups:["apps"]`, `apiVersions:["v1"]`, `resources: ["deployments"]`, a request to apps/v1beta1 or extensions/v1beta1 would not be sent to the webhook.
- Equivalent: match a request if modifies a resource listed in rules, even via another API group or version. For example, if deployments can be modified via apps/v1, apps/v1beta1, and extensions/v1beta1, and "rules" only included `apiGroups:["apps"]`, `apiVersions:["v1"]`, `resources: ["deployments"]`, a request to apps/v1beta1 or extensions/v1beta1 would be converted to apps/v1 and sent to the webhook.

Defaults to "Equivalent"

- **webhooks.namespaceSelector** ([LabelSelector](#))

NamespaceSelector decides whether to run the webhook on an object based on whether the namespace for that object matches the selector. If the object itself is a namespace, the matching is performed on object.metadata.labels. If the object is another cluster scoped resource, it never skips the webhook.

For example, to run the webhook on any objects whose namespace is not associated with "runlevel" of "0" or "1"; you will set the selector as follows: "namespaceSelector": { "matchExpressions": [{ "key": "runlevel", "operator": "NotIn", "values": ["0", "1"] }] }

If instead you want to only run the webhook on any objects whose namespace is associated with the "environment" of "prod" or "staging"; you will set the selector as follows: "namespaceSelector": { "matchExpressions": [{ "key": "environment", "operator": "In", "values": ["prod", "staging"] }] }

See <https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/> for more examples of label selectors.

Default to the empty LabelSelector, which matches everything.

- **webhooks.objectSelector** ([LabelSelector](#))

ObjectSelector decides whether to run the webhook based on if the object has matching labels. objectSelector is evaluated against both the oldObject and newObject that would be sent to the webhook, and is considered to match if either object matches the selector. A null object (oldObject in the case of create, or newObject in the case of delete) or an object that cannot have labels (like a DeploymentRollback or a PodProxyOptions object) is not considered to match. Use the object selector only if the webhook is opt-in, because end users may skip the admission webhook by setting the labels. Default to the empty LabelSelector, which matches everything.

- **webhooks.reinvocationPolicy** (string)

reinvocationPolicy indicates whether this webhook should be called multiple times as part of a single admission evaluation. Allowed values are "Never" and "IfNeeded".

Never: the webhook will not be called more than once in a single admission evaluation.

IfNeeded: the webhook will be called at least one additional time as part of the admission evaluation if the object being admitted is modified by other admission plugins after the initial webhook call. Webhooks that specify this option *must* be idempotent, able to process objects they previously admitted. Note: * the number of additional invocations is not guaranteed to be exactly one. * if additional invocations result in further modifications to the object, webhooks are not guaranteed to be invoked again. * webhooks that use this option may be reordered to minimize the number of additional invocations. * to validate an object after all mutations are guaranteed complete, use a validating admission webhook instead.

Defaults to "Never".

- **webhooks.rules** ([]RuleWithOperations)

Rules describes what operations on what resources/ subresources the webhook cares about. The webhook cares about an operation if it matches *any* Rule. However, in order to prevent ValidatingAdmissionWebhooks and MutatingAdmissionWebhooks from putting the cluster in a state which cannot be recovered from without completely disabling the plugin, ValidatingAdmissionWebhooks and MutatingAdmissionWebhooks are never called on admission requests for ValidatingWebhookConfiguration and MutatingWebhookConfiguration objects.

RuleWithOperations is a tuple of Operations and Resources. It is recommended to make sure that all the tuple expansions are valid.

- **webhooks.rules.apiGroups** ([]string)

Atomic: will be replaced during a merge

APIGroups is the API groups the resources belong to. " is

all groups. If " is present, the length of the slice must be one. Required.

- **webhooks.rules.apiVersions** ([]string)

Atomic: will be replaced during a merge

APIVersions is the API versions the resources belong to. "*is all versions. If* " is present, the length of the slice must be one. Required.

- **webhooks.rules.operations** ([]string)

Atomic: will be replaced during a merge

Operations is the operations the admission hook cares about - CREATE, UPDATE, DELETE, CONNECT or * for all of those operations and any future admission operations that are added. If '*' is present, the length of the slice must be one. Required.

- **webhooks.rules.resources** ([]string)

Atomic: will be replaced during a merge

Resources is a list of resources this rule applies to.

For example: 'pods' means pods. 'pods/log' means the log subresource of pods. "*means all resources, but not subresources. 'pods/* means all subresources of pods. '/*scale*' means all scale subresources. '/*' means all resources and their subresources.

If wildcard is present, the validation rule will ensure resources do not overlap with each other.

Depending on the enclosing object, subresources might not be allowed. Required.

- **webhooks.rules.scope** (string)

scope specifies the scope of this rule. Valid values are "Cluster", "Namespaced", and "" "*Cluster*" means that only cluster-scoped resources will match this rule. Namespace API objects are cluster-scoped. "Namespaced" means that only namespaced resources will match this rule. "" means that there are no scope restrictions. Subresources match the scope of their parent resource. Default is "*".

- **webhooks.timeoutSeconds** (int32)

TimeoutSeconds specifies the timeout for this webhook. After the timeout passes, the webhook call will be ignored or the API call will fail based on the failure policy. The timeout value must be between 1 and 30 seconds. Default to 10 seconds.

MutatingWebhookConfigurationList

MutatingWebhookConfigurationList is a list of MutatingWebhookConfiguration.

- **apiVersion**: admissionregistration.k8s.io/v1

- **kind**: MutatingWebhookConfigurationList

- **metadata** ([ListMeta](#))

Standard list metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds>

- **items** ([] [MutatingWebhookConfiguration](#)), required

List of MutatingWebhookConfiguration.

Operations

get read the specified MutatingWebhookConfiguration

HTTP Request

GET /apis/admissionregistration.k8s.io/v1/
mutatingwebhookconfigurations/{name}

Parameters

- **name** (*in path*): string, required

name of the MutatingWebhookConfiguration

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([MutatingWebhookConfiguration](#)): OK

401: Unauthorized

list list or watch objects of kind MutatingWebhookConfiguration

HTTP Request

GET /apis/admissionregistration.k8s.io/v1/
mutatingwebhookconfigurations

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string
[pretty](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)
- **watch** (*in query*): boolean
[watch](#)

Response

200 ([MutatingWebhookConfigurationList](#)): OK

401: Unauthorized

create create a MutatingWebhookConfiguration

HTTP Request

POST /apis/admissionregistration.k8s.io/v1/
mutatingwebhookconfigurations

Parameters

- **body**: [MutatingWebhookConfiguration](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([MutatingWebhookConfiguration](#)): OK

201 ([MutatingWebhookConfiguration](#)): Created

202 ([MutatingWebhookConfiguration](#)): Accepted

401: Unauthorized

update replace the specified MutatingWebhookConfiguration

HTTP Request

PUT /apis/admissionregistration.k8s.io/v1/
mutatingwebhookconfigurations/{name}

Parameters

- **name** (*in path*): string, required
name of the MutatingWebhookConfiguration
- **body**: [MutatingWebhookConfiguration](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([MutatingWebhookConfiguration](#)): OK

201 ([MutatingWebhookConfiguration](#)): Created

401: Unauthorized

patch partially update the specified MutatingWebhookConfiguration

HTTP Request

PATCH /apis/admissionregistration.k8s.io/v1/
mutatingwebhookconfigurations/{name}

Parameters

- **name** (*in path*): string, required
name of the MutatingWebhookConfiguration
- **body**: [Patch](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([MutatingWebhookConfiguration](#)): OK

201 ([MutatingWebhookConfiguration](#)): Created

401: Unauthorized

delete delete a MutatingWebhookConfiguration

HTTP Request

DELETE /apis/admissionregistration.k8s.io/v1/
mutatingwebhookconfigurations/{name}

Parameters

- **name** (*in path*): string, required

name of the MutatingWebhookConfiguration

- **body**: [DeleteOptions](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

Response

200 ([Status](#)): OK

202 ([Status](#)): Accepted

401: Unauthorized

deletecollection delete collection of MutatingWebhookConfiguration

HTTP Request

DELETE /apis/admissionregistration.k8s.io/v1/
mutatingwebhookconfigurations

Parameters

- **body**: [DeleteOptions](#)
- **continue** (*in query*): string
[continue](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.7.3 - ValidatingWebhookConfiguration

ValidatingWebhookConfiguration describes the configuration of an admission webhook that accept or reject an object without changing it.

```
apiVersion: admissionregistration.k8s.io/v1
import "k8s.io/api/admissionregistration/v1"
```

ValidatingWebhookConfiguration

ValidatingWebhookConfiguration describes the configuration of an admission webhook that accept or reject an object without changing it.

- **apiVersion**: admissionregistration.k8s.io/v1
- **kind**: ValidatingWebhookConfiguration
- **metadata** ([ObjectMeta](#))

Standard object metadata; More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>.

- **webhooks** ([]ValidatingWebhook)

Patch strategy: merge on key name

Webhooks is a list of webhooks and the affected resources and operations.

ValidatingWebhook describes an admission webhook and the resources and operations it applies to.

- **webhooks.admissionReviewVersions** ([]string), required

AdmissionReviewVersions is an ordered list of preferred AdmissionReview versions the Webhook expects. API server will try to use first version in the list which it supports. If none of the versions specified in this list supported by API server, validation will fail for this object. If a persisted webhook configuration specifies allowed versions and does not include any versions known to the API Server, calls to the webhook will fail and be subject to the failure policy.

- **webhooks.clientConfig** (WebhookClientConfig), required

ClientConfig defines how to communicate with the hook.
Required

WebhookClientConfig contains the information to make a TLS connection with the webhook

- **webhooks.clientConfig.caBundle** ([]byte)

caBundle is a PEM encoded CA bundle which will be used to validate the webhook's server certificate. If unspecified, system trust roots on the apiserver are used.

- **webhooks.clientConfig.service** (ServiceReference)

`service` is a reference to the service for this webhook.
Either `service` or `url` must be specified.

If the webhook is running within the cluster, then you should use `service`.

ServiceReference holds a reference to Service.legacy.k8s.io

- **webhooks.clientConfig.service.name** (string), required

`name` is the name of the service. Required

- **webhooks.clientConfig.service.namespace** (string), required

`namespace` is the namespace of the service.
Required

- **webhooks.clientConfig.service.path** (string)

`path` is an optional URL path which will be sent in any request to this service.

- **webhooks.clientConfig.service.port** (int32)

If specified, the port on the service that hosting webhook. Default to 443 for backward compatibility. `port` should be a valid port number (1-65535, inclusive).

- **webhooks.clientConfig.url** (string)

`url` gives the location of the webhook, in standard URL form (`scheme://host:port/path`). Exactly one of `url` or `service` must be specified.

The `host` should not refer to a service running in the cluster; use the `service` field instead. The host might be resolved via external DNS in some apiservers (e.g., `kube-apiserver` cannot resolve in-cluster DNS as that would be a layering violation). `host` may also be an IP address.

Please note that using `localhost` or `127.0.0.1` as a `host` is risky unless you take great care to run this webhook on all hosts which run an apiserver which might need to make calls to this webhook. Such installs are likely to be non-portable, i.e., not easy to turn up in a new cluster.

The scheme must be "https"; the URL must begin with "https://".

A path is optional, and if present may be any string permissible in a URL. You may use the path to pass an arbitrary string to the webhook, for example, a cluster identifier.

Attempting to use a user or basic auth e.g. "user:password@" is not allowed. Fragments ("#...") and query parameters ("?...") are not allowed, either.

- **webhooks.name** (string), required

The name of the admission webhook. Name should be fully qualified, e.g., `imagepolicy.kubernetes.io`, where "imagepolicy" is the name of the webhook, and `kubernetes.io` is the name of the organization. Required.

- **webhooks.sideEffects** (string), required

SideEffects states whether this webhook has side effects. Acceptable values are: None, NoneOnDryRun (webhooks created via v1beta1 may also specify Some or Unknown). Webhooks with side effects MUST implement a reconciliation system, since a request may be rejected by a future step in the admission chain and the side effects therefore need to be undone. Requests with the dryRun attribute will be auto-rejected if they match a webhook with sideEffects == Unknown or Some.

- **webhooks.failurePolicy** (string)

FailurePolicy defines how unrecognized errors from the admission endpoint are handled - allowed values are Ignore or Fail. Defaults to Fail.

- **webhooks.matchConditions** ([]MatchCondition)

Patch strategy: merge on key name

Map: unique values on key name will be kept during a merge

MatchConditions is a list of conditions that must be met for a request to be sent to this webhook. Match conditions filter requests that have already been matched by the rules, namespaceSelector, and objectSelector. An empty list of matchConditions matches all requests. There are a maximum of 64 match conditions allowed.

The exact matching logic is (in order):

1. If ANY matchCondition evaluates to FALSE, the webhook is skipped.
2. If ALL matchConditions evaluate to TRUE, the webhook is called.
3. If any matchCondition evaluates to an error (but none are FALSE):
 - If failurePolicy=Fail, reject the request
 - If failurePolicy=Ignore, the error is ignored and the webhook is skipped

This is a beta feature and managed by the AdmissionWebhookMatchConditions feature gate.

MatchCondition represents a condition which must be fulfilled for a request to be sent to a webhook.

- **webhooks.matchConditions.expression** (string), required

Expression represents the expression which will be evaluated by CEL. Must evaluate to bool. CEL expressions have access to the contents of the AdmissionRequest and Authorizer, organized into CEL variables:

'object' - The object from the incoming request. The value is null for DELETE requests. 'oldObject' - The existing object. The value is null for CREATE requests. 'request' - Attributes of the admission request(/pkg/apis/admission/types.go#AdmissionRequest). 'authorizer' - A CEL Authorizer. May be used to perform authorization checks for the principal (user or service account) of the request.

See <https://pkg.go.dev/k8s.io/apiserver/pkg/cel/library#Authz> 'authorizer.requestResource' - A CEL

ResourceCheck constructed from the 'authorizer' and configured with the request resource. Documentation on CEL: <https://kubernetes.io/docs/reference/using-api/cel/>

Required.

- **webhooks.matchConditions.name** (string), required

Name is an identifier for this match condition, used for strategic merging of MatchConditions, as well as providing an identifier for logging purposes. A good name should be descriptive of the associated expression. Name must be a qualified name consisting of alphanumeric characters, '-', '' or '.', and must start and end with an alphanumeric character (e.g. 'MyName', or 'my.name', or '123-abc', regex used for validation is '([A-Za-z0-9][A-Za-z0-9.]*[A-Za-z0-9])' with an optional DNS subdomain prefix and '/' (e.g. 'example.com/MyName')

Required.

- **webhooks.matchPolicy** (string)

matchPolicy defines how the "rules" list is used to match incoming requests. Allowed values are "Exact" or "Equivalent".

- Exact: match a request only if it exactly matches a specified rule. For example, if deployments can be modified via apps/v1, apps/v1beta1, and extensions/v1beta1, but "rules" only included `apiGroups:["apps"]`, `apiVersions:["v1"]`, `resources: ["deployments"]`, a request to apps/v1beta1 or extensions/v1beta1 would not be sent to the webhook.
- Equivalent: match a request if modifies a resource listed in rules, even via another API group or version. For example, if deployments can be modified via apps/v1, apps/v1beta1, and extensions/v1beta1, and "rules" only included `apiGroups:["apps"]`, `apiVersions:["v1"]`, `resources: ["deployments"]`, a request to apps/v1beta1 or extensions/v1beta1 would be converted to apps/v1 and sent to the webhook.

Defaults to "Equivalent"

- **webhooks.namespaceSelector** ([LabelSelector](#))

NamespaceSelector decides whether to run the webhook on an object based on whether the namespace for that object matches the selector. If the object itself is a namespace, the matching is performed on object.metadata.labels. If the object is another cluster scoped resource, it never skips the webhook.

For example, to run the webhook on any objects whose namespace is not associated with "runlevel" of "0" or "1"; you will set the selector as follows: "namespaceSelector": { "matchExpressions": [{ "key": "runlevel", "operator": "NotIn", "values": ["0", "1"] }] }

If instead you want to only run the webhook on any objects whose namespace is associated with the "environment" of "prod" or "staging"; you will set the selector as follows: "namespaceSelector": { "matchExpressions": [{ "key": "environment", "operator": "In", "values": ["prod", "staging"] }] }

See <https://kubernetes.io/docs/concepts/overview/working-with-objects/labels> for more examples of label selectors.

Default to the empty LabelSelector, which matches everything.

- **webhooks.objectSelector** ([LabelSelector](#))

ObjectSelector decides whether to run the webhook based on if the object has matching labels. objectSelector is evaluated against both the oldObject and newObject that would be sent to the webhook, and is considered to match if either object matches the selector. A null object (oldObject in the case of create, or newObject in the case of delete) or an object that cannot have labels (like a DeploymentRollback or a PodProxyOptions object) is not considered to match. Use the object selector only if the webhook is opt-in, because end users may skip the admission webhook by setting the labels. Default to the empty LabelSelector, which matches everything.

- **webhooks.rules** ([]RuleWithOperations)

Rules describes what operations on what resources/ subresources the webhook cares about. The webhook cares about an operation if it matches *any* Rule. However, in order to prevent ValidatingAdmissionWebhooks and MutatingAdmissionWebhooks from putting the cluster in a state which cannot be recovered from without completely disabling the plugin, ValidatingAdmissionWebhooks and MutatingAdmissionWebhooks are never called on admission requests for ValidatingWebhookConfiguration and MutatingWebhookConfiguration objects.

RuleWithOperations is a tuple of Operations and Resources. It is recommended to make sure that all the tuple expansions are valid.

- **webhooks.rules.apiGroups** ([]string)

Atomic: will be replaced during a merge

APIGroups is the API groups the resources belong to. "*is all groups*. If " is present, the length of the slice must be one. Required.

- **webhooks.rules.apiVersions** ([]string)

Atomic: will be replaced during a merge

APIVersions is the API versions the resources belong to. "*is all versions*. If " is present, the length of the slice must be one. Required.

- **webhooks.rules.operations** ([]string)

Atomic: will be replaced during a merge

Operations is the operations the admission hook cares about - CREATE, UPDATE, DELETE, CONNECT or * for all of those operations and any future admission operations that are added. If '*' is present, the length of the slice must be one. Required.

- **webhooks.rules.resources** ([]string)

Atomic: will be replaced during a merge

Resources is a list of resources this rule applies to.

For example: 'pods' means pods. 'pods/log' means the log subresource of pods. *" means all resources, but not subresources.* 'pods/' means all subresources of pods. *'/scale' means all scale subresources.* '/*' means all resources and their subresources.

If wildcard is present, the validation rule will ensure resources do not overlap with each other.

Depending on the enclosing object, subresources might not be allowed. Required.

- **webhooks.rules.scope** (string)

scope specifies the scope of this rule. Valid values are "Cluster", "Namespaced", and "" *"Cluster" means that only cluster-scoped resources will match this rule. Namespace API objects are cluster-scoped. "Namespaced" means that only namespaced resources will match this rule.* "" means that there are no scope restrictions. Subresources match the scope of their parent resource. Default is "*".

- **webhooks.timeoutSeconds** (int32)

TimeoutSeconds specifies the timeout for this webhook. After the timeout passes, the webhook call will be ignored or the API call will fail based on the failure policy. The timeout value must be between 1 and 30 seconds. Default to 10 seconds.

ValidatingWebhookConfigurationList

ValidatingWebhookConfigurationList is a list of ValidatingWebhookConfiguration.

- **apiVersion**: admissionregistration.k8s.io/v1
- **kind**: ValidatingWebhookConfigurationList
- **metadata** ([ListMeta](#))

Standard list metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds>

- **items** ([] [ValidatingWebhookConfiguration](#)), required

List of ValidatingWebhookConfiguration.

Operations

get read the specified ValidatingWebhookConfiguration

HTTP Request

GET /apis/admissionregistration.k8s.io/v1/
validatingwebhookconfigurations/{name}

Parameters

- **name** (*in path*): string, required
name of the ValidatingWebhookConfiguration
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([ValidatingWebhookConfiguration](#)): OK

401: Unauthorized

list list or watch objects of kind ValidatingWebhookConfiguration

HTTP Request

GET /apis/admissionregistration.k8s.io/v1/
validatingwebhookconfigurations

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([ValidatingWebhookConfigurationList](#)): OK

401: Unauthorized

create create a ValidatingWebhookConfiguration

HTTP Request

POST /apis/admissionregistration.k8s.io/v1/
validatingwebhookconfigurations

Parameters

- **body**: [ValidatingWebhookConfiguration](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([ValidatingWebhookConfiguration](#)): OK

201 ([ValidatingWebhookConfiguration](#)): Created

202 ([ValidatingWebhookConfiguration](#)): Accepted

401: Unauthorized

update replace the specified ValidatingWebhookConfiguration

HTTP Request

PUT /apis/admissionregistration.k8s.io/v1/
validatingwebhookconfigurations/{name}

Parameters

- **name** (*in path*): string, required
name of the ValidatingWebhookConfiguration
- **body**: [ValidatingWebhookConfiguration](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ValidatingWebhookConfiguration](#)): OK

201 ([ValidatingWebhookConfiguration](#)): Created

401: Unauthorized

patch partially update the specified ValidatingWebhookConfiguration

HTTP Request

PATCH /apis/admissionregistration.k8s.io/v1/validatingwebhookconfigurations/{name}

Parameters

- **name** (*in path*): string, required

name of the ValidatingWebhookConfiguration

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ValidatingWebhookConfiguration](#)): OK

201 ([ValidatingWebhookConfiguration](#)): Created

401: Unauthorized

delete delete a ValidatingWebhookConfiguration

HTTP Request

DELETE /apis/admissionregistration.k8s.io/v1/validatingwebhookconfigurations/{name}

Parameters

- **name** (*in path*): string, required
name of the ValidatingWebhookConfiguration
- **body**: [DeleteOptions](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)

Response

200 ([Status](#)): OK

202 ([Status](#)): Accepted

401: Unauthorized

deletecollection delete collection of
ValidatingWebhookConfiguration

HTTP Request

DELETE /apis/admissionregistration.k8s.io/v1/
validatingwebhookconfigurations

Parameters

- **body**: [DeleteOptions](#)
- **continue** (*in query*): string
[continue](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)

- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.7.4 - ValidatingAdmissionPolicy v1beta1

ValidatingAdmissionPolicy describes the definition of an admission validation policy that accepts or rejects an object without changing it.

```
apiVersion: admissionregistration.k8s.io/v1beta1

import "k8s.io/api/admissionregistration/v1beta1"
```

ValidatingAdmissionPolicy

ValidatingAdmissionPolicy describes the definition of an admission validation policy that accepts or rejects an object without changing it.

- **apiVersion**: admissionregistration.k8s.io/v1beta1

- **kind**: ValidatingAdmissionPolicy

- **metadata** ([ObjectMeta](#))

Standard object metadata; More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>.

- **spec** (ValidatingAdmissionPolicySpec)

Specification of the desired behavior of the ValidatingAdmissionPolicy.

ValidatingAdmissionPolicySpec is the specification of the desired behavior of the AdmissionPolicy.

- **spec.auditAnnotations** ([]AuditAnnotation)

Atomic: will be replaced during a merge

auditAnnotations contains CEL expressions which are used to produce audit annotations for the audit event of the API request. validations and auditAnnotations may not both be empty; at least one of validations or auditAnnotations is required.

AuditAnnotation describes how to produce an audit annotation for an API request.

- **spec.auditAnnotations.key** (string), required

key specifies the audit annotation key. The audit annotation keys of a ValidatingAdmissionPolicy must be unique. The key must be a qualified name ([A-Za-z0-9][-A-Za-z0-9_.]*) no more than 63 bytes in length.

The key is combined with the resource name of the ValidatingAdmissionPolicy to construct an audit annotation key: "{ValidatingAdmissionPolicy name}/{key}".

If an admission webhook uses the same resource name as this ValidatingAdmissionPolicy and the same audit annotation key, the annotation key will be identical. In

this case, the first annotation written with the key will be included in the audit event and all subsequent annotations with the same key will be discarded.

Required.

- **spec.auditAnnotations.valueExpression** (string), required

valueExpression represents the expression which is evaluated by CEL to produce an audit annotation value. The expression must evaluate to either a string or null value. If the expression evaluates to a string, the audit annotation is included with the string value. If the expression evaluates to null or empty string the audit annotation will be omitted. The valueExpression may be no longer than 5kb in length. If the result of the valueExpression is more than 10kb in length, it will be truncated to 10kb.

If multiple ValidatingAdmissionPolicyBinding resources match an API request, then the valueExpression will be evaluated for each binding. All unique values produced by the valueExpressions will be joined together in a comma-separated list.

Required.

- **spec.failurePolicy** (string)

failurePolicy defines how to handle failures for the admission policy. Failures can occur from CEL expression parse errors, type check errors, runtime errors and invalid or mis-configured policy definitions or bindings.

A policy is invalid if spec.paramKind refers to a non-existent Kind. A binding is invalid if spec.paramRef.name refers to a non-existent resource.

failurePolicy does not define how validations that evaluate to false are handled.

When failurePolicy is set to Fail, ValidatingAdmissionPolicyBinding validationActions define how failures are enforced.

Allowed values are Ignore or Fail. Defaults to Fail.

- **spec.matchConditions** ([]MatchCondition)

Patch strategy: merge on key name

Map: unique values on key name will be kept during a merge

MatchConditions is a list of conditions that must be met for a request to be validated. Match conditions filter requests that have already been matched by the rules, namespaceSelector, and objectSelector. An empty list of matchConditions matches all requests. There are a maximum of 64 match conditions allowed.

If a parameter object is provided, it can be accessed via the `params` handle in the same manner as validation expressions.

The exact matching logic is (in order):

1. If ANY matchCondition evaluates to FALSE, the policy is skipped.

2. If ALL matchConditions evaluate to TRUE, the policy is evaluated.
3. If any matchCondition evaluates to an error (but none are FALSE):
 - If failurePolicy=Fail, reject the request
 - If failurePolicy=Ignore, the policy is skipped

MatchCondition represents a condition which must be fulfilled for a request to be sent to a webhook.

- **spec.matchConditions.expression** (string), required

Expression represents the expression which will be evaluated by CEL. Must evaluate to bool. CEL expressions have access to the contents of the AdmissionRequest and Authorizer, organized into CEL variables:

'object' - The object from the incoming request. The value is null for DELETE requests. 'oldObject' - The existing object. The value is null for CREATE requests. 'request' - Attributes of the admission request(/pkg/apis/admission/types.go#AdmissionRequest). 'authorizer' - A CEL Authorizer. May be used to perform authorization checks for the principal (user or service account) of the request.

See <https://pkg.go.dev/k8s.io/apiserver/pkg/cel/library#Authz> 'authorizer.requestResource' - A CEL ResourceCheck constructed from the 'authorizer' and configured with the request resource. Documentation on CEL: <https://kubernetes.io/docs/reference/using-api/cel/>

Required.

- **spec.matchConditions.name** (string), required

Name is an identifier for this match condition, used for strategic merging of MatchConditions, as well as providing an identifier for logging purposes. A good name should be descriptive of the associated expression. Name must be a qualified name consisting of alphanumeric characters, '-' or '.', and must start and end with an alphanumeric character (e.g. 'MyName', or 'my.name', or '123-abc', regex used for validation is '([A-Za-z0-9][-A-Za-z0-9.]*)?[A-Za-z0-9]') with an optional DNS subdomain prefix and '/' (e.g. 'example.com/MyName')

Required.

- **spec.matchConstraints** (MatchResources)

MatchConstraints specifies what resources this policy is designed to validate. The AdmissionPolicy cares about a request if it matches *all* Constraints. However, in order to prevent clusters from being put into an unstable state that cannot be recovered from via the API

ValidatingAdmissionPolicy cannot match

ValidatingAdmissionPolicy and

ValidatingAdmissionPolicyBinding. Required.

MatchResources decides whether to run the admission control policy on an object based on whether it meets the match criteria. The exclude rules take precedence over include rules (if a resource matches both, it is excluded)

- **spec.matchConstraints.excludeResourceRules** ([]NamedRuleWithOperations)

Atomic: will be replaced during a merge

ExcludeResourceRules describes what operations on what resources/subresources the ValidatingAdmissionPolicy should not care about. The exclude rules take precedence over include rules (if a resource matches both, it is excluded)

NamedRuleWithOperations is a tuple of Operations and Resources with ResourceNames.

- **spec.matchConstraints.excludeResourceRules.apiGroups**
([]string)

Atomic: will be replaced during a merge

APIGroups is the API groups the resources belong to. *"is all groups. If"* is present, the length of the slice must be one. Required.

- **spec.matchConstraints.excludeResourceRules.apiVersions**
([]string)

Atomic: will be replaced during a merge

APIVersions is the API versions the resources belong to. *"is all versions. If"* is present, the length of the slice must be one. Required.

- **spec.matchConstraints.excludeResourceRules.operations**
([]string)

Atomic: will be replaced during a merge

Operations is the operations the admission hook cares about - CREATE, UPDATE, DELETE, CONNECT or * for all of those operations and any future admission operations that are added. If '*' is present, the length of the slice must be one. Required.

- **spec.matchConstraints.excludeResourceRules.resourceNames**
([]string)

Atomic: will be replaced during a merge

ResourceNames is an optional white list of names that the rule applies to. An empty set means that everything is allowed.

- **spec.matchConstraints.excludeResourceRules.resources**
([]string)

Atomic: will be replaced during a merge

Resources is a list of resources this rule applies to.

For example: 'pods' means pods. 'pods/log' means the log subresource of pods. *"means all resources, but not subresources. 'pods/' means all subresources of pods. '/scale' means all scale subresources. '/*' means all resources and their subresources.*

If wildcard is present, the validation rule will ensure resources do not overlap with each other.

Depending on the enclosing object, subresources might not be allowed. Required.

- **spec.matchConstraints.excludeResourceRules.scope**
(string)

scope specifies the scope of this rule. Valid values are "Cluster", "Namespaced", and "" "*Cluster*" means that only cluster-scoped resources will match this rule. Namespace API objects are cluster-scoped. "Namespaced" means that only namespaced resources will match this rule. "" means that there are no scope restrictions. Subresources match the scope of their parent resource. Default is "*".

- **spec.matchConstraints.matchPolicy** (string)

matchPolicy defines how the "MatchResources" list is used to match incoming requests. Allowed values are "Exact" or "Equivalent".

- Exact: match a request only if it exactly matches a specified rule. For example, if deployments can be modified via apps/v1, apps/v1beta1, and extensions/v1beta1, but "rules" only included `apiGroups:["apps"], apiVersions:["v1"], resources: ["deployments"]`, a request to apps/v1beta1 or extensions/v1beta1 would not be sent to the ValidatingAdmissionPolicy.
- Equivalent: match a request if modifies a resource listed in rules, even via another API group or version. For example, if deployments can be modified via apps/v1, apps/v1beta1, and extensions/v1beta1, and "rules" only included `apiGroups:["apps"], apiVersions:["v1"], resources: ["deployments"]`, a request to apps/v1beta1 or extensions/v1beta1 would be converted to apps/v1 and sent to the ValidatingAdmissionPolicy.

Defaults to "Equivalent"

- **spec.matchConstraints.namespaceSelector**
([LabelSelector](#))

NamespaceSelector decides whether to run the admission control policy on an object based on whether the namespace for that object matches the selector. If the object itself is a namespace, the matching is performed on object.metadata.labels. If the object is another cluster scoped resource, it never skips the policy.

For example, to run the webhook on any objects whose namespace is not associated with "runlevel" of "0" or "1"; you will set the selector as follows: "namespaceSelector": { "matchExpressions": [{ "key": "runlevel", "operator": "NotIn", "values": ["0", "1"] }] }

If instead you want to only run the policy on any objects whose namespace is associated with the "environment" of "prod" or "staging"; you will set the selector as follows: "namespaceSelector": { "matchExpressions": [{ "key": "environment", "operator": "In", "values": ["prod", "staging"] }] }

See <https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/> for more examples of label

selectors.

Default to the empty LabelSelector, which matches everything.

- **spec.matchConstraints.objectSelector** ([LabelSelector](#))

ObjectSelector decides whether to run the validation based on if the object has matching labels. objectSelector is evaluated against both the oldObject and newObject that would be sent to the cel validation, and is considered to match if either object matches the selector. A null object (oldObject in the case of create, or newObject in the case of delete) or an object that cannot have labels (like a DeploymentRollback or a PodProxyOptions object) is not considered to match. Use the object selector only if the webhook is opt-in, because end users may skip the admission webhook by setting the labels. Default to the empty LabelSelector, which matches everything.

- **spec.matchConstraints.resourceRules**

([]NamedRuleWithOperations)

Atomic: will be replaced during a merge

ResourceRules describes what operations on what resources/subresources the ValidatingAdmissionPolicy matches. The policy cares about an operation if it matches *any* Rule.

NamedRuleWithOperations is a tuple of Operations and Resources with ResourceNames.

- **spec.matchConstraints.resourceRules.apiGroups**
([]string)

Atomic: will be replaced during a merge

APIGroups is the API groups the resources belong to. *"is all groups. If"* is present, the length of the slice must be one. Required.

- **spec.matchConstraints.resourceRules.apiVersions**
([]string)

Atomic: will be replaced during a merge

APIVersions is the API versions the resources belong to. *"is all versions. If"* is present, the length of the slice must be one. Required.

- **spec.matchConstraints.resourceRules.operations**
([]string)

Atomic: will be replaced during a merge

Operations is the operations the admission hook cares about - CREATE, UPDATE, DELETE, CONNECT or * for all of those operations and any future admission operations that are added. If '*' is present, the length of the slice must be one. Required.

- **spec.matchConstraints.resourceRules.resourceNames**
([]string)

Atomic: will be replaced during a merge

ResourceNames is an optional white list of names that the rule applies to. An empty set means that everything is allowed.

- **spec.matchConstraints.resourceRules.resources**
([]string)

Atomic: will be replaced during a merge

Resources is a list of resources this rule applies to.

For example: 'pods' means pods. 'pods/log' means the log subresource of pods. *" means all resources, but not subresources. 'pods/' means all subresources of pods. '/scale' means all scale subresources. '//*' means all resources and their subresources.*

If wildcard is present, the validation rule will ensure resources do not overlap with each other.

Depending on the enclosing object, subresources might not be allowed. Required.

- **spec.matchConstraints.resourceRules.scope**
(string)

scope specifies the scope of this rule. Valid values are "Cluster", "Namespaced", and *" Cluster" means that only cluster-scoped resources will match this rule. Namespace API objects are cluster-scoped. "Namespaced" means that only namespaced resources will match this rule. "" means that there are no scope restrictions. Subresources match the scope of their parent resource. Default is "*".*

- **spec.paramKind** (ParamKind)

ParamKind specifies the kind of resources used to parameterize this policy. If absent, there are no parameters for this policy and the param CEL variable will not be provided to validation expressions. If ParamKind refers to a non-existent kind, this policy definition is mis-configured and the FailurePolicy is applied. If paramKind is specified but paramRef is unset in ValidatingAdmissionPolicyBinding, the params variable will be null.

ParamKind is a tuple of Group Kind and Version.

- **spec.paramKind.apiVersion** (string)

APIVersion is the API group version the resources belong to. In format of "group/version". Required.

- **spec.paramKind.kind** (string)

Kind is the API kind the resources belong to. Required.

- **spec.validations** ([]Validation)

Atomic: will be replaced during a merge

Validations contain CEL expressions which is used to apply the validation. Validations and AuditAnnotations may not both be empty; a minimum of one Validations or AuditAnnotations is required.

Validation specifies the CEL expression which is used to apply the validation.

- **spec.validations.expression** (string), required

Expression represents the expression which will be evaluated by CEL. ref: <https://github.com/google/cel-spec>
CEL expressions have access to the contents of the API request/response, organized into CEL variables as well as some other useful variables:

- 'object' - The object from the incoming request. The value is null for DELETE requests. - 'oldObject' - The existing object. The value is null for CREATE requests. - 'request' - Attributes of the API request([ref](#)). - 'params' - Parameter resource referred to by the policy binding being evaluated. Only populated if the policy has a ParamKind. - 'namespaceObject' - The namespace object that the incoming object belongs to. The value is null for cluster-scoped resources. - 'variables' - Map of composited variables, from its name to its lazily evaluated value. For example, a variable named 'foo' can be accessed as 'variables.foo'.
- 'authorizer' - A CEL Authorizer. May be used to perform authorization checks for the principal (user or service account) of the request. See <https://pkg.go.dev/k8s.io/apiserver/pkg/cel/library#Authz>
- 'authorizer.requestResource' - A CEL ResourceCheck constructed from the 'authorizer' and configured with the request resource.

The `apiVersion`, `kind`, `metadata.name` and `metadata.generateName` are always accessible from the root of the object. No other metadata properties are accessible.

Only property names of the form `[a-zA-Z_.-/][a-zA-Z0-9_.-/*]` are accessible. Accessible property names are escaped according to the following rules when accessed in the expression: - **escapes to 'underscores'** - `'` **escapes to 'dot'** - `'` **escapes to 'dash'** - `'/` **escapes to 'slash'** - **Property names that exactly match a CEL RESERVED keyword escape to '{keyword}__'**. The keywords are: "true", "false", "null", "in", "as", "break", "const", "continue", "else", "for", "function", "if", "import", "let", "loop", "package", "namespace", "return". Examples:

- Expression accessing a property named "namespace": {"Expression": "object.namespace > 0"}
- Expression accessing a property named "x-prop": {"Expression": "object.x_dash_prop > 0"}
- Expression accessing a property named "redact_d": {"Expression": "object.redact_underscores_d > 0"}

Equality on arrays with list type of 'set' or 'map' ignores element order, i.e. `[1, 2] == [2, 1]`. Concatenation on arrays with x-kubernetes-list-type use the semantics of the list type:

- 'set': `x + y` performs a union where the array positions of all elements in `x` are preserved and non-intersecting elements in `y` are appended, retaining their partial order.
- 'map': `x + y` performs a merge where the array positions of all keys in `x` are preserved but the

values are overwritten by values in `y` when the key sets of `x` and `y` intersect. Elements in `y` with non-intersecting keys are appended, retaining their partial order. Required.

- **spec.validations.message** (string)

Message represents the message displayed when validation fails. The message is required if the Expression contains line breaks. The message must not contain line breaks. If unset, the message is "failed rule: {Rule}". e.g. "must be a URL with the host matching spec.host" If the Expression contains line breaks. Message is required. The message must not contain line breaks. If unset, the message is "failed Expression: {Expression}".

- **spec.validations.messageExpression** (string)

`messageExpression` declares a CEL expression that evaluates to the validation failure message that is returned when this rule fails. Since `messageExpression` is used as a failure message, it must evaluate to a string. If both `message` and `messageExpression` are present on a validation, then `messageExpression` will be used if validation fails. If `messageExpression` results in a runtime error, the runtime error is logged, and the validation failure message is produced as if the `messageExpression` field were unset. If `messageExpression` evaluates to an empty string, a string with only spaces, or a string that contains line breaks, then the validation failure message will also be produced as if the `messageExpression` field were unset, and the fact that `messageExpression` produced an empty string/string with only spaces/string with line breaks will be logged. `messageExpression` has access to all the same variables as the `expression` except for 'authorizer' and 'authorizer.requestResource'.

Example: "object.x must be less than max
(" + string(params.max) + ")"

- **spec.validations.reason** (string)

Reason represents a machine-readable description of why this validation failed. If this is the first validation in the list to fail, this reason, as well as the corresponding HTTP response code, are used in the HTTP response to the client. The currently supported reasons are: "Unauthorized", "Forbidden", "Invalid", "RequestEntityTooLarge". If not set, `StatusReasonInvalid` is used in the response to the client.

- **spec.variables** ([]Variable)

Patch strategy: merge on key name

Map: unique values on key name will be kept during a merge

Variables contain definitions of variables that can be used in composition of other expressions. Each variable is defined as a named CEL expression. The variables defined here will be available under `variables` in other expressions of the policy except MatchConditions because MatchConditions are evaluated before the rest of the policy.

The expression of a variable can refer to other variables defined earlier in the list but not those after. Thus, Variables must be sorted by the order of first appearance and acyclic.

Variable is the definition of a variable that is used for composition. A variable is defined as a named expression.

- **spec.variables.expression** (string), required

Expression is the expression that will be evaluated as the value of the variable. The CEL expression has access to the same identifiers as the CEL expressions in Validation.

- **spec.variables.name** (string), required

Name is the name of the variable. The name must be a valid CEL identifier and unique among all variables. The variable can be accessed in other expressions through `variables`. For example, if name is "foo", the variable will be available as `variables.foo`

- **status** (ValidatingAdmissionPolicyStatus)

The status of the ValidatingAdmissionPolicy, including warnings that are useful to determine if the policy behaves in the expected way. Populated by the system. Read-only.

ValidatingAdmissionPolicyStatus represents the status of an admission validation policy.

- **status.conditions** ([]Condition)

Map: unique values on key type will be kept during a merge

The conditions represent the latest available observations of a policy's current state.

Condition contains details for one aspect of the current state of this API Resource.

- **status.conditions.lastTransitionTime** (Time), required

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.

Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

- **status.conditions.message** (string), required

message is a human readable message indicating details about the transition. This may be an empty string.

- **status.conditions.reason** (string), required

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.conditions.status** (string), required

status of the condition, one of True, False, Unknown.

- **status.conditions.type** (string), required

type of condition in CamelCase or in foo.example.com/

CamelCase.

- **status.conditions.observedGeneration** (int64)

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.

- **status.observedGeneration** (int64)

The generation observed by the controller.

- **status.typeChecking** (TypeChecking)

The results of type checking for each expression. Presence of this field indicates the completion of the type checking.

TypeChecking contains results of type checking the expressions in the ValidatingAdmissionPolicy

- **status.typeChecking.expressionWarnings**

([]ExpressionWarning)

Atomic: will be replaced during a merge

The type checking warnings for each expression.

ExpressionWarning is a warning information that targets a specific expression.

- **status.typeChecking.expressionWarnings.fieldRef**

(string), required

The path to the field that refers the expression. For example, the reference to the expression of the first item of validations is "spec.validations[0].expression"

- **status.typeChecking.expressionWarnings.warning**

(string), required

The content of type checking information in a human-readable form. Each line of the warning contains the type that the expression is checked against, followed by the type check error from the compiler.

ValidatingAdmissionPolicyList

ValidatingAdmissionPolicyList is a list of ValidatingAdmissionPolicy.

- **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>

- **items** ([]ValidatingAdmissionPolicy)

List of ValidatingAdmissionPolicy.

- **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** (ListMeta)

Standard list metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds>

ValidatingAdmissionPolicyBinding

ValidatingAdmissionPolicyBinding binds the ValidatingAdmissionPolicy with parameterized resources. ValidatingAdmissionPolicyBinding and parameter CRDs together define how cluster administrators configure policies for clusters.

For a given admission request, each binding will cause its policy to be evaluated N times, where N is 1 for policies/bindings that don't use params, otherwise N is the number of parameters selected by the binding.

The CEL expressions of a policy must have a computed CEL cost below the maximum CEL budget. Each evaluation of the policy is given an independent CEL cost budget. Adding/removing policies, bindings, or params can not affect whether a given (policy, binding, param) combination is within its own CEL budget.

- **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)

Standard object metadata; More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>.

- **spec** (ValidatingAdmissionPolicyBindingSpec)

Specification of the desired behavior of the ValidatingAdmissionPolicyBinding.

ValidatingAdmissionPolicyBindingSpec is the specification of the ValidatingAdmissionPolicyBinding.

- **spec.matchResources** (MatchResources)

MatchResources declares what resources match this binding and will be validated by it. Note that this is intersected with the

policy's matchConstraints, so only requests that are matched by the policy can be selected by this. If this is unset, all resources matched by the policy are validated by this binding. When resourceRules is unset, it does not constrain resource matching. If a resource is matched by the other fields of this object, it will be validated. Note that this is differs from ValidatingAdmissionPolicy matchConstraints, where resourceRules are required.

MatchResources decides whether to run the admission control policy on an object based on whether it meets the match criteria. The exclude rules take precedence over include rules (if a resource matches both, it is excluded)

- **spec.matchResources.excludeResourceRules**
([]NamedRuleWithOperations)

Atomic: will be replaced during a merge

ExcludeResourceRules describes what operations on what resources/subresources the ValidatingAdmissionPolicy should not care about. The exclude rules take precedence over include rules (if a resource matches both, it is excluded)

NamedRuleWithOperations is a tuple of Operations and Resources with ResourceNames.

- **spec.matchResources.excludeResourceRules.apiGroups**
([]string)

Atomic: will be replaced during a merge

APIGroups is the API groups the resources belong to. *"is all groups. If "* is present, the length of the slice must be one. Required.

- **spec.matchResources.excludeResourceRules.apiVersions**
([]string)

Atomic: will be replaced during a merge

APIVersions is the API versions the resources belong to. *"is all versions. If "* is present, the length of the slice must be one. Required.

- **spec.matchResources.excludeResourceRules.operations**
([]string)

Atomic: will be replaced during a merge

Operations is the operations the admission hook cares about - CREATE, UPDATE, DELETE, CONNECT or * for all of those operations and any future admission operations that are added. If '*' is present, the length of the slice must be one. Required.

- **spec.matchResources.excludeResourceRules.resourceNames**
([]string)

Atomic: will be replaced during a merge

ResourceNames is an optional white list of names that the rule applies to. An empty set means that everything is allowed.

- **spec.matchResources.excludeResourceRules.resources**

([]string)

Atomic: will be replaced during a merge

Resources is a list of resources this rule applies to.

For example: 'pods' means pods. 'pods/log' means the log subresource of pods. *" means all resources, but not subresources. 'pods/' means all subresources of pods. '/scale' means all scale subresources. '/*' means all resources and their subresources.*

If wildcard is present, the validation rule will ensure resources do not overlap with each other.

Depending on the enclosing object, subresources might not be allowed. Required.

- **spec.matchResources.excludeResourceRules.scope**

(string)

scope specifies the scope of this rule. Valid values are "Cluster", "Namespaced", and *" Cluster means that only cluster-scoped resources will match this rule.*

Namespace API objects are cluster-scoped.

"Namespaced" means that only namespaced resources will match this rule. "" means that there are no scope restrictions. Subresources match the scope of their parent resource. Default is "".*

- **spec.matchResources.matchPolicy** (string)

matchPolicy defines how the "MatchResources" list is used to match incoming requests. Allowed values are "Exact" or "Equivalent".

- Exact: match a request only if it exactly matches a specified rule. For example, if deployments can be modified via apps/v1, apps/v1beta1, and extensions/v1beta1, but "rules" only included

```
apiGroups:["apps"], apiVersions:["v1"], resources:[ "deployments" ] , a request to apps/v1beta1 or extensions/v1beta1 would not be sent to the ValidatingAdmissionPolicy.
```

- Equivalent: match a request if modifies a resource listed in rules, even via another API group or version. For example, if deployments can be modified via apps/v1, apps/v1beta1, and extensions/v1beta1, and "rules" only included

```
apiGroups:["apps"], apiVersions:["v1"], resources:[ "deployments" ] , a request to apps/v1beta1 or extensions/v1beta1 would be converted to apps/v1 and sent to the ValidatingAdmissionPolicy.
```

Defaults to "Equivalent"

- **spec.matchResources.namespaceSelector**

([LabelSelector](#))

NamespaceSelector decides whether to run the admission control policy on an object based on whether the namespace for that object matches the selector. If the object itself is a namespace, the matching is performed on object.metadata.labels. If the object is another cluster scoped resource, it never skips the

policy.

For example, to run the webhook on any objects whose namespace is not associated with "runlevel" of "0" or "1"; you will set the selector as follows: "namespaceSelector": { "matchExpressions": [{ "key": "runlevel", "operator": "NotIn", "values": ["0", "1"] }] }

If instead you want to only run the policy on any objects whose namespace is associated with the "environment" of "prod" or "staging"; you will set the selector as follows: "namespaceSelector": { "matchExpressions": [{ "key": "environment", "operator": "In", "values": ["prod", "staging"] }] }

See <https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/> for more examples of label selectors.

Default to the empty LabelSelector, which matches everything.

- **spec.matchResources.objectSelector** ([LabelSelector](#))

ObjectSelector decides whether to run the validation based on if the object has matching labels. objectSelector is evaluated against both the oldObject and newObj that would be sent to the cel validation, and is considered to match if either object matches the selector. A null object (oldObject in the case of create, or newObj in the case of delete) or an object that cannot have labels (like a DeploymentRollback or a PodProxyOptions object) is not considered to match. Use the object selector only if the webhook is opt-in, because end users may skip the admission webhook by setting the labels. Default to the empty LabelSelector, which matches everything.

- **spec.matchResources.resourceRules**

([][NamedRuleWithOperations](#))

Atomic: will be replaced during a merge

ResourceRules describes what operations on what resources/subresources the ValidatingAdmissionPolicy matches. The policy cares about an operation if it matches *any* Rule.

NamedRuleWithOperations is a tuple of Operations and Resources with ResourceNames.

- **spec.matchResources.resourceRules.apiGroups**
([][string](#))

Atomic: will be replaced during a merge

APIGroups is the API groups the resources belong to. *"is all groups. If"* is present, the length of the slice must be one. Required.

- **spec.matchResources.resourceRules.apiVersions**
([][string](#))

Atomic: will be replaced during a merge

APIVersions is the API versions the resources belong to. *"is all versions. If"* is present, the length

of the slice must be one. Required.

- **spec.matchResources.resourceRules.operations**
([]string)

Atomic: will be replaced during a merge

Operations is the operations the admission hook cares about - CREATE, UPDATE, DELETE, CONNECT or * for all of those operations and any future admission operations that are added. If '*' is present, the length of the slice must be one. Required.

- **spec.matchResources.resourceRules.resourceNames**
([]string)

Atomic: will be replaced during a merge

ResourceNames is an optional white list of names that the rule applies to. An empty set means that everything is allowed.

- **spec.matchResources.resourceRules.resources**
([]string)

Atomic: will be replaced during a merge

Resources is a list of resources this rule applies to.

For example: 'pods' means pods. 'pods/log' means the log subresource of pods. *" means all resources, but not subresources. '/pods/' means all subresources of pods. '/scale' means all scale subresources. '//*' means all resources and their subresources.*

If wildcard is present, the validation rule will ensure resources do not overlap with each other.

Depending on the enclosing object, subresources might not be allowed. Required.

- **spec.matchResources.resourceRules.scope**
(string)

scope specifies the scope of this rule. Valid values are "Cluster", "Namespaced", and "" *"Cluster" means that only cluster-scoped resources will match this rule. Namespace API objects are cluster-scoped. "Namespaced" means that only namespaced resources will match this rule. "" means that there are no scope restrictions. Subresources match the scope of their parent resource. Default is "*".*

- **spec.paramRef** (ParamRef)

paramRef specifies the parameter resource used to configure the admission control policy. It should point to a resource of the type specified in ParamKind of the bound ValidatingAdmissionPolicy. If the policy specifies a ParamKind and the resource referred to by ParamRef does not exist, this binding is considered mis-configured and the FailurePolicy of the ValidatingAdmissionPolicy applied. If the policy does not specify a ParamKind then this field is ignored, and the rules are evaluated without a param.

ParamRef describes how to locate the params to be used as input

to expressions of rules applied by a policy binding.

- **spec.paramRef.name** (string)

name is the name of the resource being referenced.

One of `name` or `selector` must be set, but `name` and `selector` are mutually exclusive properties. If one is set, the other must be unset.

A single parameter used for all admission requests can be configured by setting the `name` field, leaving `selector` blank, and setting `namespace` if `paramKind` is namespace-scoped.

- **spec.paramRef.namespace** (string)

`namespace` is the namespace of the referenced resource.

Allows limiting the search for params to a specific namespace. Applies to both `name` and `selector` fields.

A per-namespace parameter may be used by specifying a namespace-scoped `paramKind` in the policy and leaving this field empty.

- If `paramKind` is cluster-scoped, this field MUST be unset. Setting this field results in a configuration error.
- If `paramKind` is namespace-scoped, the namespace of the object being evaluated for admission will be used when this field is left unset. Take care that if this is left empty the binding must not match any cluster-scoped resources, which will result in an error.

- **spec.paramRef.parameterNotFoundAction** (string)

`parameterNotFoundAction` controls the behavior of the binding when the resource exists, and `name` or `selector` is valid, but there are no parameters matched by the binding. If the value is set to `Allow`, then no matched parameters will be treated as successful validation by the binding. If set to `Deny`, then no matched parameters will be subject to the `failurePolicy` of the policy.

Allowed values are `Allow` or `Deny`

Required

- **spec.paramRef.selector** ([LabelSelector](#))

`selector` can be used to match multiple param objects based on their labels. Supply `selector: {}` to match all resources of the `ParamKind`.

If multiple params are found, they are all evaluated with the policy expressions and the results are ANDed together.

One of `name` or `selector` must be set, but `name` and `selector` are mutually exclusive properties. If one is set, the other must be unset.

- **spec.policyName** (string)

`PolicyName` references a `ValidatingAdmissionPolicy` name which the `ValidatingAdmissionPolicyBinding` binds to. If the

referenced resource does not exist, this binding is considered invalid and will be ignored. Required.

- **spec.validationActions ([]string)**

Set: unique values will be kept during a merge

validationActions declares how Validations of the referenced ValidatingAdmissionPolicy are enforced. If a validation evaluates to false it is always enforced according to these actions.

Failures defined by the ValidatingAdmissionPolicy's FailurePolicy are enforced according to these actions only if the FailurePolicy is set to Fail, otherwise the failures are ignored. This includes compilation errors, runtime errors and misconfigurations of the policy.

validationActions is declared as a set of action values. Order does not matter. validationActions may not contain duplicates of the same action.

The supported actions values are:

"Deny" specifies that a validation failure results in a denied request.

"Warn" specifies that a validation failure is reported to the request client in HTTP Warning headers, with a warning code of 299. Warnings can be sent both for allowed or denied admission responses.

"Audit" specifies that a validation failure is included in the published audit event for the request. The audit event will contain a `validation.policy.admission.k8s.io/validation_failure` audit annotation with a value containing the details of the validation failures, formatted as a JSON list of objects, each with the following fields:

- `message`: The validation failure message string
- `policy`: The resource name of the ValidatingAdmissionPolicy
- `binding`: The resource name of the ValidatingAdmissionPolicyBinding
- `expressionIndex`: The index of the failed validations in the ValidatingAdmissionPolicy
- `validationActions`: The enforcement actions enacted for the validation failure

Example audit annotation:

```
"validation.policy.admission.k8s.io/validation_failure":  
  "[{"message": "Invalid value", "policy":  
    "policy.example.com", "binding":  
    "policybinding.example.com", "expressionIndex": "1",  
    "validationActions": ["Audit"]}]"
```

Clients should expect to handle additional values by ignoring any values not recognized.

"Deny" and "Warn" may not be used together since this combination needlessly duplicates the validation failure both in the API response body and the HTTP warning headers.

Required.

Operations

get read the specified ValidatingAdmissionPolicy

HTTP Request

GET /apis/admissionregistration.k8s.io/v1beta1/
validatingadmissionpolicies/{name}

Parameters

- **name** (*in path*): string, required
name of the ValidatingAdmissionPolicy
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([ValidatingAdmissionPolicy](#)): OK

401: Unauthorized

get read status of the specified ValidatingAdmissionPolicy

HTTP Request

GET /apis/admissionregistration.k8s.io/v1beta1/
validatingadmissionpolicies/{name}/status

Parameters

- **name** (*in path*): string, required
name of the ValidatingAdmissionPolicy
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([ValidatingAdmissionPolicy](#)): OK

401: Unauthorized

list list or watch objects of kind ValidatingAdmissionPolicy

HTTP Request

GET /apis/admissionregistration.k8s.io/v1beta1/
validatingadmissionpolicies

Parameters

- **allowWatchBookmarks** (*in query*): boolean
[allowWatchBookmarks](#)
- **continue** (*in query*): string
[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([ValidatingAdmissionPolicyList](#)): OK

401: Unauthorized

create create a ValidatingAdmissionPolicy

HTTP Request

POST /apis/admissionregistration.k8s.io/v1beta1/
validatingadmissionpolicies

Parameters

- **body**: [ValidatingAdmissionPolicy](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ValidatingAdmissionPolicy](#)): OK

201 ([ValidatingAdmissionPolicy](#)): Created

202 ([ValidatingAdmissionPolicy](#)): Accepted

401: Unauthorized

update replace the specified ValidatingAdmissionPolicy

HTTP Request

PUT /apis/admissionregistration.k8s.io/v1beta1/
validatingadmissionpolicies/{name}

Parameters

- **name** (*in path*): string, required

name of the ValidatingAdmissionPolicy

- **body**: [ValidatingAdmissionPolicy](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ValidatingAdmissionPolicy](#)): OK

201 ([ValidatingAdmissionPolicy](#)): Created

401: Unauthorized

update replace status of the specified
ValidatingAdmissionPolicy

HTTP Request

PUT /apis/admissionregistration.k8s.io/v1beta1/
validatingadmissionpolicies/{name}/status

Parameters

- **name** (*in path*): string, required

name of the ValidatingAdmissionPolicy

- **body**: [ValidatingAdmissionPolicy](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ValidatingAdmissionPolicy](#)): OK

201 ([ValidatingAdmissionPolicy](#)): Created

401: Unauthorized

patch partially update the specified [ValidatingAdmissionPolicy](#)

HTTP Request

PATCH /apis/admissionregistration.k8s.io/v1beta1/
validatingadmissionpolicies/{name}

Parameters

- **name** (*in path*): string, required

name of the [ValidatingAdmissionPolicy](#)

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ValidatingAdmissionPolicy](#)): OK

201 ([ValidatingAdmissionPolicy](#)): Created

401: Unauthorized

patch partially update status of the specified [ValidatingAdmissionPolicy](#)

ValidatingAdmissionPolicy

HTTP Request

PATCH /apis/admissionregistration.k8s.io/v1beta1/
validatingadmissionpolicies/{name}/status

Parameters

- **name** (*in path*): string, required
name of the ValidatingAdmissionPolicy
- **body**: [Patch](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **force** (*in query*): boolean
[force](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([ValidatingAdmissionPolicy](#)): OK

201 ([ValidatingAdmissionPolicy](#)): Created

401: Unauthorized

delete delete a ValidatingAdmissionPolicy

HTTP Request

DELETE /apis/admissionregistration.k8s.io/v1beta1/
validatingadmissionpolicies/{name}

Parameters

- **name** (*in path*): string, required
name of the ValidatingAdmissionPolicy
- **body**: [DeleteOptions](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **pretty** (*in query*): string
[pretty](#)

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

Response

200 ([Status](#)): OK

202 ([Status](#)): Accepted

401: Unauthorized

deletecollection delete collection of
ValidatingAdmissionPolicy

HTTP Request

DELETE /apis/admissionregistration.k8s.io/v1beta1/
validatingadmissionpolicies

Parameters

- **body**: [DeleteOptions](#)

- **continue** (*in query*): string

[continue](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.8 - Cluster Resources

5.8.1 - Node

Node is a worker node in Kubernetes.

```
apiVersion: v1

import "k8s.io/api/core/v1"
```

Node

Node is a worker node in Kubernetes. Each node will have a unique identifier in the cache (i.e. in etcd).

- **apiVersion**: v1

- **kind**: Node

- **metadata** ([ObjectMeta](#))

Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **spec** ([NodeSpec](#))

Spec defines the behavior of a node. <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status>

- **status** ([NodeStatus](#))

Most recently observed status of the node. Populated by the system. Read-only. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status>

NodeSpec

NodeSpec describes the attributes that a node is created with.

- **configSource** ([NodeConfigSource](#))

Deprecated: Previously used to specify the source of the node's configuration for the DynamicKubeletConfig feature. This feature is removed.

NodeConfigSource specifies a source of node configuration. Exactly one subfield (excluding metadata) must be non-nil. This API is deprecated since 1.22

- **configSource.configMap** ([ConfigMapNodeConfigSource](#))

ConfigMap is a reference to a Node's ConfigMap

ConfigMapNodeConfigSource contains the information to reference a ConfigMap as a config source for the Node. This API is deprecated since 1.22: <https://git.k8s.io/enhancements/keps/sig-node/281-dynamic-kubelet-configuration>

- **configSource.configMap.kubeletConfigKey** (string),

required

KubeletConfigKey declares which key of the referenced ConfigMap corresponds to the KubeletConfiguration structure. This field is required in all cases.

- **configSource.configMap.name** (string), required

Name is the metadata.name of the referenced ConfigMap. This field is required in all cases.

- **configSource.configMap.namespace** (string), required

Namespace is the metadata.namespace of the referenced ConfigMap. This field is required in all cases.

- **configSource.configMap.resourceVersion** (string)

ResourceVersion is the metadata.ResourceVersion of the referenced ConfigMap. This field is forbidden in Node.Spec, and required in Node.Status.

- **configSource.configMap.uid** (string)

UID is the metadata.UID of the referenced ConfigMap. This field is forbidden in Node.Spec, and required in Node.Status.

- **externalID** (string)

Deprecated. Not all kubelets will set this field. Remove field after 1.13. see: <https://issues.k8s.io/61966>

- **podCIDR** (string)

PodCIDR represents the pod IP range assigned to the node.

- **podCIDRs** ([]string)

podCIDRs represents the IP ranges assigned to the node for usage by Pods on that node. If this field is specified, the 0th entry must match the podCIDR field. It may contain at most 1 value for each of IPv4 and IPv6.

- **providerID** (string)

ID of the node assigned by the cloud provider in the format:
<ProviderName>://<ProviderSpecificNodeID>

- **taints** ([]Taint)

If specified, the node's taints.

The node this Taint is attached to has the "effect" on any pod that does not tolerate the Taint.

- **taints.effect** (string), required

Required. The effect of the taint on pods that do not tolerate the taint. Valid effects are NoSchedule, PreferNoSchedule and NoExecute.

- **taints.key** (string), required

Required. The taint key to be applied to a node.

- **taints.timeAdded** (Time)

TimeAdded represents the time at which the taint was added. It is only written for NoExecute taints.

Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

- **taints.value** (string)

The taint value corresponding to the taint key.

- **unschedulable** (boolean)

Unschedulable controls node schedulability of new pods. By default, node is schedulable. More info: <https://kubernetes.io/docs/concepts/nodes/node/#manual-node-administration>

NodeStatus

NodeStatus is information about the current status of a node.

- **addresses** ([]NodeAddress)

Patch strategy: merge on key type

List of addresses reachable to the node. Queried from cloud provider, if available. More info: <https://kubernetes.io/docs/concepts/nodes/node/#addresses> Note: This field is declared as mergeable, but the merge key is not sufficiently unique, which can cause data corruption when it is merged. Callers should instead use a full-replacement patch. See <https://pr.k8s.io/79391> for an example. Consumers should assume that addresses can change during the lifetime of a Node. However, there are some exceptions where this may not be possible, such as Pods that inherit a Node's address in its own status or consumers of the downward API (status.hostIP).

NodeAddress contains information for the node's address.

- **addresses.address** (string), required

The node address.

- **addresses.type** (string), required

Node address type, one of Hostname, ExternalIP or InternalIP.

- **allocatable** (map[string]Quantity)

Allocatable represents the resources of a node that are available for scheduling. Defaults to Capacity.

- **capacity** (map[string]Quantity)

Capacity represents the total resources of a node. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#capacity>

- **conditions** ([]NodeCondition)

Patch strategy: merge on key type

Conditions is an array of current observed node conditions. More info: <https://kubernetes.io/docs/concepts/nodes/node/#condition>

NodeCondition contains condition information for a node.

- **conditions.status** (string), required

Status of the condition, one of True, False, Unknown.

- **conditions.type** (string), required

Type of node condition.

- **conditions.lastHeartbeatTime** (Time)

Last time we got an update on a given condition.

Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

- **conditions.lastTransitionTime** (Time)

Last time the condition transit from one status to another.

Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

- **conditions.message** (string)

Human readable message indicating details about last transition.

- **conditions.reason** (string)

(brief) reason for the condition's last transition.

- **config** (NodeConfigStatus)

Status of the config assigned to the node via the dynamic Kubelet config feature.

NodeConfigStatus describes the status of the config assigned by Node.Spec.ConfigSource.

- **config.active** (NodeConfigSource)

Active reports the checkpointed config the node is actively using. Active will represent either the current version of the Assigned config, or the current LastKnownGood config, depending on whether attempting to use the Assigned config results in an error.

NodeConfigSource specifies a source of node configuration. Exactly one subfield (excluding metadata) must be non-nil. This API is deprecated since 1.22

- **config.active.configMap** (ConfigMapNodeConfigSource)

ConfigMap is a reference to a Node's ConfigMap

ConfigMapNodeConfigSource contains the information to reference a ConfigMap as a config source for the Node. This API is deprecated since 1.22: <https://git.k8s.io/enhancements/keps/sig-node/281-dynamic-kubelet-configuration>

- **config.active.configMap.kubeletConfigKey**
(string), required

KubeletConfigKey declares which key of the referenced ConfigMap corresponds to the KubeletConfiguration structure. This field is required in all cases.

- **config.active.configMap.name** (string), required

Name is the metadata.name of the referenced

ConfigMap. This field is required in all cases.

- **config.active.configMap.namespace** (string), required

Namespace is the metadata.namespace of the referenced ConfigMap. This field is required in all cases.

- **config.active.configMap.resourceVersion** (string)

ResourceVersion is the metadata.ResourceVersion of the referenced ConfigMap. This field is forbidden in Node.Spec, and required in Node.Status.

- **config.active.configMap.uid** (string)

UID is the metadata.UID of the referenced ConfigMap. This field is forbidden in Node.Spec, and required in Node.Status.

- **config.assigned** (NodeConfigSource)

Assigned reports the checkpointed config the node will try to use. When Node.Spec.ConfigSource is updated, the node checkpoints the associated config payload to local disk, along with a record indicating intended config. The node refers to this record to choose its config checkpoint, and reports this record in Assigned. Assigned only updates in the status after the record has been checkpointed to disk. When the Kubelet is restarted, it tries to make the Assigned config the Active config by loading and validating the checkpointed payload identified by Assigned.

NodeConfigSource specifies a source of node configuration. Exactly one subfield (excluding metadata) must be non-nil. This API is deprecated since 1.22

- **config.assigned.configMap**

(ConfigMapNodeConfigSource)

ConfigMap is a reference to a Node's ConfigMap

ConfigMapNodeConfigSource contains the information to reference a ConfigMap as a config source for the Node. This API is deprecated since 1.22: <https://git.k8s.io/enhancements/keps/sig-node/281-dynamic-kubelet-configuration>

- **config.assigned.configMap.kubeletConfigKey** (string), required

KubeletConfigKey declares which key of the referenced ConfigMap corresponds to the KubeletConfiguration structure. This field is required in all cases.

- **config.assigned.configMap.name** (string), required

Name is the metadata.name of the referenced ConfigMap. This field is required in all cases.

- **config.assigned.configMap.namespace** (string), required

Namespace is the metadata.namespace of the referenced ConfigMap. This field is required in all

cases.

- **config.assigned.configMap.resourceVersion**
(string)

ResourceVersion is the metadata.ResourceVersion of the referenced ConfigMap. This field is forbidden in Node.Spec, and required in Node.Status.

- **config.assigned.configMap.uid** (string)

UID is the metadata.UID of the referenced ConfigMap. This field is forbidden in Node.Spec, and required in Node.Status.

- **config.error** (string)

Error describes any problems reconciling the Spec.ConfigSource to the Active config. Errors may occur, for example, attempting to checkpoint Spec.ConfigSource to the local Assigned record, attempting to checkpoint the payload associated with Spec.ConfigSource, attempting to load or validate the Assigned config, etc. Errors may occur at different points while syncing config. Earlier errors (e.g. download or checkpointing errors) will not result in a rollback to LastKnownGood, and may resolve across Kubelet retries. Later errors (e.g. loading or validating a checkpointed config) will result in a rollback to LastKnownGood. In the latter case, it is usually possible to resolve the error by fixing the config assigned in Spec.ConfigSource. You can find additional information for debugging by searching the error message in the Kubelet log. Error is a human-readable description of the error state; machines can check whether or not Error is empty, but should not rely on the stability of the Error text across Kubelet versions.

- **config.lastKnownGood** (NodeConfigSource)

LastKnownGood reports the checkpointed config the node will fall back to when it encounters an error attempting to use the Assigned config. The Assigned config becomes the LastKnownGood config when the node determines that the Assigned config is stable and correct. This is currently implemented as a 10-minute soak period starting when the local record of Assigned config is updated. If the Assigned config is Active at the end of this period, it becomes the LastKnownGood. Note that if Spec.ConfigSource is reset to nil (use local defaults), the LastKnownGood is also immediately reset to nil, because the local default config is always assumed good. You should not make assumptions about the node's method of determining config stability and correctness, as this may change or become configurable in the future.

NodeConfigSource specifies a source of node configuration. Exactly one subfield (excluding metadata) must be non-nil. This API is deprecated since 1.22

- **config.lastKnownGood.configMap**

(ConfigMapNodeConfigSource)

ConfigMap is a reference to a Node's ConfigMap

ConfigMapNodeConfigSource contains the information to reference a ConfigMap as a config source for the Node. This API is deprecated since 1.22: <https://git.k8s.io/enhancements/keps/sig-node/281-dynamic-kubelet->

[configuration](#)

- **config.lastKnownGood.configMap.kubeletConfigKey**

(string), required

KubeletConfigKey declares which key of the referenced ConfigMap corresponds to the KubeletConfiguration structure. This field is required in all cases.

- **config.lastKnownGood.configMap.name** (string),

required

Name is the metadata.name of the referenced ConfigMap. This field is required in all cases.

- **config.lastKnownGood.configMap.namespace**

(string), required

Namespace is the metadata.namespace of the referenced ConfigMap. This field is required in all cases.

- **config.lastKnownGood.configMap.resourceVersion**

(string)

ResourceVersion is the metadata.ResourceVersion of the referenced ConfigMap. This field is forbidden in Node.Spec, and required in Node.Status.

- **config.lastKnownGood.configMap.uid** (string)

UID is the metadata.UID of the referenced ConfigMap. This field is forbidden in Node.Spec, and required in Node.Status.

- **daemonEndpoints** (NodeDaemonEndpoints)

Endpoints of daemons running on the Node.

NodeDaemonEndpoints lists ports opened by daemons running on the Node.

- **daemonEndpoints.kubeletEndpoint** (DaemonEndpoint)

Endpoint on which Kubelet is listening.

DaemonEndpoint contains information about a single Daemon endpoint.

- **daemonEndpoints.kubeletEndpoint.Port** (int32),

required

Port number of the given endpoint.

- **images** ([]ContainerImage)

List of container images on this node

Describe a container image

- **images.names** ([]string)

Names by which this image is known. e.g.

["kubernetes.example/hyperkube:v1.0.7", "cloud-
vendor.registry.example/cloud-vendor/hyperkube:v1.0.7"]

- **images.sizeBytes** (int64)

The size of the image in bytes.

- **nodeInfo** (NodeSystemInfo)

Set of ids/uuids to uniquely identify the node. More info: <https://kubernetes.io/docs/concepts/nodes/node/#info>

NodeSystemInfo is a set of ids/uuids to uniquely identify the node.

- **nodeInfo.architecture** (string), required

The Architecture reported by the node

- **nodeInfo.bootID** (string), required

Boot ID reported by the node.

- **nodeInfo.containerRuntimeVersion** (string), required

ContainerRuntime Version reported by the node through runtime remote API (e.g. containerd://1.4.2).

- **nodeInfo.kernelVersion** (string), required

Kernel Version reported by the node from 'uname -r' (e.g. 3.16.0-0.bpo.4-amd64).

- **nodeInfo.kubeProxyVersion** (string), required

KubeProxy Version reported by the node.

- **nodeInfo.kubeletVersion** (string), required

Kubelet Version reported by the node.

- **nodeInfo.machineID** (string), required

MachineID reported by the node. For unique machine identification in the cluster this field is preferred. Learn more from man(5) machine-id: <http://man7.org/linux/man-pages/man5/machine-id.5.html>

- **nodeInfo.operatingSystem** (string), required

The Operating System reported by the node

- **nodeInfo.osImage** (string), required

OS Image reported by the node from /etc/os-release (e.g. Debian GNU/Linux 7 (wheezy)).

- **nodeInfo.systemUUID** (string), required

SystemUUID reported by the node. For unique machine identification MachineID is preferred. This field is specific to Red Hat hosts https://access.redhat.com/documentation/en-us/red_hat_subscription_management/1/html/rhsm/uuid

- **phase** (string)

NodePhase is the recently observed lifecycle phase of the node.

More info: <https://kubernetes.io/docs/concepts/nodes/node/#phase> The field is never populated, and now is deprecated.

- **volumesAttached** ([]AttachedVolume)

List of volumes that are attached to the node.

AttachedVolume describes a volume attached to a node

- **volumesAttached.devicePath** (string), required

DevicePath represents the device path where the volume

should be available

- **volumesAttached.name** (string), required

Name of the attached volume

- **volumesInUse** ([]string)

List of attachable volumes in use (mounted) by the node.

NodeList

NodeList is the whole list of all Nodes which have been registered with master.

- **apiVersion**: v1

- **kind**: NodeList

- **metadata** ([ListMeta](#))

Standard list metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds>

- **items** ([]Node), required

List of nodes

Operations

get read the specified Node

HTTP Request

GET /api/v1/nodes/{name}

Parameters

- **name** (*in path*): string, required

name of the Node

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Node](#)): OK

401: Unauthorized

get read status of the specified Node

HTTP Request

GET /api/v1/nodes/{name}/status

Parameters

- **name** (*in path*): string, required

name of the Node

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Node](#)): OK

401: Unauthorized

list list or watch objects of kind Node

HTTP Request

GET /api/v1/nodes

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([NodeList](#)): OK

401: Unauthorized

create create a Node

HTTP Request

POST /api/v1/nodes

Parameters

- **body**: [Node](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([Node](#)): OK

201 ([Node](#)): Created

202 ([Node](#)): Accepted

401: Unauthorized

update replace the specified Node

HTTP Request

PUT /api/v1/nodes/{name}

Parameters

- **name** (*in path*): string, required
name of the Node
- **body**: [Node](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([Node](#)): OK

201 ([Node](#)): Created

401: Unauthorized

update replace status of the specified Node

HTTP Request

PUT /api/v1/nodes/{name}/status

Parameters

- **name** (*in path*): string, required

name of the Node

- **body**: [Node](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Node](#)): OK

201 ([Node](#)): Created

401: Unauthorized

patch partially update the specified Node

HTTP Request

PATCH /api/v1/nodes/{name}

Parameters

- **name** (*in path*): string, required

name of the Node

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Node](#)): OK

201 ([Node](#)): Created

401: Unauthorized

patch partially update status of the specified Node

HTTP Request

PATCH /api/v1/nodes/{name}/status

Parameters

- **name** (*in path*): string, required

name of the Node

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Node](#)): OK

201 ([Node](#)): Created

401: Unauthorized

delete delete a Node

HTTP Request

DELETE /api/v1/nodes/{name}

Parameters

- **name** (*in path*): string, required
name of the Node
- **body**: [DeleteOptions](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)

Response

200 ([Status](#)): OK

202 ([Status](#)): Accepted

401: Unauthorized

deletecollection delete collection of Node

HTTP Request

DELETE /api/v1/nodes

Parameters

- **body**: [DeleteOptions](#)
- **continue** (*in query*): string
[continue](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.8.2 - Namespace

Namespace provides a scope for Names.

```
apiVersion: v1
import "k8s.io/api/core/v1"
```

Namespace

Namespace provides a scope for Names. Use of multiple namespaces is optional.

- **apiVersion**: v1
- **kind**: Namespace
- **metadata** ([ObjectMeta](#))

Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **spec** ([NamespaceSpec](#))

Spec defines the behavior of the Namespace. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status>

- **status** ([NamespaceStatus](#))

Status describes the current status of a Namespace. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status>

NamespaceSpec

NamespaceSpec describes the attributes on a Namespace.

- **finalizers** ([]string)

Finalizers is an opaque list of values that must be empty to permanently remove object from storage. More info: <https://kubernetes.io/docs/tasks/administer-cluster/namespaces/>

NamespaceStatus

NamespaceStatus is information about the current status of a Namespace.

- **conditions** ([]NamespaceCondition)

Patch strategy: merge on key type

Represents the latest available observations of a namespace's current state.

NamespaceCondition contains details about state of namespace.

- **conditions.status** (string), required

Status of the condition, one of True, False, Unknown.

- **conditions.type** (string), required
 - Type of namespace controller condition.
- **conditions.lastTransitionTime** (Time)
Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.
 - **conditions.message** (string)
 - **conditions.reason** (string)
- **phase** (string)

Phase is the current lifecycle phase of the namespace. More info:
<https://kubernetes.io/docs/tasks/administer-cluster/namespaces/>

NamespaceList

NamespaceList is a list of Namespaces.

-
- **apiVersion**: v1
 - **kind**: NamespaceList
 - **metadata** ([ListMeta](#))
Standard list metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds>
 - **items** ([\[\]Namespace](#)), required
Items is the list of Namespace objects in the list. More info: <https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/>

Operations

get read the specified Namespace

HTTP Request

GET /api/v1/namespaces/{name}

Parameters

- **name** (*in path*): string, required
 - name of the Namespace
- **pretty** (*in query*): string
 - [pretty](#)

Response

200 ([Namespace](#)): OK

401: Unauthorized

get read status of the specified Namespace

HTTP Request

GET /api/v1/namespaces/{name}/status

Parameters

- **name** (*in path*): string, required

name of the Namespace

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Namespace](#)): OK

401: Unauthorized

list list or watch objects of kind Namespace

HTTP Request

GET /api/v1/namespaces

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([NamespaceList](#)): OK

401: Unauthorized

create create a Namespace

HTTP Request

POST /api/v1/namespaces

Parameters

- **body**: [Namespace](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Namespace](#)): OK

201 ([Namespace](#)): Created

202 ([Namespace](#)): Accepted

401: Unauthorized

update replace the specified Namespace

HTTP Request

PUT /api/v1/namespaces/{name}

Parameters

- **name** (*in path*): string, required

name of the Namespace

- **body**: [Namespace](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Namespace](#)): OK

201 ([Namespace](#)): Created

401: Unauthorized

update replace finalize of the specified Namespace

HTTP Request

PUT /api/v1/namespaces/{name}/finalize

Parameters

- **name** (*in path*): string, required

name of the Namespace

- **body**: [Namespace](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Namespace](#)): OK

201 ([Namespace](#)): Created

401: Unauthorized

update replace status of the specified Namespace

HTTP Request

PUT /api/v1/namespaces/{name}/status

Parameters

- **name** (*in path*): string, required

name of the Namespace

- **body**: [Namespace](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Namespace](#)): OK

201 ([Namespace](#)): Created

401: Unauthorized

patch partially update the specified Namespace

HTTP Request

PATCH /api/v1/namespaces/{name}

Parameters

- **name** (*in path*): string, required

name of the Namespace

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Namespace](#)): OK

201 ([Namespace](#)): Created

401: Unauthorized

patch partially update status of the specified Namespace

HTTP Request

PATCH /api/v1/namespaces/{name}/status

Parameters

- **name** (*in path*): string, required

name of the Namespace

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Namespace](#)): OK

201 ([Namespace](#)): Created

401: Unauthorized

delete delete a Namespace

HTTP Request

DELETE /api/v1/namespaces/{name}

Parameters

- **name** (*in path*): string, required

name of the Namespace

- **body**: [DeleteOptions](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)

Response

200 ([Status](#)): OK

202 ([Status](#)): Accepted

401: Unauthorized

5.8.3 - Event

Event is a report of an event somewhere in the cluster.

```
apiVersion: events.k8s.io/v1

import "k8s.io/api/events/v1"
```

Event

Event is a report of an event somewhere in the cluster. It generally denotes some state change in the system. Events have a limited retention time and triggers and messages may evolve with time. Event consumers should not rely on the timing of an event with a given Reason reflecting a consistent underlying trigger, or the continued existence of events with that Reason. Events should be treated as informative, best-effort, supplemental data.

- **apiVersion**: events.k8s.io/v1

- **kind**: Event

- **metadata** ([ObjectMeta](#))

Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **eventTime** (MicroTime), required

eventTime is the time when this Event was first observed. It is required.

MicroTime is version of Time with microsecond level precision.

- **action** (string)

action is what action was taken/failed regarding to the regarding object. It is machine-readable. This field cannot be empty for new Events and it can have at most 128 characters.

- **deprecatedCount** (int32)

deprecatedCount is the deprecated field assuring backward compatibility with core.v1 Event type.

- **deprecatedFirstTimestamp** (Time)

deprecatedFirstTimestamp is the deprecated field assuring backward compatibility with core.v1 Event type.

Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

- **deprecatedLastTimestamp** (Time)

deprecatedLastTimestamp is the deprecated field assuring backward compatibility with core.v1 Event type.

Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

- **deprecatedSource** (EventSource)

deprecatedSource is the deprecated field assuring backward compatibility with core.v1 Event type.

EventSource contains information for an event.

- **deprecatedSource.component** (string)

Component from which the event is generated.

- **deprecatedSource.host** (string)

Node name on which the event is generated.

- **note** (string)

note is a human-readable description of the status of this operation. Maximal length of the note is 1kB, but libraries should be prepared to handle values up to 64kB.

- **reason** (string)

reason is why the action was taken. It is human-readable. This field cannot be empty for new Events and it can have at most 128 characters.

- **regarding** ([ObjectReference](#))

regarding contains the object this Event is about. In most cases it's an Object reporting controller implements, e.g. ReplicaSetController implements ReplicaSets and this event is emitted because it acts on some changes in a ReplicaSet object.

- **related** ([ObjectReference](#))

related is the optional secondary object for more complex actions. E.g. when regarding object triggers a creation or deletion of related object.

- **reportingController** (string)

reportingController is the name of the controller that emitted this Event, e.g. `kubernetes.io/kubelet`. This field cannot be empty for new Events.

- **reportingInstance** (string)

reportingInstance is the ID of the controller instance, e.g. `kubelet-xyzf`. This field cannot be empty for new Events and it can have at most 128 characters.

- **series** (EventSeries)

series is data about the Event series this event represents or nil if it's a singleton Event.

EventSeries contain information on series of events, i.e. thing that was/is happening continuously for some time. How often to update the EventSeries is up to the event reporters. The default event reporter in "k8s.io/client-go/tools/events/event_broadcaster.go" shows how this struct is updated on heartbeats and can guide customized reporter implementations.

- **series.count** (int32), required

count is the number of occurrences in this series up to the last heartbeat time.

- **series.lastObservedTime** (MicroTime), required

lastObservedTime is the time when last Event from the series

was seen before last heartbeat.

MicroTime is version of Time with microsecond level precision.

- **type** (string)

type is the type of this event (Normal, Warning), new types could be added in the future. It is machine-readable. This field cannot be empty for new Events.

EventList

EventList is a list of Event objects.

- **apiVersion**: events.k8s.io/v1

- **kind**: EventList

- **metadata** ([ListMeta](#))

Standard list metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **items** ([\[\]Event](#)), required

items is a list of schema objects.

Operations

get read the specified Event

HTTP Request

GET /apis/events.k8s.io/v1/namespaces/{namespace}/events/{name}

Parameters

- **name** (*in path*): string, required

name of the Event

- **namespace** (*in path*): string, required

[namespace](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Event](#)): OK

401: Unauthorized

list list or watch objects of kind Event

HTTP Request

GET /apis/events.k8s.io/v1/namespaces/{namespace}/events

Parameters

- **namespace** (*in path*): string, required
[namespace](#)
- **allowWatchBookmarks** (*in query*): boolean
[allowWatchBookmarks](#)
- **continue** (*in query*): string
[continue](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)
- **watch** (*in query*): boolean
[watch](#)

Response

200 ([EventList](#)): OK

401: Unauthorized

list list or watch objects of kind Event

HTTP Request

GET /apis/events.k8s.io/v1/events

Parameters

- **allowWatchBookmarks** (*in query*): boolean
[allowWatchBookmarks](#)
- **continue** (*in query*): string
[continue](#)

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([EventList](#)): OK

401: Unauthorized

create create an Event

HTTP Request

POST /apis/events.k8s.io/v1/namespaces/{namespace}/events

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [Event](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Event](#)): OK

201 ([Event](#)): Created

202 ([Event](#)): Accepted

401: Unauthorized

update replace the specified Event

HTTP Request

PUT /apis/events.k8s.io/v1/namespaces/{namespace}/events/{name}

Parameters

- **name** (*in path*): string, required

name of the Event

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [Event](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Event](#)): OK

201 ([Event](#)): Created

401: Unauthorized

patch partially update the specified Event

HTTP Request

PATCH /apis/events.k8s.io/v1/namespaces/{namespace}/events/{name}

Parameters

- **name** (*in path*): string, required

name of the Event

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Event](#)): OK

201 ([Event](#)): Created

401: Unauthorized

delete delete an Event

HTTP Request

DELETE /apis/events.k8s.io/v1/namespaces/{namespace}/events/{name}

Parameters

- **name** (*in path*): string, required

name of the Event

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [DeleteOptions](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

Response

200 ([Status](#)): OK

202 ([Status](#)): Accepted

401: Unauthorized

deletecollection delete collection of Event

HTTP Request

DELETE /apis/events.k8s.io/v1/namespaces/{namespace}/events

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [DeleteOptions](#)

- **continue** (*in query*): string

[continue](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.8.4 - APIService

APIService represents a server for a particular GroupVersion.

```
apiVersion: apiregistration.k8s.io/v1

import "k8s.io/kube-aggregator/pkg/apis/apiregistration/v1"
```

APIService

APIService represents a server for a particular GroupVersion. Name must be "version.group".

- **apiVersion**: apiregistration.k8s.io/v1

- **kind**: APIService

- **metadata** ([ObjectMeta](#))

Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **spec** ([APIServiceSpec](#))

Spec contains information for locating and communicating with a server

- **status** ([APIServiceStatus](#))

Status contains derived information about an API server

APIServiceSpec

APIServiceSpec contains information for locating and communicating with a server. Only https is supported, though you are able to disable certificate verification.

- **groupPriorityMinimum** (int32), required

GroupPriorityMinimum is the priority this group should have at least. Higher priority means that the group is preferred by clients over lower priority ones. Note that other versions of this group might specify even higher GroupPriorityMinimum values such that the whole group gets a higher priority. The primary sort is based on GroupPriorityMinimum, ordered highest number to lowest (20 before 10). The secondary sort is based on the alphabetical comparison of the name of the object. (v1.bar before v1.foo) We'd recommend something like: *.k8s.io (except extensions) at 18000 and PaaSes (OpenShift, Deis) are recommended to be in the 2000s

- **versionPriority** (int32), required

VersionPriority controls the ordering of this API version inside of its group. Must be greater than zero. The primary sort is based on VersionPriority, ordered highest to lowest (20 before 10). Since it's inside of a group, the number can be small, probably in the 10s. In case of equal version priorities, the version string will be used to compute the order inside a group. If the version string is "kube-like", it will sort above non "kube-like" version strings, which are ordered lexicographically. "Kube-like" versions start with a "v", then are

followed by a number (the major version), then optionally the string "alpha" or "beta" and another number (the minor version). These are sorted first by GA > beta > alpha (where GA is a version with no suffix such as beta or alpha), and then by comparing major version, then minor version. An example sorted list of versions: v10, v2, v1, v11beta2, v10beta3, v3beta1, v12alpha1, v11alpha2, foo1, foo10.

- **caBundle** ([]byte)

Atomic: will be replaced during a merge

CA Bundle is a PEM encoded CA bundle which will be used to validate an API server's serving certificate. If unspecified, system trust roots on the apiserver are used.

- **group** (string)

Group is the API group name this server hosts

- **insecureSkipTLSVerify** (boolean)

InsecureSkipTLSVerify disables TLS certificate verification when communicating with this server. This is strongly discouraged. You should use the CABundle instead.

- **service** (ServiceReference)

Service is a reference to the service for this API server. It must communicate on port 443. If the Service is nil, that means the handling for the API groupversion is handled locally on this server. The call will simply delegate to the normal handler chain to be fulfilled.

ServiceReference holds a reference to Service.legacy.k8s.io

- **service.name** (string)

Name is the name of the service

- **service.namespace** (string)

Namespace is the namespace of the service

- **service.port** (int32)

If specified, the port on the service that hosting webhook.

Default to 443 for backward compatibility. `port` should be a valid port number (1-65535, inclusive).

- **version** (string)

Version is the API version this server hosts. For example, "v1"

APIServiceStatus

APIServiceStatus contains derived information about an API server

- **conditions** ([]APIServiceCondition)

Patch strategy: merge on key type

Map: unique values on key type will be kept during a merge

Current service state of apiService.

APIServiceCondition describes the state of an APIService at a particular point

- **conditions.status** (string), required

Status is the status of the condition. Can be True, False, Unknown.

- **conditions.type** (string), required

Type is the type of the condition.

- **conditions.lastTransitionTime** (Time)

Last time the condition transitioned from one status to another.

Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

- **conditions.message** (string)

Human-readable message indicating details about last transition.

- **conditions.reason** (string)

Unique, one-word, CamelCase reason for the condition's last transition.

APIServiceList

APIServiceList is a list of APIService objects.

- **apiVersion**: apiregistration.k8s.io/v1
- **kind**: APIServiceList
- **metadata** ([ListMeta](#))

Standard list metadata More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **items** ([][APIService](#)), required

Items is the list of APIService

Operations

get read the specified APIService

HTTP Request

GET /apis/apiregistration.k8s.io/v1/apiservices/{name}

Parameters

- **name** (*in path*): string, required

name of the APIService

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([APIService](#)): OK

401: Unauthorized

get read status of the specified APIService

HTTP Request

GET /apis/apiregistration.k8s.io/v1/apiservices/{name}/status

Parameters

- **name** (*in path*): string, required

name of the APIService

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([APIService](#)): OK

401: Unauthorized

list list or watch objects of kind APIService

HTTP Request

GET /apis/apiregistration.k8s.io/v1/apiservices

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([APIServiceList](#)): OK

401: Unauthorized

create create an APIService

HTTP Request

POST /apis/apiregistration.k8s.io/v1/apiservices

Parameters

- **body**: [APIService](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([APIService](#)): OK

201 ([APIService](#)): Created

202 ([APIService](#)): Accepted

401: Unauthorized

update replace the specified APIService

HTTP Request

PUT /apis/apiregistration.k8s.io/v1/apiservices/{name}

Parameters

- **name** (*in path*): string, required

name of the APIService

- **body**: [APIService](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([APIService](#)): OK

201 ([APIService](#)): Created

401: Unauthorized

update replace status of the specified APIService

HTTP Request

PUT /apis/apiregistration.k8s.io/v1/apiservices/{name}/status

Parameters

- **name** (*in path*): string, required
name of the APIService
- **body**: [APIService](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([APIService](#)): OK

201 ([APIService](#)): Created

401: Unauthorized

patch partially update the specified APIService

HTTP Request

PATCH /apis/apiregistration.k8s.io/v1/apiservices/{name}

Parameters

- **name** (*in path*): string, required

name of the APIService

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([APIService](#)): OK

201 ([APIService](#)): Created

401: Unauthorized

patch partially update status of the specified APIService

HTTP Request

PATCH /apis/apiregistration.k8s.io/v1/apiservices/{name}/status

Parameters

- **name** (*in path*): string, required

name of the APIService

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([APIService](#)): OK

201 ([APIService](#)): Created

401: Unauthorized

delete delete an APIService

HTTP Request

DELETE /apis/apiregistration.k8s.io/v1/apiservices/{name}

Parameters

- **name** (*in path*): string, required

name of the APIService

- **body**: [DeleteOptions](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

Response

200 ([Status](#)): OK

202 ([Status](#)): Accepted

401: Unauthorized

deletecollection delete collection of APIService

HTTP Request

DELETE /apis/apiregistration.k8s.io/v1/apiservices

Parameters

- **body**: [DeleteOptions](#)

- **continue** (*in query*): string

[continue](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.8.5 - Lease

Lease defines a lease concept.

```
apiVersion: coordination.k8s.io/v1
import "k8s.io/api/coordination/v1"
```

Lease

Lease defines a lease concept.

- **apiVersion**: coordination.k8s.io/v1

- **kind**: Lease

- **metadata** ([ObjectMeta](#))

More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **spec** ([LeaseSpec](#))

spec contains the specification of the Lease. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status>

LeaseSpec

LeaseSpec is a specification of a Lease.

- **acquireTime** (MicroTime)

acquireTime is a time when the current lease was acquired.

MicroTime is version of Time with microsecond level precision.

- **holderIdentity** (string)

holderIdentity contains the identity of the holder of a current lease.

- **leaseDurationSeconds** (int32)

leaseDurationSeconds is a duration that candidates for a lease need to wait to force acquire it. This is measure against time of last observed renewTime.

- **leaseTransitions** (int32)

leaseTransitions is the number of transitions of a lease between holders.

- **renewTime** (MicroTime)

renewTime is a time when the current holder of a lease has last updated the lease.

MicroTime is version of Time with microsecond level precision.

LeaseList

LeaseList is a list of Lease objects.

- **apiVersion**: coordination.k8s.io/v1
- **kind**: LeaseList
- **metadata** ([ListMeta](#))

Standard list metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **items** ([\[\]Lease](#)), required

items is a list of schema objects.

Operations

get read the specified Lease

HTTP Request

GET /apis/coordination.k8s.io/v1/namespaces/{namespace}/leases/{name}

Parameters

- **name** (*in path*): string, required
name of the Lease
- **namespace** (*in path*): string, required
[namespace](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([Lease](#)): OK

401: Unauthorized

list list or watch objects of kind Lease

HTTP Request

GET /apis/coordination.k8s.io/v1/namespaces/{namespace}/leases

Parameters

- **namespace** (*in path*): string, required
[namespace](#)
- **allowWatchBookmarks** (*in query*): boolean
[allowWatchBookmarks](#)
- **continue** (*in query*): string
[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([LeaseList](#)): OK

401: Unauthorized

list list or watch objects of kind Lease

HTTP Request

GET /apis/coordination.k8s.io/v1/leases

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string
[pretty](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)
- **watch** (*in query*): boolean
[watch](#)

Response

200 ([LeaseList](#)): OK

401: Unauthorized

create create a Lease

HTTP Request

POST /apis/coordination.k8s.io/v1/namespaces/{namespace}/leases

Parameters

- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [Lease](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([Lease](#)): OK

201 ([Lease](#)): Created

202 ([Lease](#)): Accepted

401: Unauthorized

update replace the specified Lease

HTTP Request

PUT /apis/coordination.k8s.io/v1/namespaces/{namespace}/leases/{name}

Parameters

- **name** (*in path*): string, required
name of the Lease
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [Lease](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([Lease](#)): OK

201 ([Lease](#)): Created

401: Unauthorized

patch partially update the specified Lease

HTTP Request

PATCH /apis/coordination.k8s.io/v1/namespaces/{namespace}/leases/{name}

Parameters

- **name** (*in path*): string, required
name of the Lease
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [Patch](#), required
- **dryRun** (*in query*): string
[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Lease](#)): OK

201 ([Lease](#)): Created

401: Unauthorized

delete delete a Lease

HTTP Request

DELETE /apis/coordination.k8s.io/v1/namespaces/{namespace}/leases/{name}

Parameters

- **name** (*in path*): string, required

name of the Lease

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [DeleteOptions](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

Response

200 ([Status](#)): OK

202 ([Status](#)): Accepted

401: Unauthorized

deletecollection delete collection of Lease

HTTP Request

DELETE /apis/coordination.k8s.io/v1/namespaces/{namespace}/leases

Parameters

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [DeleteOptions](#)

- **continue** (*in query*): string

[continue](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.8.6 - LeaseCandidate

This is a placeholder page to be overwritten when the docs are generated for 1.31

5.8.7 - RuntimeClass

RuntimeClass defines a class of container runtime supported in the cluster.

```
apiVersion: node.k8s.io/v1

import "k8s.io/api/node/v1"
```

RuntimeClass

RuntimeClass defines a class of container runtime supported in the cluster. The RuntimeClass is used to determine which container runtime is used to run all containers in a pod. RuntimeClasses are manually defined by a user or cluster provisioner, and referenced in the PodSpec. The Kubelet is responsible for resolving the RuntimeClassName reference before running the pod. For more details, see <https://kubernetes.io/docs/concepts/containers/runtime-class/>

- **apiVersion**: node.k8s.io/v1
- **kind**: RuntimeClass
- **metadata** ([ObjectMeta](#))

More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **handler** (string), required

handler specifies the underlying runtime and configuration that the CRI implementation will use to handle pods of this class. The possible values are specific to the node & CRI configuration. It is assumed that all handlers are available on every node, and handlers of the same name are equivalent on every node. For example, a handler called "runc" might specify that the runc OCI runtime (using native Linux containers) will be used to run the containers in a pod. The Handler must be lowercase, conform to the DNS Label (RFC 1123) requirements, and is immutable.

- **overhead** (Overhead)

overhead represents the resource overhead associated with running a pod for a given RuntimeClass. For more details, see <https://kubernetes.io/docs/concepts/scheduling-eviction/pod-overhead/>

Overhead structure represents the resource overhead associated with running a pod.

- **overhead.podFixed** ([map\[string\]Quantity](#))

podFixed represents the fixed resource overhead associated with running a pod.

- **scheduling** (Scheduling)

scheduling holds the scheduling constraints to ensure that pods running with this RuntimeClass are scheduled to nodes that support it. If scheduling is nil, this RuntimeClass is assumed to be supported by all nodes.

Scheduling specifies the scheduling constraints for nodes supporting a RuntimeClass.

- **scheduling.nodeSelector** (map[string]string)

nodeSelector lists labels that must be present on nodes that support this RuntimeClass. Pods using this RuntimeClass can only be scheduled to a node matched by this selector. The RuntimeClass nodeSelector is merged with a pod's existing nodeSelector. Any conflicts will cause the pod to be rejected in admission.

- **scheduling.tolerations** ([]Toleration)

Atomic: will be replaced during a merge

tolerations are appended (excluding duplicates) to pods running with this RuntimeClass during admission, effectively unioning the set of nodes tolerated by the pod and the RuntimeClass.

The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator

.

- **scheduling.tolerations.key** (string)

Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is empty, operator must be Exists; this combination means to match all values and all keys.

- **scheduling.tolerations.operator** (string)

Operator represents a key's relationship to the value. Valid operators are Exists and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category.

- **scheduling.tolerations.value** (string)

Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

- **scheduling.tolerations.effect** (string)

Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.

- **scheduling.tolerations.tolerationSeconds** (int64)

TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.

Runtimelist

Runtimelist is a list of Runtimelist objects.

- **apiVersion**: node.k8s.io/v1

- **kind**: Runtimelist

- **metadata** ([ListMeta](#))

Standard list metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **items** ([][RuntimeClass](#)), required

items is a list of schema objects.

Operations

get read the specified RuntimeClass

HTTP Request

GET /apis/node.k8s.io/v1/runtimeclasses/{name}

Parameters

- **name** (*in path*): string, required

name of the RuntimeClass

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([RuntimeClass](#)): OK

401: Unauthorized

list list or watch objects of kind RuntimeClass

HTTP Request

GET /apis/node.k8s.io/v1/runtimeclasses

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)
- **watch** (*in query*): boolean
[watch](#)

Response

200 ([RuntimeClassList](#)): OK

401: Unauthorized

create create a RuntimeClass

HTTP Request

POST /apis/node.k8s.io/v1/runtimeclasses

Parameters

- **body**: [RuntimeClass](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([RuntimeClass](#)): OK

201 ([RuntimeClass](#)): Created

202 ([RuntimeClass](#)): Accepted

401: Unauthorized

update replace the specified RuntimeClass

HTTP Request

PUT /apis/node.k8s.io/v1/runtimeclasses/{name}

Parameters

- **name** (*in path*): string, required

name of the RuntimeClass

- **body**: [RuntimeClass](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([RuntimeClass](#)): OK

201 ([RuntimeClass](#)): Created

401: Unauthorized

patch partially update the specified RuntimeClass

HTTP Request

PATCH /apis/node.k8s.io/v1/runtimeclasses/{name}

Parameters

- **name** (*in path*): string, required

name of the RuntimeClass

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([RuntimeClass](#)): OK

201 ([RuntimeClass](#)): Created

401: Unauthorized

delete delete a RuntimeClass

HTTP Request

DELETE /apis/node.k8s.io/v1/runtimeclasses/{name}

Parameters

- **name** (*in path*): string, required

name of the RuntimeClass

- **body**: [DeleteOptions](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

Response

200 ([Status](#)): OK

202 ([Status](#)): Accepted

401: Unauthorized

deletecollection delete collection of RuntimeClass

HTTP Request

DELETE /apis/node.k8s.io/v1/runtimeclasses

Parameters

- **body**: [DeleteOptions](#)

- **continue** (*in query*): string

[continue](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.8.8 - FlowSchema v1beta3

FlowSchema defines the schema of a group of flows.

```
apiVersion: flowcontrol.apiserver.k8s.io/v1beta3

import "k8s.io/api/flowcontrol/v1beta3"
```

FlowSchema

FlowSchema defines the schema of a group of flows. Note that a flow is made up of a set of inbound API requests with similar attributes and is identified by a pair of strings: the name of the FlowSchema and a "flow distinguisher".

- **apiVersion**: flowcontrol.apiserver.k8s.io/v1beta3
- **kind**: FlowSchema
- **metadata** ([ObjectMeta](#))

`metadata` is the standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **spec** ([FlowSchemaSpec](#))

`spec` is the specification of the desired behavior of a FlowSchema. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status>

- **status** ([FlowSchemaStatus](#))

`status` is the current status of a FlowSchema. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status>

FlowSchemaSpec

FlowSchemaSpec describes how the FlowSchema's specification looks like.

- **priorityLevelConfiguration** ([PriorityLevelConfigurationReference](#)), required

`priorityLevelConfiguration` should reference a PriorityLevelConfiguration in the cluster. If the reference cannot be resolved, the FlowSchema will be ignored and marked as invalid in its status. Required.

PriorityLevelConfigurationReference contains information that points to the "request-priority" being used.

- **priorityLevelConfiguration.name** (string), required

`name` is the name of the priority level configuration being referenced Required.

- **distinguisherMethod** ([FlowDistinguisherMethod](#))

`distinguisherMethod` defines how to compute the flow distinguisher for requests that match this schema. `nil` specifies that the

distinguisher is disabled and thus will always be the empty string.

FlowDistinguisherMethod specifies the method of a flow distinguisher.

- **distinguisherMethod.type** (string), required

`type` is the type of flow distinguisher method. The supported types are "ByUser" and "ByNamespace". Required.

- **matchingPrecedence** (int32)

`matchingPrecedence` is used to choose among the FlowSchemas that match a given request. The chosen FlowSchema is among those with the numerically lowest (which we take to be logically highest) MatchingPrecedence. Each MatchingPrecedence value must be ranged in [1,10000]. Note that if the precedence is not specified, it will be set to 1000 as default.

- **rules** ([]PolicyRulesWithSubjects)

Atomic: will be replaced during a merge

`rules` describes which requests will match this flow schema. This FlowSchema matches a request if and only if at least one member of `rules` matches the request. If it is an empty slice, there will be no requests matching the FlowSchema.

PolicyRulesWithSubjects prescribes a test that applies to a request to an apiserver. The test considers the subject making the request, the verb being requested, and the resource to be acted upon. This PolicyRulesWithSubjects matches a request if and only if both (a) at least one member of `subjects` matches the request and (b) at least one member of `resourceRules` or `nonResourceRules` matches the request.

- **rules.subjects** ([]Subject), required

Atomic: will be replaced during a merge

`subjects` is the list of normal user, serviceaccount, or group that this rule cares about. There must be at least one member in this slice. A slice that includes both the `system:authenticated` and `system:unauthenticated` user groups matches every request. Required.

Subject matches the originator of a request, as identified by the request authentication system. There are three ways of matching an originator; by user, group, or service account.

- **rules.subjects.kind** (string), required

`kind` indicates which one of the other fields is non-empty. Required

- **rules.subjects.group** (GroupSubject)

`group` matches based on user group name.

GroupSubject holds detailed information for group-kind subject.

- **rules.subjects.group.name** (string), required

`name` is the user group that matches, or "*" to match all user groups. See <https://github.com/kubernetes/apiserver/blob/master/pkg/authentication/user/user.go> for some well-known group names. Required.

- **rules.subjects.serviceAccount** (ServiceAccountSubject)

`serviceAccount` matches ServiceAccounts.

ServiceAccountSubject holds detailed information for service-account-kind subject.

- **rules.subjects.serviceAccount.name** (string), required

`name` is the name of matching ServiceAccount objects, or "*" to match regardless of name. Required.

- **rules.subjects.serviceAccount.namespace** (string), required

`namespace` is the namespace of matching ServiceAccount objects. Required.

- **rules.subjects.user** (UserSubject)

`user` matches based on username.

UserSubject holds detailed information for user-kind subject.

- **rules.subjects.user.name** (string), required

`name` is the username that matches, or "*" to match all usernames. Required.

- **rules.nonResourceRules** ([]NonResourcePolicyRule)

Atomic: will be replaced during a merge

`nonResourceRules` is a list of NonResourcePolicyRules that identify matching requests according to their verb and the target non-resource URL.

NonResourcePolicyRule is a predicate that matches non-resource requests according to their verb and the target non-resource URL. A NonResourcePolicyRule matches a request if and only if both (a) at least one member of verbs matches the request and (b) at least one member of nonResourceURLs matches the request.

- **rules.nonResourceRules.nonResourceURLs** ([]string), required

Set: unique values will be kept during a merge

`nonResourceURLs` is a set of url prefixes that a user should have access to and may not be empty. For example:

- "/healthz" is legal
- "/hea*" is illegal
- "/hea" is legal but matches nothing
- "/hea/*" also matches nothing
- "/healthz/" matches all per-component health checks. "" matches all non-resource urls. if it is present, it must be the only entry. Required.

- **rules.nonResourceRules.verbs** ([]string), required

Set: unique values will be kept during a merge

`verbs` is a list of matching verbs and may not be empty. "*" matches all verbs. If it is present, it must be the only entry. Required.

- **rules.resourceRules** ([]ResourcePolicyRule)

Atomic: will be replaced during a merge

`resourceRules` is a slice of `ResourcePolicyRules` that identify matching requests according to their verb and the target resource. At least one of `resourceRules` and `nonResourceRules` has to be non-empty.

ResourcePolicyRule is a predicate that matches some resource requests, testing the request's verb and the target resource. A ResourcePolicyRule matches a resource request if and only if: (a) at least one member of verbs matches the request, (b) at least one member of apiGroups matches the request, (c) at least one member of resources matches the request, and (d) either (d1) the request does not specify a namespace (i.e., `Namespace==""`) and clusterScope is true or (d2) the request specifies a namespace and least one member of namespaces matches the request's namespace.

- **rules.resourceRules.apiGroups** (`[]string`), required

Set: unique values will be kept during a merge

`apiGroups` is a list of matching API groups and may not be empty. "*" matches all API groups and, if present, must be the only entry. Required.

- **rules.resourceRules.resources** (`[]string`), required

Set: unique values will be kept during a merge

`resources` is a list of matching resources (i.e., lowercase and plural) with, if desired, subresource. For example, ["services", "nodes/status"]. This list may not be empty. "*" matches all resources and, if present, must be the only entry. Required.

- **rules.resourceRules.verbs** (`[]string`), required

Set: unique values will be kept during a merge

`verbs` is a list of matching verbs and may not be empty. "*" matches all verbs and, if present, must be the only entry. Required.

- **rules.resourceRules.clusterScope** (boolean)

`clusterScope` indicates whether to match requests that do not specify a namespace (which happens either because the resource is not namespaced or the request targets all namespaces). If this field is omitted or false then the `namespaces` field must contain a non-empty list.

- **rules.resourceRules.namespaces** (`[]string`)

Set: unique values will be kept during a merge

`namespaces` is a list of target namespaces that restricts matches. A request that specifies a target namespace matches only if either (a) this list contains that target namespace or (b) this list contains "". Note that "" matches any specified namespace but does not match a request that *does not specify* a namespace (see the `clusterScope` field for that). This list may be empty, but only if `clusterScope` is true.

FlowSchemaStatus

FlowSchemaStatus represents the current state of a FlowSchema.

- **conditions** ([]FlowSchemaCondition)

Patch strategy: merge on key type

Map: unique values on key type will be kept during a merge

`conditions` is a list of the current states of FlowSchema.

FlowSchemaCondition describes conditions for a FlowSchema.

- **conditions.lastTransitionTime** (Time)

`lastTransitionTime` is the last time the condition transitioned from one status to another.

Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

- **conditions.message** (string)

`message` is a human-readable message indicating details about last transition.

- **conditions.reason** (string)

`reason` is a unique, one-word, CamelCase reason for the condition's last transition.

- **conditions.status** (string)

`status` is the status of the condition. Can be True, False, Unknown. Required.

- **conditions.type** (string)

`type` is the type of the condition. Required.

FlowSchemaList

FlowSchemaList is a list of FlowSchema objects.

- **apiVersion**: flowcontrol.apiserver.k8s.io/v1beta3
- **kind**: FlowSchemaList
- **metadata** ([ListMeta](#))

`metadata` is the standard list metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **items** ([]FlowSchema), required

`items` is a list of FlowSchemas.

Operations

get read the specified FlowSchema

HTTP Request

GET /apis/flowcontrol.apiserver.k8s.io/v1beta3/flowschemas/{name}

Parameters

- **name** (*in path*): string, required

name of the FlowSchema

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([FlowSchema](#)): OK

401: Unauthorized

get read status of the specified FlowSchema

HTTP Request

GET /apis/flowcontrol.apiserver.k8s.io/v1beta3/flowschemas/{name}/status

Parameters

- **name** (*in path*): string, required

name of the FlowSchema

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([FlowSchema](#)): OK

401: Unauthorized

list list or watch objects of kind FlowSchema

HTTP Request

GET /apis/flowcontrol.apiserver.k8s.io/v1beta3/flowschemas

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)
- **watch** (*in query*): boolean
[watch](#)

Response

200 ([FlowSchemaList](#)): OK

401: Unauthorized

create create a FlowSchema

HTTP Request

POST /apis/flowcontrol.apiserver.k8s.io/v1beta3/flowschemas

Parameters

- **body**: [FlowSchema](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([FlowSchema](#)): OK

201 ([FlowSchema](#)): Created

202 ([FlowSchema](#)): Accepted

401: Unauthorized

update replace the specified FlowSchema

HTTP Request

PUT /apis/flowcontrol.apiserver.k8s.io/v1beta3/flowschemas/{name}

Parameters

- **name** (*in path*): string, required

name of the FlowSchema

- **body**: [FlowSchema](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([FlowSchema](#)): OK

201 ([FlowSchema](#)): Created

401: Unauthorized

update replace status of the specified FlowSchema

HTTP Request

PUT /apis/flowcontrol.apiserver.k8s.io/v1beta3/flowschemas/{name}/status

Parameters

- **name** (*in path*): string, required

name of the FlowSchema

- **body**: [FlowSchema](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([FlowSchema](#)): OK

201 ([FlowSchema](#)): Created

401: Unauthorized

patch partially update the specified FlowSchema

HTTP Request

PATCH /apis/flowcontrol.apiserver.k8s.io/v1beta3/flowschemas/{name}

Parameters

- **name** (*in path*): string, required

name of the FlowSchema

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([FlowSchema](#)): OK

201 ([FlowSchema](#)): Created

401: Unauthorized

patch partially update status of the specified FlowSchema

HTTP Request

PATCH /apis/flowcontrol.apiserver.k8s.io/v1beta3/flowschemas/{name}/status

Parameters

- **name** (*in path*): string, required

name of the FlowSchema

- **body**: [Patch](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **force** (*in query*): boolean
[force](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([FlowSchema](#)): OK

201 ([FlowSchema](#)): Created

401: Unauthorized

delete delete a FlowSchema

HTTP Request

DELETE /apis/flowcontrol.apiserver.k8s.io/v1beta3/flowschemas/{name}

Parameters

- **name** (*in path*): string, required
name of the FlowSchema
- **body**: [DeleteOptions](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)

Response

200 ([Status](#)): OK

202 ([Status](#)): Accepted

401: Unauthorized

deletecollection delete collection of FlowSchema

HTTP Request

DELETE /apis/flowcontrol.apiserver.k8s.io/v1beta3/flowschemas

Parameters

- **body**: [DeleteOptions](#)
- **continue** (*in query*): string
[continue](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **sendInitialEvents** (*in query*): boolean
[sendInitialEvents](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.8.9 - PriorityLevelConfiguration v1beta3

PriorityLevelConfiguration represents the configuration of a priority level.

```
apiVersion: flowcontrol.apiserver.k8s.io/v1beta3
import "k8s.io/api/flowcontrol/v1beta3"
```

PriorityLevelConfiguration

PriorityLevelConfiguration represents the configuration of a priority level.

- **apiVersion**: flowcontrol.apiserver.k8s.io/v1beta3

- **kind**: PriorityLevelConfiguration

- **metadata** ([ObjectMeta](#))

`metadata` is the standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **spec** ([PriorityLevelConfigurationSpec](#))

`spec` is the specification of the desired behavior of a "request-priority". More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status>

- **status** ([PriorityLevelConfigurationStatus](#))

`status` is the current status of a "request-priority". More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status>

PriorityLevelConfigurationSpec

PriorityLevelConfigurationSpec specifies the configuration of a priority level.

- **type** (string), required

`type` indicates whether this priority level is subject to limitation on request execution. A value of "Exempt" means that requests of this priority level are not subject to a limit (and thus are never queued) and do not detract from the capacity made available to other priority levels. A value of "Limited" means that (a) requests of this priority level *are* subject to limits and (b) some of the server's limited capacity is made available exclusively to this priority level. Required.

- **exempt** ([ExemptPriorityLevelConfiguration](#))

`exempt` specifies how requests are handled for an exempt priority level. This field MUST be empty if `type` is "Limited". This field MAY be non-empty if `type` is "Exempt". If empty and `type` is "Exempt" then the default values for `ExemptPriorityLevelConfiguration` apply.

ExemptPriorityLevelConfiguration describes the configurable aspects of the handling of exempt requests. In the mandatory `exempt` configuration object the values in the fields here can be modified by authorized users, unlike the rest of the `spec`.

- **exempt.lendablePercent** (int32)

`lendablePercent` prescribes the fraction of the level's NominalCL that can be borrowed by other priority levels. This value of this field must be between 0 and 100, inclusive, and it defaults to 0. The number of seats that other levels can borrow from this level, known as this level's LendableConcurrencyLimit (LendableCL), is defined as follows.

$$\text{LendableCL}(i) = \text{round}(\text{NominalCL}(i) * \text{lendablePercent}(i)/100.0)$$

- **exempt.nominalConcurrencyShares** (int32)

`nominalConcurrencyShares` (NCS) contributes to the computation of the NominalConcurrencyLimit (NominalCL) of this level. This is the number of execution seats nominally reserved for this priority level. This DOES NOT limit the dispatching from this priority level but affects the other priority levels through the borrowing mechanism. The server's concurrency limit (ServerCL) is divided among all the priority levels in proportion to their NCS values:

$$\text{NominalCL}(i) = \text{ceil}(\text{ServerCL} * \text{NCS}(i) / \text{sum_ncs})$$

`sum_ncs` = $\sum_{k \in \text{priority level } k} \text{NCS}(k)$

Bigger numbers mean a larger nominal concurrency limit, at the expense of every other priority level. This field has a default value of zero.

- **limited** (LimitedPriorityLevelConfiguration)

`limited` specifies how requests are handled for a Limited priority level. This field must be non-empty if and only if `type` is "Limited".

*LimitedPriorityLevelConfiguration specifies how to handle requests that are subject to limits. It addresses two issues:

- How are requests for this priority level limited?
- What should be done with requests that exceed the limit?*
- **limited.borrowingLimitPercent** (int32)

`borrowingLimitPercent`, if present, configures a limit on how many seats this priority level can borrow from other priority levels. The limit is known as this level's BorrowingConcurrencyLimit (BorrowingCL) and is a limit on the total number of seats that this level may borrow at any one time. This field holds the ratio of that limit to the level's nominal concurrency limit. When this field is non-nil, it must hold a non-negative integer and the limit is calculated as follows.

$$\text{BorrowingCL}(i) = \text{round}(\text{NominalCL}(i) * \text{borrowingLimitPercent}(i)/100.0)$$

The value of this field can be more than 100, implying that this priority level can borrow a number of seats that is greater than its own nominal concurrency limit (NominalCL). When this field is left `nil`, the limit is effectively infinite.

- o **limited.lendablePercent** (int32)

`lendablePercent` prescribes the fraction of the level's NominalCL that can be borrowed by other priority levels. The value of this field must be between 0 and 100, inclusive, and it defaults to 0. The number of seats that other levels can borrow from this level, known as this level's LendableConcurrencyLimit (LendableCL), is defined as follows.

```
LendableCL(i) = round( NominalCL(i) * lendablePercent(i)/100.0
)
```

- o **limited.limitResponse** (LimitResponse)

`limitResponse` indicates what to do with requests that can not be executed right now

LimitResponse defines how to handle requests that can not be executed right now.

- **limited.limitResponse.type** (string), required

`type` is "Queue" or "Reject". "Queue" means that requests that can not be executed upon arrival are held in a queue until they can be executed or a queuing limit is reached. "Reject" means that requests that can not be executed upon arrival are rejected. Required.

- **limited.limitResponse.queuing** (QueuingConfiguration)

`queuing` holds the configuration parameters for queuing. This field may be non-empty only if `type` is "Queue".

QueuingConfiguration holds the configuration parameters for queuing

- **limited.limitResponse.queuing.handSize** (int32)

`handSize` is a small positive number that configures the shuffle sharding of requests into queues. When enqueueing a request at this priority level the request's flow identifier (a string pair) is hashed and the hash value is used to shuffle the list of queues and deal a hand of the size specified here. The request is put into one of the shortest queues in that hand. `handSize` must be no larger than `queues`, and should be significantly smaller (so that a few heavy flows do not saturate most of the queues). See the user-facing documentation for more extensive guidance on setting this field. This field has a default value of 8.

- **limited.limitResponse.queuing.queueLengthLimit** (int32)

`queueLengthLimit` is the maximum number of requests allowed to be waiting in a given queue of this priority level at a time; excess requests are rejected. This value must be positive. If not specified, it will be defaulted to 50.

- **limited.limitResponse.queuing.queues** (int32)

`queues` is the number of queues for this priority level. The queues exist independently at each apiserver. The value must be positive. Setting it to 1 effectively precludes shufflesharding and thus

makes the distinguisher method of associated flow schemas irrelevant. This field has a default value of 64.

- **limited.nominalConcurrencyShares** (int32)

`nominalConcurrencyShares` (NCS) contributes to the computation of the NominalConcurrencyLimit (NominalCL) of this level. This is the number of execution seats available at this priority level. This is used both for requests dispatched from this priority level as well as requests dispatched from other priority levels borrowing seats from this level. The server's concurrency limit (ServerCL) is divided among the Limited priority levels in proportion to their NCS values:

$$\text{NominalCL}(i) = \text{ceil}(\text{ServerCL} * \text{NCS}(i) / \text{sum_ncs})$$
$$\text{sum_ncs} = \sum_{k} \text{NCS}(k)$$

Bigger numbers mean a larger nominal concurrency limit, at the expense of every other priority level. This field has a default value of 30.

PriorityLevelConfigurationStatus

`PriorityLevelConfigurationStatus` represents the current state of a "request-priority".

- **conditions** ([]PriorityLevelConfigurationCondition)

Patch strategy: merge on key type

Map: unique values on key type will be kept during a merge

`conditions` is the current state of "request-priority".

`PriorityLevelConfigurationCondition` defines the condition of priority level.

- **conditions.lastTransitionTime** (Time)

`lastTransitionTime` is the last time the condition transitioned from one status to another.

Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

- **conditions.message** (string)

`message` is a human-readable message indicating details about last transition.

- **conditions.reason** (string)

`reason` is a unique, one-word, CamelCase reason for the condition's last transition.

- **conditions.status** (string)

`status` is the status of the condition. Can be True, False, Unknown. Required.

- **conditions.type** (string)

`type` is the type of the condition. Required.

PriorityLevelConfigurationList

PriorityLevelConfigurationList is a list of PriorityLevelConfiguration objects.

- **apiVersion**: flowcontrol.apiserver.k8s.io/v1beta3
- **kind**: PriorityLevelConfigurationList
- **metadata** ([ListMeta](#))

`metadata` is the standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **items** (`[]PriorityLevelConfiguration`), required

`items` is a list of request-priorities.

Operations

get read the specified PriorityLevelConfiguration

HTTP Request

GET /apis/flowcontrol.apiserver.k8s.io/v1beta3/
prioritylevelconfigurations/{name}

Parameters

- **name** (*in path*): string, required
name of the PriorityLevelConfiguration
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([PriorityLevelConfiguration](#)): OK

401: Unauthorized

get read status of the specified PriorityLevelConfiguration

HTTP Request

GET /apis/flowcontrol.apiserver.k8s.io/v1beta3/
prioritylevelconfigurations/{name}/status

Parameters

- **name** (*in path*): string, required
name of the PriorityLevelConfiguration
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([PriorityLevelConfiguration](#)): OK

401: Unauthorized

list list or watch objects of kind PriorityLevelConfiguration

HTTP Request

GET /apis/flowcontrol.apiserver.k8s.io/v1beta3/
prioritylevelconfigurations

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([PriorityLevelConfigurationList](#)): OK

401: Unauthorized

create create a PriorityLevelConfiguration

HTTP Request

POST /apis/flowcontrol.apiserver.k8s.io/v1beta3/
prioritylevelconfigurations

Parameters

- **body**: [PriorityLevelConfiguration](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([PriorityLevelConfiguration](#)): OK

201 ([PriorityLevelConfiguration](#)): Created

202 ([PriorityLevelConfiguration](#)): Accepted

401: Unauthorized

update replace the specified PriorityLevelConfiguration

HTTP Request

PUT /apis/flowcontrol.apiserver.k8s.io/v1beta3/
prioritylevelconfigurations/{name}

Parameters

- **name** (*in path*): string, required

name of the PriorityLevelConfiguration

- **body**: [PriorityLevelConfiguration](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([PriorityLevelConfiguration](#)): OK

201 ([PriorityLevelConfiguration](#)): Created

401: Unauthorized

update replace status of the specified PriorityLevelConfiguration

HTTP Request

PUT /apis/flowcontrol.apiserver.k8s.io/v1beta3/prioritylevelconfigurations/{name}/status

Parameters

- **name** (*in path*): string, required

name of the PriorityLevelConfiguration

- **body**: [PriorityLevelConfiguration](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([PriorityLevelConfiguration](#)): OK

201 ([PriorityLevelConfiguration](#)): Created

401: Unauthorized

patch partially update the specified PriorityLevelConfiguration

HTTP Request

PATCH /apis/flowcontrol.apiserver.k8s.io/v1beta3/prioritylevelconfigurations/{name}

Parameters

- **name** (*in path*): string, required

name of the PriorityLevelConfiguration

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([PriorityLevelConfiguration](#)): OK

201 ([PriorityLevelConfiguration](#)): Created

401: Unauthorized

patch partially update status of the specified PriorityLevelConfiguration

HTTP Request

PATCH /apis/flowcontrol.apiserver.k8s.io/v1beta3/prioritylevelconfigurations/{name}/status

Parameters

- **name** (*in path*): string, required

name of the PriorityLevelConfiguration

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([PriorityLevelConfiguration](#)): OK

201 ([PriorityLevelConfiguration](#)): Created

401: Unauthorized

delete delete a PriorityLevelConfiguration

HTTP Request

DELETE /apis/flowcontrol.apiserver.k8s.io/v1beta3/
prioritylevelconfigurations/{name}

Parameters

- **name** (*in path*): string, required
name of the PriorityLevelConfiguration
- **body**: [DeleteOptions](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#)

Response

200 ([Status](#)): OK

202 ([Status](#)): Accepted

401: Unauthorized

deletecollection delete collection of PriorityLevelConfiguration

HTTP Request

DELETE /apis/flowcontrol.apiserver.k8s.io/v1beta3/
prioritylevelconfigurations

Parameters

- **body**: [DeleteOptions](#)
- **continue** (*in query*): string
[continue](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.8.10 - Binding

Binding ties one object to another; for example, a pod is bound to a node by a scheduler.

```
apiVersion: v1

import "k8s.io/api/core/v1"
```

Binding

Binding ties one object to another; for example, a pod is bound to a node by a scheduler. Deprecated in 1.7, please use the bindings subresource of pods instead.

- **apiVersion**: v1
- **kind**: Binding
- **metadata** ([ObjectMeta](#))

Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **target** ([ObjectReference](#)), required

The target object that you want to bind to the standard object.

Operations

create create a Binding

HTTP Request

POST /api/v1/namespaces/{namespace}/bindings

Parameters

- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [Binding](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([Binding](#)): OK

201 ([Binding](#)): Created

202 ([Binding](#)): Accepted

401: Unauthorized

create create binding of a Pod

HTTP Request

POST /api/v1/namespaces/{namespace}/pods/{name}/binding

Parameters

- **name** (*in path*): string, required

name of the Binding

- **namespace** (*in path*): string, required

[namespace](#)

- **body**: [Binding](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([Binding](#)): OK

201 ([Binding](#)): Created

202 ([Binding](#)): Accepted

401: Unauthorized

5.8.11 - ComponentStatus

ComponentStatus (and ComponentStatusList) holds the cluster validation info.

```
apiVersion: v1

import "k8s.io/api/core/v1"
```

ComponentStatus

ComponentStatus (and ComponentStatusList) holds the cluster validation info. Deprecated: This API is deprecated in v1.19+

- **apiVersion**: v1

- **kind**: ComponentStatus

- **metadata** ([ObjectMeta](#))

Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **conditions** ([]ComponentCondition)

Patch strategy: merge on key type

List of component conditions observed

Information about the condition of a component.

- **conditions.status** (string), required

Status of the condition for a component. Valid values for "Healthy": "True", "False", or "Unknown".

- **conditions.type** (string), required

Type of condition for a component. Valid value: "Healthy"

- **conditions.error** (string)

Condition error code for a component. For example, a health check error code.

- **conditions.message** (string)

Message about the condition for a component. For example, information about a health check.

ComponentStatusList

Status of all the conditions for the component as a list of ComponentStatus objects. Deprecated: This API is deprecated in v1.19+

- **apiVersion**: v1

- **kind**: ComponentStatusList

- **metadata** ([ListMeta](#))

Standard list metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types->

[kinds](#)

- **items** ([][ComponentStatus](#)), required

List of ComponentStatus objects.

Operations

[get](#) read the specified ComponentStatus

HTTP Request

GET /api/v1/componentstatuses/{name}

Parameters

- **name** (*in path*): string, required

name of the ComponentStatus

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ComponentStatus](#)): OK

401: Unauthorized

[list](#) list objects of kind ComponentStatus

HTTP Request

GET /api/v1/componentstatuses

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([ComponentStatusList](#)): OK

401: Unauthorized

5.8.12 - ClusterCIDR v1alpha1

ClusterCIDR represents a single configuration for per-Node Pod CIDR allocations when the MultiCIDRRRangeAllocator is enabled (see the config for kube-controller-manager).

```
apiVersion: networking.k8s.io/v1alpha1

import "k8s.io/api/networking/v1alpha1"
```

ClusterCIDR

ClusterCIDR represents a single configuration for per-Node Pod CIDR allocations when the MultiCIDRRRangeAllocator is enabled (see the config for kube-controller-manager). A cluster may have any number of ClusterCIDR resources, all of which will be considered when allocating a CIDR for a Node. A ClusterCIDR is eligible to be used for a given Node when the node selector matches the node in question and has free CIDRs to allocate. In case of multiple matching ClusterCIDR resources, the allocator will attempt to break ties using internal heuristics, but any ClusterCIDR whose node selector matches the Node may be used.

- **apiVersion**: networking.k8s.io/v1alpha1

- **kind**: ClusterCIDR

- **metadata** ([ObjectMeta](#))

Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **spec** ([ClusterCIDRSpec](#))

spec is the desired state of the ClusterCIDR. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status>

ClusterCIDRSpec

ClusterCIDRSpec defines the desired state of ClusterCIDR.

- **perNodeHostBits** (int32), required

perNodeHostBits defines the number of host bits to be configured per node. A subnet mask determines how much of the address is used for network bits and host bits. For example an IPv4 address of 192.168.0.0/24, splits the address into 24 bits for the network portion and 8 bits for the host portion. To allocate 256 IPs, set this field to 8 (a /24 mask for IPv4 or a /120 for IPv6). Minimum value is 4 (16 IPs). This field is immutable.

- **ipv4** (string)

ipv4 defines an IPv4 IP block in CIDR notation(e.g. "10.0.0.0/8"). At least one of ipv4 and ipv6 must be specified. This field is immutable.

- **ipv6** (string)

ipv6 defines an IPv6 IP block in CIDR notation(e.g. "2001:db8::/64").

At least one of ipv4 and ipv6 must be specified. This field is immutable.

- **nodeSelector** (NodeSelector)

nodeSelector defines which nodes the config is applicable to. An empty or nil nodeSelector selects all nodes. This field is immutable.

A node selector represents the union of the results of one or more label queries over a set of nodes; that is, it represents the OR of the selectors represented by the node selector terms.

- **nodeSelector.nodeSelectorTerms** ([]NodeSelectorTerm), required

Required. A list of node selector terms. The terms are ORed.

A null or empty node selector term matches no objects. The requirements of them are ANDed. The TopologySelectorTerm type implements a subset of the NodeSelectorTerm.

- **nodeSelector.nodeSelectorTerms.matchExpressions** ([]NodeSelectorRequirement)

A list of node selector requirements by node's labels.

- **nodeSelector.nodeSelectorTerms.matchFields** ([]NodeSelectorRequirement)

A list of node selector requirements by node's fields.

ClusterCIDRList

ClusterCIDRList contains a list of ClusterCIDR.

- **apiVersion**: networking.k8s.io/v1alpha1

- **kind**: ClusterCIDRList

- **metadata** ([ListMeta](#))

Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

- **items** ([]ClusterCIDR), required

items is the list of ClusterCIDRs.

Operations

get read the specified ClusterCIDR

HTTP Request

GET /apis/networking.k8s.io/v1alpha1/clustercidrs/{name}

Parameters

- **name** (*in path*): string, required

name of the ClusterCIDR

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ClusterCIDR](#)): OK

401: Unauthorized

list list or watch objects of kind ClusterCIDR

HTTP Request

GET /apis/networking.k8s.io/v1alpha1/clustercidrs

Parameters

- **allowWatchBookmarks** (*in query*): boolean

[allowWatchBookmarks](#)

- **continue** (*in query*): string

[continue](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

- **watch** (*in query*): boolean

[watch](#)

Response

200 ([ClusterCIDRList](#)): OK

401: Unauthorized

create create a ClusterCIDR

HTTP Request

POST /apis/networking.k8s.io/v1alpha1/clustercidrs

Parameters

- **body**: [ClusterCIDR](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([ClusterCIDR](#)): OK

201 ([ClusterCIDR](#)): Created

202 ([ClusterCIDR](#)): Accepted

401: Unauthorized

update replace the specified ClusterCIDR

HTTP Request

PUT /apis/networking.k8s.io/v1alpha1/clustercidrs/{name}

Parameters

- **name** (*in path*): string, required
name of the ClusterCIDR
- **body**: [ClusterCIDR](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **fieldValidation** (*in query*): string
[fieldValidation](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([ClusterCIDR](#)): OK

201 ([ClusterCIDR](#)): Created

401: Unauthorized

patch partially update the specified ClusterCIDR

HTTP Request

PATCH /apis/networking.k8s.io/v1alpha1/clustercidrs/{name}

Parameters

- **name** (*in path*): string, required

name of the ClusterCIDR

- **body**: [Patch](#), required

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldManager** (*in query*): string

[fieldManager](#)

- **fieldValidation** (*in query*): string

[fieldValidation](#)

- **force** (*in query*): boolean

[force](#)

- **pretty** (*in query*): string

[pretty](#)

Response

200 ([ClusterCIDR](#)): OK

201 ([ClusterCIDR](#)): Created

401: Unauthorized

delete delete a ClusterCIDR

HTTP Request

DELETE /apis/networking.k8s.io/v1alpha1/clustercidrs/{name}

Parameters

- **name** (*in path*): string, required

name of the ClusterCIDR

- **body**: [DeleteOptions](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

Response

200 ([Status](#)): OK

202 ([Status](#)): Accepted

401: Unauthorized

deletecollection delete collection of ClusterCIDR

HTTP Request

DELETE /apis/networking.k8s.io/v1alpha1/clustercidrs

Parameters

- **body**: [DeleteOptions](#)

- **continue** (*in query*): string

[continue](#)

- **dryRun** (*in query*): string

[dryRun](#)

- **fieldSelector** (*in query*): string

[fieldSelector](#)

- **gracePeriodSeconds** (*in query*): integer

[gracePeriodSeconds](#)

- **labelSelector** (*in query*): string

[labelSelector](#)

- **limit** (*in query*): integer

[limit](#)

- **pretty** (*in query*): string

[pretty](#)

- **propagationPolicy** (*in query*): string

[propagationPolicy](#)

- **resourceVersion** (*in query*): string

[resourceVersion](#)

- **resourceVersionMatch** (*in query*): string

[resourceVersionMatch](#)

- **sendInitialEvents** (*in query*): boolean

[sendInitialEvents](#)

- **timeoutSeconds** (*in query*): integer

[timeoutSeconds](#)

Response

200 ([Status](#)): OK

401: Unauthorized

5.9 - Common Definitions

5.9.1 - DeleteOptions

DeleteOptions may be provided when deleting an API object.

```
import "k8s.io/apimachinery/pkg/apis/meta/v1"
```

DeleteOptions may be provided when deleting an API object.

- **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>

- **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

- **gracePeriodSeconds** (int64)

The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately.

- **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>

- **orphanDependents** (boolean)

Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both.

- **preconditions** (Preconditions)

Must be fulfilled before a deletion is carried out. If not possible, a 409 Conflict status will be returned.

Preconditions must be fulfilled before an operation (update, delete, etc.) is carried out.

- **preconditions.resourceVersion** (string)

Specifies the target ResourceVersion

- **preconditions.uid** (string)

Specifies the target UID.

- **propagationPolicy** (string)

Whether and how garbage collection will be performed. Either this field or `OrphanDependents` may be set, but not both. The default policy is decided by the existing finalizer set in the `metadata.finalizers` and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground.

5.9.2 - LabelSelector

A label selector is a label query over a set of resources.

```
import "k8s.io/apimachinery/pkg/apis/meta/v1"
```

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.

- **matchExpressions** ([]LabelSelectorRequirement)

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

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

- **matchExpressions.key** (string), required

key is the label key that the selector applies to.

- **matchExpressions.operator** (string), required

operator represents a key's relationship to a set of values.

Valid operators are In, NotIn, Exists and DoesNotExist.

- **matchExpressions.values** ([]string)

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.

- **matchLabels** (map[string]string)

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.

5.9.3 - ListMeta

ListMeta describes metadata that synthetic resources must have, including lists and various status objects.

```
import "k8s.io/apimachinery/pkg/apis/meta/v1"
```

ListMeta describes metadata that synthetic resources must have, including lists and various status objects. A resource may have only one of {ObjectMeta, ListMeta}.

- **continue** (string)

continue may be set if the user set a limit on the number of items returned, and indicates that the server has more data available. The value is opaque and may be used to issue another request to the endpoint that served this list to retrieve the next set of available objects. Continuing a consistent list may not be possible if the server configuration has changed or more than a few minutes have passed. The resourceVersion field returned when using this continue value will be identical to the value in the first response, unless you have received this token from an error message.

- **remainingItemCount** (int64)

remainingItemCount is the number of subsequent items in the list which are not included in this list response. If the list request contained label or field selectors, then the number of remaining items is unknown and the field will be left unset and omitted during serialization. If the list is complete (either because it is not chunking or because this is the last chunk), then there are no more remaining items and this field will be left unset and omitted during serialization. Servers older than v1.15 do not set this field. The intended use of the remainingItemCount is *estimating* the size of a collection. Clients should not rely on the remainingItemCount to be set or to be exact.

- **resourceVersion** (string)

String that identifies the server's internal version of this object that can be used by clients to determine when objects have changed. Value must be treated as opaque by clients and passed unmodified back to the server. Populated by the system. Read-only. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency>

- **selfLink** (string)

Deprecated: selfLink is a legacy read-only field that is no longer populated by the system.

5.9.4 - LocalObjectReference

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

```
import "k8s.io/api/core/v1"
```

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

-
- **name** (string)

Name of the referent. More info: <https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names>

5.9.5 - NodeSelectorRequirement

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

```
import "k8s.io/api/core/v1"
```

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

- **key** (string), required

The label key that the selector applies to.

- **operator** (string), required

Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist, Gt, and Lt.

- **values** ([]string)

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. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

5.9.6 - ObjectFieldSelector

ObjectFieldSelector selects an APIVersioned field of an object.

```
import "k8s.io/api/core/v1"
```

ObjectFieldSelector selects an APIVersioned field of an object.

- **fieldPath** (string), required

Path of the field to select in the specified API version.

- **apiVersion** (string)

Version of the schema the FieldPath is written in terms of, defaults to "v1".

5.9.7 - ObjectMeta

ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.

```
import "k8s.io/apimachinery/pkg/apis/meta/v1"
```

ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.

- **name** (string)

Name must be unique within a namespace. Is required when creating resources, although some resources may allow a client to request the generation of an appropriate name automatically. Name is primarily intended for creation idempotence and configuration definition. Cannot be updated. More info: <https://kubernetes.io/docs/concepts/overview/working-with-objects/names#names>

- **generateName** (string)

GenerateName is an optional prefix, used by the server, to generate a unique name ONLY IF the Name field has not been provided. If this field is used, the name returned to the client will be different than the name passed. This value will also be combined with a unique suffix. The provided value has the same validation rules as the Name field, and may be truncated by the length of the suffix required to make the value unique on the server.

If this field is specified and the generated name exists, the server will return a 409.

Applied only if Name is not specified. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#idempotency>

- **namespace** (string)

Namespace defines the space within which each name must be unique. An empty namespace is equivalent to the "default" namespace, but "default" is the canonical representation. Not all objects are required to be scoped to a namespace - the value of this field for those objects will be empty.

Must be a DNS_LABEL. Cannot be updated. More info: <https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces>

- **labels** (map[string]string)

Map of string keys and values that can be used to organize and categorize (scope and select) objects. May match selectors of replication controllers and services. More info: <https://kubernetes.io/docs/concepts/overview/working-with-objects/labels>

- **annotations** (map[string]string)

Annotations is an unstructured key value map stored with a resource that may be set by external tools to store and retrieve arbitrary metadata. They are not queryable and should be preserved when modifying objects. More info: <https://kubernetes.io/docs/concepts/overview/working-with-objects/annotations>

System

- **finalizers** ([]string)

Must be empty before the object is deleted from the registry. Each entry is an identifier for the responsible component that will remove the entry from the list. If the deletionTimestamp of the object is non-nil, entries in this list can only be removed. Finalizers may be processed and removed in any order. Order is NOT enforced because it introduces significant risk of stuck finalizers. finalizers is a shared field, any actor with permission can reorder it. If the finalizer list is processed in order, then this can lead to a situation in which the component responsible for the first finalizer in the list is waiting for a signal (field value, external system, or other) produced by a component responsible for a finalizer later in the list, resulting in a deadlock. Without enforced ordering finalizers are free to order amongst themselves and are not vulnerable to ordering changes in the list.

- **managedFields** ([]ManagedFieldsEntry)

ManagedFields maps workflow-id and version to the set of fields that are managed by that workflow. This is mostly for internal housekeeping, and users typically shouldn't need to set or understand this field. A workflow can be the user's name, a controller's name, or the name of a specific apply path like "ci-cd". The set of fields is always in the version that the workflow used when modifying the object.

ManagedFieldsEntry is a workflow-id, a FieldSet and the group version of the resource that the fieldset applies to.

- **managedFields.apiVersion** (string)

APIVersion defines the version of this resource that this field set applies to. The format is "group/version" just like the top-level APIVersion field. It is necessary to track the version of a field set because it cannot be automatically converted.

- **managedFields.fieldsType** (string)

FieldsType is the discriminator for the different fields format and version. There is currently only one possible value: "FieldsV1"

- **managedFields.fieldsV1** (FieldsV1)

FieldsV1 holds the first JSON version format as described in the "FieldsV1" type.

*FieldsV1 stores a set of fields in a data structure like a Trie, in JSON format.

Each key is either a '.' representing the field itself, and will always map to an empty set, or a string representing a sub-field or item. The string will follow one of these four formats: 'f:', where is the name of a field in a struct, or key in a map 'v:', where is the exact json formatted value of a list item 'i:', where is position of a item in a list 'k:', where is a map of a list item's key fields to their unique values If a key maps to an empty Fields value, the field that key represents is part of the set.

The exact format is defined in [sigs.k8s.io/structured-merge-diff*](https://sigs.k8s.io/structured-merge-diff)

- **managedFields.manager** (string)

Manager is an identifier of the workflow managing these fields.

- **managedFields.operation** (string)

Operation is the type of operation which lead to this ManagedFieldsEntry being created. The only valid values for this field are 'Apply' and 'Update'.

- **managedFields.subresource** (string)

Subresource is the name of the subresource used to update that object, or empty string if the object was updated through the main resource. The value of this field is used to distinguish between managers, even if they share the same name. For example, a status update will be distinct from a regular update using the same manager name. Note that the APIVersion field is not related to the Subresource field and it always corresponds to the version of the main resource.

- **managedFields.time** (Time)

Time is the timestamp of when the ManagedFields entry was added. The timestamp will also be updated if a field is added, the manager changes any of the owned fields value or removes a field. The timestamp does not update when a field is removed from the entry because another manager took it over.

Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

- **ownerReferences** ([]OwnerReference)

Patch strategy: merge on key uid

List of objects depended by this object. If ALL objects in the list have been deleted, this object will be garbage collected. If this object is managed by a controller, then an entry in this list will point to this controller, with the controller field set to true. There cannot be more than one managing controller.

OwnerReference contains enough information to let you identify an owning object. An owning object must be in the same namespace as the dependent, or be cluster-scoped, so there is no namespace field.

- **ownerReferences.apiVersion** (string), required

API version of the referent.

- **ownerReferences.kind** (string), required

Kind of the referent. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds>

- **ownerReferences.name** (string), required

Name of the referent. More info: <https://kubernetes.io/docs/concepts/overview/working-with-objects/names#names>

- **ownerReferences.uid** (string), required

UID of the referent. More info: <https://kubernetes.io/docs/concepts/overview/working-with-objects/names#uids>

- **ownerReferences.blockOwnerDeletion** (boolean)

If true, AND if the owner has the "foregroundDeletion" finalizer, then the owner cannot be deleted from the key-value store until this reference is removed. See <https://kubernetes.io/docs/concepts/architecture/garbage-collection/#foreground-deletion> for how the garbage collector interacts with this field and enforces the foreground deletion. Defaults to false. To set this field, a user needs "delete" permission of the owner, otherwise 422 (Unprocessable Entity) will be returned.

- **ownerReferences.controller** (boolean)

If true, this reference points to the managing controller.

Read-only

- **creationTimestamp** (Time)

CreationTimestamp is a timestamp representing the server time when this object was created. It is not guaranteed to be set in happens-before order across separate operations. Clients may not set this value. It is represented in RFC3339 form and is in UTC.

Populated by the system. Read-only. Null for lists. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

- **deletionGracePeriodSeconds** (int64)

Number of seconds allowed for this object to gracefully terminate before it will be removed from the system. Only set when deletionTimestamp is also set. May only be shortened. Read-only.

- **deletionTimestamp** (Time)

DeletionTimestamp is RFC 3339 date and time at which this resource will be deleted. This field is set by the server when a graceful deletion is requested by the user, and is not directly settable by a client. The resource is expected to be deleted (no longer visible from resource lists, and not reachable by name) after the time in this field, once the finalizers list is empty. As long as the finalizers list contains items, deletion is blocked. Once the deletionTimestamp is set, this value may not be unset or be set further into the future, although it may be shortened or the resource may be deleted prior to this time. For example, a user may request that a pod is deleted in 30 seconds. The Kubelet will react by sending a graceful termination signal to the containers in the pod. After that 30 seconds, the Kubelet will send a hard termination signal (SIGKILL) to the container and after cleanup, remove the pod from the API. In the presence of network partitions, this object may still exist after this timestamp, until an administrator or automated process can determine the resource is fully terminated. If not set, graceful deletion of the object has not been requested.

Populated by the system when a graceful deletion is requested. Read-only. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

- **generation** (int64)

A sequence number representing a specific generation of the desired state. Populated by the system. Read-only.

- **resourceVersion** (string)

An opaque value that represents the internal version of this object that can be used by clients to determine when objects have changed. May be used for optimistic concurrency, change detection, and the watch operation on a resource or set of resources. Clients must treat these values as opaque and passed unmodified back to the server. They may only be valid for a particular resource or set of resources.

Populated by the system. Read-only. Value must be treated as opaque by clients and . More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency>

- **selfLink** (string)

Deprecated: selfLink is a legacy read-only field that is no longer populated by the system.

- **uid** (string)

UID is the unique in time and space value for this object. It is typically generated by the server on successful creation of a resource and is not allowed to change on PUT operations.

Populated by the system. Read-only. More info: <https://kubernetes.io/docs/concepts/overview/working-with-objects/names#uids>

5.9.8 - ObjectReference

ObjectReference contains enough information to let you inspect or modify the referred object.

```
import "k8s.io/api/core/v1"
```

ObjectReference contains enough information to let you inspect or modify the referred object.

- **apiVersion** (string)

API version of the referent.

- **fieldPath** (string)

If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as `desiredState.manifest.containers[2]`. For example, if the object reference is to a container within a pod, this would take on a value like: `"spec.containers{name}"` (where "name" refers to the name of the container that triggered the event) or if no container name is specified `"spec.containers[2]"` (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.

- **kind** (string)

Kind of the referent. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds>

- **name** (string)

Name of the referent. More info: <https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names>

- **namespace** (string)

Namespace of the referent. More info: <https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/>

- **resourceVersion** (string)

Specific resourceVersion to which this reference is made, if any.

More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency>

- **uid** (string)

UID of the referent. More info: <https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids>

5.9.9 - Patch

Patch is provided to give a concrete name and type to the Kubernetes PATCH request body.

```
import "k8s.io/apimachinery/pkg/apis/meta/v1"
```

Patch is provided to give a concrete name and type to the Kubernetes PATCH request body.

5.9.10 - Quantity

Quantity is a fixed-point representation of a number.

```
import "k8s.io/apimachinery/pkg/api/resource"
```

Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON and YAML, in addition to String() and AsInt64() accessors.

The serialization format is:

```
(Note that \<suffix> may be empty, from the "" case in \<decim
\<digit>      ::= 0 | 1 | ... | 9 \<digits>      ::= \<digit>
  (International System of units; See: http://physics.nist.gov/
\<decimalSI>    ::= m | "" | k | M | G | T | P | E

  (Note that 1024 = 1Ki but 1000 = 1k; I didn't choose the capi
\<decimalExponent> ::= "e" \<signedNumber> | "E" \<signedNumber> ```

No matter which of the three exponent forms is used, no quantity may
When a Quantity is parsed from a string, it will remember the type of
Before serializing, Quantity will be put in "canonical form". This me
- No precision is lost - No fractional digits will be emitted - The e
The sign will be omitted unless the number is negative.

Examples:
- 1.5 will be serialized as "1500m" - 1.5Gi will be serialized as "15
Note that the quantity will NEVER be internally represented by a floa
Non-canonical values will still parse as long as they are well formed
This format is intended to make it difficult to use these numbers wit
<hr>
```

5.9.11 - ResourceFieldSelector

ResourceFieldSelector represents container resources (cpu, memory) and their output format.

```
import "k8s.io/api/core/v1"
```

ResourceFieldSelector represents container resources (cpu, memory) and their output format

- **resource** (string), required

Required: resource to select

- **containerName** (string)

Container name: required for volumes, optional for env vars

- **divisor** ([Quantity](#))

Specifies the output format of the exposed resources, defaults to "1"

5.9.12 - Status

Status is a return value for calls that don't return other objects.

```
import "k8s.io/apimachinery/pkg/apis/meta/v1"
```

Status is a return value for calls that don't return other objects.

- **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>

- **code** (int32)

Suggested HTTP return code for this status, 0 if not set.

- **details** (StatusDetails)

Extended data associated with the reason. Each reason may define its own extended details. This field is optional and the data returned is not guaranteed to conform to any schema except that defined by the reason type.

StatusDetails is a set of additional properties that MAY be set by the server to provide additional information about a response. The Reason field of a Status object defines what attributes will be set. Clients must ignore fields that do not match the defined type of each attribute, and should assume that any attribute may be empty, invalid, or under defined.

- **details.causes** ([]StatusCause)

The Causes array includes more details associated with the StatusReason failure. Not all StatusReasons may provide detailed causes.

StatusCause provides more information about an api.Status failure, including cases when multiple errors are encountered.

- **details.causes.field** (string)

The field of the resource that has caused this error, as named by its JSON serialization. May include dot and postfix notation for nested attributes. Arrays are zero-indexed. Fields may appear more than once in an array of causes due to fields having multiple errors. Optional.

Examples: "name" - the field "name" on the current resource "items[0].name" - the field "name" on the first array entry in "items"

- **details.causes.message** (string)

A human-readable description of the cause of the error. This field may be presented as-is to a reader.

- **details.causes.reason** (string)

A machine-readable description of the cause of the error. If this value is empty there is no information available.

- **details.group** (string)

The group attribute of the resource associated with the status StatusReason.

- **details.kind** (string)

The kind attribute of the resource associated with the status StatusReason. On some operations may differ from the requested resource Kind. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds>

- **details.name** (string)

The name attribute of the resource associated with the status StatusReason (when there is a single name which can be described).

- **details.retryAfterSeconds** (int32)

If specified, the time in seconds before the operation should be retried. Some errors may indicate the client must take an alternate action - for those errors this field may indicate how long to wait before taking the alternate action.

- **details.uid** (string)

UID of the resource. (when there is a single resource which can be described). More info: <https://kubernetes.io/docs/concepts/overview/working-with-objects/names#uids>

- **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>

- **message** (string)

A human-readable description of the status of this operation.

- **metadata** ([ListMeta](#))

Standard list metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds>

- **reason** (string)

A machine-readable description of why this operation is in the "Failure" status. If this value is empty there is no information available. A Reason clarifies an HTTP status code but does not override it.

- **status** (string)

Status of the operation. One of: "Success" or "Failure". More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status>

5.9.13 - **TypedLocalObjectReference**

TypedLocalObjectReference contains enough information to let you locate the typed referenced object inside the same namespace.

```
import "k8s.io/api/core/v1"
```

TypedLocalObjectReference contains enough information to let you locate the typed referenced object inside the same namespace.

- **kind** (string), required

Kind is the type of resource being referenced

- **name** (string), required

Name is the name of resource being referenced

- **apiGroup** (string)

APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.

5.10 - Other Resources

5.10.1 - **ValidatingAdmissionPolicyBindingList** **v1beta1**

```
apiVersion: admissionregistration.k8s.io/v1beta1
```

```
import "k8s.io/api/admissionregistration/v1beta1"
```

5.11 - Common Parameters

allowWatchBookmarks

allowWatchBookmarks requests watch events with type "BOOKMARK". Servers that do not implement bookmarks may ignore this flag and bookmarks are sent at the server's discretion. Clients should not assume bookmarks are returned at any specific interval, nor may they assume the server will send any BOOKMARK event during a session. If this is not a watch, this field is ignored.

continue

The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server, the server will respond with a 410 ResourceExpired error together with a continue token. If the client needs a consistent list, it must restart their list without the continue field. Otherwise, the client may send another list request with the token received with the 410 error, the server will respond with a list starting from the next key, but from the latest snapshot, which is inconsistent from the previous list results - objects that are created, modified, or deleted after the first list request will be included in the response, as long as their keys are after the "next key".

This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications.

dryRun

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

fieldManager

fieldManager is a name associated with the actor or entity that is making these changes. The value must be less than or 128 characters long, and only contain printable characters, as defined by <https://pkg.go.dev/unicode#IsPrint>.

fieldSelector

A selector to restrict the list of returned objects by their fields. Defaults to everything.

fieldValidation

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.

force

Force is going to "force" Apply requests. It means user will re-acquire conflicting fields owned by other people. Force flag must be unset for non-apply patch requests.

gracePeriodSeconds

The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately.

labelSelector

A selector to restrict the list of returned objects by their labels. Defaults to everything.

limit

limit is a maximum number of responses to return for a list call. If more items exist, the server will set the `continue` field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the `continue` field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the `continue` field is empty, clients may assume that no more results are available. This field is not supported if `watch` is true.

The server guarantees that the objects returned when using `continue` will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is

using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned.

namespace

object name and auth scope, such as for teams and projects

pretty

If 'true', then the output is pretty printed.

propagationPolicy

Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground.

resourceVersion

resourceVersion sets a constraint on what resource versions a request may be served from. See <https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions> for details.

Defaults to unset

resourceVersionMatch

resourceVersionMatch determines how resourceVersion is applied to list calls. It is highly recommended that resourceVersionMatch be set for list calls where resourceVersion is set. See <https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions> for details.

Defaults to unset

sendInitialEvents

`sendInitialEvents=true` may be set together with `watch=true`. In that case, the watch stream will begin with synthetic events to produce the current state of objects in the collection. Once all such events have been sent, a synthetic "Bookmark" event will be sent. The bookmark will report the ResourceVersion (RV) corresponding to the set of objects, and be marked with `"k8s.io/initial-events-end": "true"` annotation. Afterwards, the watch stream will proceed as usual, sending watch events corresponding to changes (subsequent to the RV) to objects watched.

When `sendInitialEvents` option is set, we require `resourceVersionMatch` option to also be set. The semantic of the watch request is as following: - `resourceVersionMatch = NotOlderThan` is interpreted as "data at least as new as the provided `resourceVersion`" and the bookmark event is send when the state is synced to a `resourceVersion` at least as fresh as the one provided by the `ListOptions`. If `resourceVersion` is unset, this is interpreted as "consistent read" and the bookmark event is send when the state is synced at least to the moment when request started being processed.

- `resourceVersionMatch` set to any other value or unset Invalid error is returned.

Defaults to true if `resourceVersion=""` or `resourceVersion="0"` (for backward compatibility reasons) and to false otherwise.

timeoutSeconds

Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity.

watch

Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify `resourceVersion`.

6 - Instrumentation

6.1 - Kubernetes Component SLI Metrics

ⓘ FEATURE STATE: Kubernetes v1.29 [stable]

By default, Kubernetes 1.31 publishes Service Level Indicator (SLI) metrics for each Kubernetes component binary. This metric endpoint is exposed on the serving HTTPS port of each component, at the path `/metrics/slis`. The `ComponentSLIs` [feature gate](#) defaults to enabled for each Kubernetes component as of v1.27.

SLI Metrics

With SLI metrics enabled, each Kubernetes component exposes two metrics, labeled per healthcheck:

- a gauge (which represents the current state of the healthcheck)
 - a counter (which records the cumulative counts observed for each healthcheck state)

You can use the metric information to calculate per-component availability statistics. For example, the API server checks the health of etcd. You can work out and report how available or unavailable etcd has been - as reported by its client, the API server.

The prometheus gauge data looks like this:

```
# HELP kubernetes_healthcheck [ALPHA] This metric records the result
# TYPE kubernetes_healthcheck gauge
kubernetes_healthcheck{name="autoregister-completion",type="healthz"} 1
kubernetes_healthcheck{name="autoregister-completion",type="readyz"} 1
kubernetes_healthcheck{name="etcd",type="healthz"} 1
kubernetes_healthcheck{name="etcd",type="readyz"} 1
kubernetes_healthcheck{name="etcd-readiness",type="readyz"} 1
kubernetes_healthcheck{name="informer-sync",type="readyz"} 1
kubernetes_healthcheck{name="log",type="healthz"} 1
kubernetes_healthcheck{name="log",type="readyz"} 1
kubernetes_healthcheck{name="ping",type="healthz"} 1
kubernetes_healthcheck{name="ping",type="readyz"} 1
```

While the counter data looks like this:

```
# HELP kubernetes_healthchecks_total [ALPHA] This metric records the
# TYPE kubernetes_healthchecks_total counter
kubernetes_healthchecks_total{name="autoregister-completion",status="up",type="healthcheck",value=1}
kubernetes_healthchecks_total{name="autoregister-completion",status="down",type="healthcheck",value=0}
kubernetes_healthchecks_total{name="autoregister-completion",status="unreachable",type="healthcheck",value=0}
kubernetes_healthchecks_total{name="etcd",status="success",type="healthcheck",value=1}
kubernetes_healthchecks_total{name="etcd",status="error",type="healthcheck",value=0}
kubernetes_healthchecks_total{name="etcd-readiness",status="success",type="healthcheck",value=1}
kubernetes_healthchecks_total{name="informer-sync",status="error",type="healthcheck",value=0}
kubernetes_healthchecks_total{name="informer-sync",status="success",type="healthcheck",value=1}
kubernetes_healthchecks_total{name="log",status="success",type="healthcheck",value=1}
kubernetes_healthchecks_total{name="log",status="error",type="healthcheck",value=0}
kubernetes_healthchecks_total{name="log",status="down",type="healthcheck",value=0}
kubernetes_healthchecks_total{name="log",status="unreachable",type="healthcheck",value=0}
kubernetes_healthchecks_total{name="ping",status="success",type="healthcheck",value=1}
kubernetes_healthchecks_total{name="ping",status="error",type="healthcheck",value=0}
```

Using this data

The component SLIs metrics endpoint is intended to be scraped at a high frequency. Scraping at a high frequency means that you end up with greater granularity of the gauge's signal, which can be then used to calculate SLOs. The `/metrics/slis` endpoint provides the raw data necessary to calculate an availability SLO for the respective Kubernetes component.

6.2 - CRI Pod & Container Metrics

Collection of Pod & Container metrics via the CRI.

ⓘ FEATURE STATE: Kubernetes v1.23 [alpha]

The [kubelet](#) collects pod and container metrics via [cAdvisor](#). As an alpha feature, Kubernetes lets you configure the collection of pod and container metrics via the Container Runtime Interface (CRI). You must enable the `PodAndContainerStatsFromCRI` [feature gate](#) and use a compatible CRI implementation (containerd $\geq 1.6.0$, CRI-O $\geq 1.23.0$) to use the CRI based collection mechanism.

CRI Pod & Container Metrics

With `PodAndContainerStatsFromCRI` enabled, the kubelet polls the underlying container runtime for pod and container stats instead of inspecting the host system directly using cAdvisor. The benefits of relying on the container runtime for this information as opposed to direct collection with cAdvisor include:

- Potential improved performance if the container runtime already collects this information during normal operations. In this case, the data can be re-used instead of being aggregated again by the kubelet.
- It further decouples the kubelet and the container runtime allowing collection of metrics for container runtimes that don't run processes directly on the host with kubelet where they are observable by cAdvisor (for example: container runtimes that use virtualization).

6.3 - Node metrics data

Mechanisms for accessing metrics at node, volume, pod and container level, as seen by the kubelet.

The [kubelet](#) gathers metric statistics at the node, volume, pod and container level, and emits this information in the [Summary API](#).

You can send a proxied request to the stats summary API via the Kubernetes API server.

Here is an example of a Summary API request for a node named `minikube` :

```
kubectl get --raw "/api/v1/nodes/minikube/proxy/stats/summary"
```

Here is the same API call using `curl` :

```
# You need to run "kubectl proxy" first
# Change 8080 to the port that "kubectl proxy" assigns
curl http://localhost:8080/api/v1/nodes/minikube/proxy/stats/summary
```

Note:

Beginning with `metrics-server` 0.6.x, `metrics-server` queries the `/metrics/resource` kubelet endpoint, and not `/stats/summary`.

Summary metrics API source

By default, Kubernetes fetches node summary metrics data using an embedded [cAdvisor](#) that runs within the kubelet. If you enable the `PodAndContainerStatsFromCRI` [feature gate](#) in your cluster, and you use a container runtime that supports statistics access via Container Runtime Interface (CRI), then the kubelet [fetches Pod- and container-level metric data using CRI](#), and not via cAdvisor.

What's next

The task pages for [Troubleshooting Clusters](#) discuss how to use a metrics pipeline that rely on these data.

6.4 - Kubernetes Metrics Reference

Details of the metric data that Kubernetes components export.

Metrics (v1.30)

This page details the metrics that different Kubernetes components export. You can query the metrics endpoint for these components using an HTTP scrape, and fetch the current metrics data in Prometheus format.

List of Stable Kubernetes Metrics

Stable metrics observe strict API contracts and no labels can be added or removed from stable metrics during their lifetime.

apiserver_admission_controller_admission_duration_seconds

Admission controller latency histogram in seconds, identified by name and broken out for each operation and API resource and type (validate or admit).

- **Stability Level:** STABLE
- **Type:** Histogram
- **Labels:** `name` `operation` `rejected` `type`

apiserver_admission_step_admission_duration_seconds

Admission sub-step latency histogram in seconds, broken out for each operation and API resource and step type (validate or admit).

- **Stability Level:** STABLE
- **Type:** Histogram
- **Labels:** `operation` `rejected` `type`

apiserver_admission_webhook_admission_duration_seconds

Admission webhook latency histogram in seconds, identified by name and broken out for each operation and API resource and type (validate or admit).

- **Stability Level:** STABLE
- **Type:** Histogram
- **Labels:** `name` `operation` `rejected` `type`

apiserver_current_inflight_requests

Maximal number of currently used inflight request limit of this apiserver per request kind in last second.

- **Stability Level:** STABLE
- **Type:** Gauge

• Labels: `request_duration`

apiserver_longrunning_requests

Gauge of all active long-running apiserver requests broken out by verb, group, version, resource, scope and component. Not all requests are tracked this way.

- **Stability Level:** STABLE

- **Type:** Gauge

- **Labels:** `component` `group` `resource` `scope` `subresource` `verb`
`version`

apiserver_request_duration_seconds

Response latency distribution in seconds for each verb, dry run value, group, version, resource, subresource, scope and component.

- **Stability Level:** STABLE

- **Type:** Histogram

- **Labels:** `component` `dry_run` `group` `resource` `scope` `subresource`
`verb` `version`

apiserver_request_total

Counter of apiserver requests broken out for each verb, dry run value, group, version, resource, scope, component, and HTTP response code.

- **Stability Level:** STABLE

- **Type:** Counter

- **Labels:** `code` `component` `dry_run` `group` `resource` `scope`
`subresource` `verb` `version`

apiserver_requested_DEPRECATED_apis

Gauge of deprecated APIs that have been requested, broken out by API group, version, resource, subresource, and removed_release.

- **Stability Level:** STABLE

- **Type:** Gauge

- **Labels:** `group` `removed_release` `resource` `subresource` `version`

apiserver_response_sizes

Response size distribution in bytes for each group, version, verb, resource, subresource, scope and component.

- **Stability Level:** STABLE

- **Type:** Histogram

- **Labels:** `component` `group` `resource` `scope` `subresource` `verb`
`version`

apiserver_storage_objects

Number of stored objects at the time of last check split by kind. In case of a fetching error, the value will be -1.

- **Stability Level:** STABLE

- **Type:** Gauge

- **Labels:** `resource`

apiserver_storage_size_bytes

Size of the storage database file physically allocated in bytes.

- **Stability Level:** STABLE

- **Type:** Custom

- **Labels:** `storage_cluster_id`

container_cpu_usage_seconds_total

Cumulative cpu time consumed by the container in core-seconds

- **Stability Level:** STABLE

- **Type:** Custom

- **Labels:** `container` `pod` `namespace`

container_memory_working_set_bytes

Current working set of the container in bytes

- **Stability Level:** STABLE

- **Type:** Custom

- **Labels:** `container` `pod` `namespace`

container_start_time_seconds

Start time of the container since unix epoch in seconds

- **Stability Level:** STABLE

- **Type:** Custom

- **Labels:** `container` `pod` `namespace`

cronjob_controller_job_creation_skew_duration_seconds

Time between when a cronjob is scheduled to be run, and when the corresponding job is created

- **Stability Level:** STABLE

- **Type:** Histogram

job_controller_job_pods_finished_total

The number of finished Pods that are fully tracked

- **Stability Level:** STABLE
- **Type:** Counter
- **Labels:** `completion_mode` `result`

job_controller_job_sync_duration_seconds

The time it took to sync a job

- **Stability Level:** STABLE
- **Type:** Histogram
- **Labels:** `action` `completion_mode` `result`

job_controller_job_syncs_total

The number of job syncs

- **Stability Level:** STABLE
- **Type:** Counter
- **Labels:** `action` `completion_mode` `result`

job_controller_jobs_finished_total

The number of finished jobs

- **Stability Level:** STABLE
- **Type:** Counter
- **Labels:** `completion_mode` `reason` `result`

kube_pod_resource_limit

Resources limit for workloads on the cluster, broken down by pod. This shows the resource usage the scheduler and kubelet expect per pod for resources along with the unit for the resource if any.

- **Stability Level:** STABLE
- **Type:** Custom
- **Labels:** `namespace` `pod` `node` `scheduler` `priority` `resource` `unit`

kube_pod_resource_request

Resources requested by workloads on the cluster, broken down by pod. This shows the resource usage the scheduler and kubelet expect per pod for resources along with the unit for the resource if any.

- **Stability Level:** STABLE
- **Type:** Custom
- **Labels:** `namespace` `pod` `node` `scheduler` `priority` `resource` `unit`

kubernetes_healthcheck

This metric records the result of a single healthcheck.

- **Stability Level:** STABLE

- **Type:** Gauge

- **Labels:** `name` `type`

kubernetes_healthchecks_total

This metric records the results of all healthcheck.

- **Stability Level:** STABLE

- **Type:** Counter

- **Labels:** `name` `status` `type`

node_collector_evictions_total

Number of Node evictions that happened since current instance of NodeController started.

- **Stability Level:** STABLE

- **Type:** Counter

- **Labels:** `zone`

node_cpu_usage_seconds_total

Cumulative cpu time consumed by the node in core-seconds

- **Stability Level:** STABLE

- **Type:** Custom

node_memory_working_set_bytes

Current working set of the node in bytes

- **Stability Level:** STABLE

- **Type:** Custom

pod_cpu_usage_seconds_total

Cumulative cpu time consumed by the pod in core-seconds

- **Stability Level:** STABLE

- **Type:** Custom

- **Labels:** `pod` `namespace`

pod_memory_working_set_bytes

Current working set of the pod in bytes

- **Stability Level:** STABLE
- **Type:** Custom
- **Labels:** `pod` `namespace`

resource_scrape_error

1 if there was an error while getting container metrics, 0 otherwise

- **Stability Level:** STABLE
- **Type:** Custom

scheduler_framework_extension_point_duration_seconds

Latency for running all plugins of a specific extension point.

- **Stability Level:** STABLE
- **Type:** Histogram
- **Labels:** `extension_point` `profile` `status`

scheduler_pending_pods

Number of pending pods, by the queue type. 'active' means number of pods in activeQ; 'backoff' means number of pods in backoffQ; 'unschedulable' means number of pods in unschedulablePods that the scheduler attempted to schedule and failed; 'gated' is the number of unschedulable pods that the scheduler never attempted to schedule because they are gated.

- **Stability Level:** STABLE
- **Type:** Gauge
- **Labels:** `queue`

scheduler_pod_scheduling_attempts

Number of attempts to successfully schedule a pod.

- **Stability Level:** STABLE
- **Type:** Histogram

scheduler_pod_scheduling_duration_seconds

E2e latency for a pod being scheduled which may include multiple scheduling attempts.

- **Stability Level:** STABLE
- **Type:** Histogram
- **Labels:** `attempts`
- **Deprecated Versions:** 1.29.0