OpenShift Container Platform 4.18

Extension APIs

Reference guide for extension APIs

OpenShift Container Platform 4.18 Extension APIs

Reference guide for extension APIs

Legal Notice

Abstract

This document describes the OpenShift Container Platform extension API objects and their detailed specifications.

Table of Contents

CHAPTER 1. EXTENSION APIS

1.1. APISERVICE [APIREGISTRATION.K8S.IO/V1]

Description

APIService represents a server for a particular GroupVersion. Name must be "version.group".

Type

object

1.2. CUSTOMRESOURCEDEFINITION [APIEXTENSIONS.K8S.IO/V1]

Description

CustomResourceDefinition represents a resource that should be exposed on the API server. Its name MUST be in the format <.spec.name>.<.spec.group>.

Type

object

1.3. MUTATINGWEBHOOKCONFIGURATION [ADMISSIONREGISTRATION.K8S.IO/V1]

Description

MutatingWebhookConfiguration describes the configuration of and admission webhook that accept or reject and may change the object.

Type

object

1.4. VALIDATINGADMISSIONPOLICY [ADMISSIONREGISTRATION.K8S.IO/V1]

Description

ValidatingAdmissionPolicy describes the definition of an admission validation policy that accepts or rejects an object without changing it.

Type

object

1.5. VALIDATINGADMISSIONPOLICYBINDING [ADMISSIONREGISTRATION.K8S.IO/V1]

Description

ValidatingAdmissionPolicyBinding binds the ValidatingAdmissionPolicy with paramerized 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.

Type

   object

1.6. VALIDATINGWEBHOOKCONFIGURATION [ADMISSIONREGISTRATION.K8S.IO/V1]

Description

   ValidatingWebhookConfiguration describes the configuration of and admission webhook that accept or reject and object without changing it.

Type

   object

CHAPTER 2. APISERVICE [APIREGISTRATION.K8S.IO/V1]

Description

APIService represents a server for a particular GroupVersion. Name must be "version.group".

Type

object

2.1. SPECIFICATION

Property	Type	Description
apiVersion	**string**	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	**string**	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	**ObjectMeta**	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	**object**	APIServiceSpec contains information for locating and communicating with a server. Only https is supported, though you are able to disable certificate verification.
status	**object**	APIServiceStatus contains derived information about an API server

2.1.1. .spec

Description

APIServiceSpec contains information for locating and communicating with a server. Only https is supported, though you are able to disable certificate verification.

Type

object

Required

- **groupPriorityMinimum**

- **versionPriority**

Property	Type	Description
caBundle	**string**	CABundle 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
groupPriorityMinimum	**integer**	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

Property	Type	Description
insecureSkipTLSVerify	**boolean**	InsecureSkipTLSVerify disables TLS certificate verification when communicating with this server. This is strongly discouraged. You should use the CABundle instead.
service	**object**	ServiceReference holds a reference to Service.legacy.k8s.io
version	**string**	Version is the API version this server hosts. For example, "v1"
versionPriority	**integer**	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.

2.1.2. .spec.service

Description

ServiceReference holds a reference to Service.legacy.k8s.io

Type

object

Property	Type	Description
name	**string**	Name is the name of the service
namespace	**string**	Namespace is the namespace of the service
port	**integer**	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).

2.1.3. .status

Description

APIServiceStatus contains derived information about an API server

Type

object

Property	Type	Description
conditions	**array**	Current service state of apiService.
conditions[]	**object**	APIServiceCondition describes the state of an APIService at a particular point

2.1.4. .status.conditions

Description

Current service state of apiService.

Type

array

2.1.5. .status.conditions[]

Description

APIServiceCondition describes the state of an APIService at a particular point

Type

object

Required

- **type**

- **status**

Property	Type	Description
lastTransitionTime	**Time**	Last time the condition transitioned from one status to another.
message	**string**	Human-readable message indicating details about last transition.
reason	**string**	Unique, one-word, CamelCase reason for the condition's last transition.
status	**string**	Status is the status of the condition. Can be True, False, Unknown.
type	**string**	Type is the type of the condition.

2.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/apiregistration.k8s.io/v1/apiservices**

    - **DELETE**: delete collection of APIService

    - **GET**: list or watch objects of kind APIService

    - **POST**: create an APIService

- **/apis/apiregistration.k8s.io/v1/watch/apiservices**

    - **GET**: watch individual changes to a list of APIService. deprecated: use the 'watch' parameter with a list operation instead.

- **/apis/apiregistration.k8s.io/v1/apiservices/{name}**

    - **DELETE**: delete an APIService

    - **GET**: read the specified APIService

    - **PATCH**: partially update the specified APIService

    - **PUT**: replace the specified APIService

- **/apis/apiregistration.k8s.io/v1/watch/apiservices/{name}**

    - **GET**: watch changes to an object of kind APIService. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

- **/apis/apiregistration.k8s.io/v1/apiservices/{name}/status**

  - **GET**: read status of the specified APIService

  - **PATCH**: partially update status of the specified APIService

  - **PUT**: replace status of the specified APIService

2.2.1. /apis/apiregistration.k8s.io/v1/apiservices

HTTP method

    DELETE

Description

    delete collection of APIService

Table 2.1. Query parameters

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

Table 2.2. HTTP responses

HTTP code	Reponse body
200 – OK	**Status** schema
401 – Unauthorized	Empty

HTTP method

    GET

Description

    list or watch objects of kind APIService

Table 2.3. HTTP responses

HTTP code	Reponse body
200 – OK	**APIServiceList** schema
401 – Unauthorized	Empty

HTTP method

    POST

Description

create an APIService

Table 2.4. Query parameters

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

Table 2.5. Body parameters

Parameter	Type	Description
body	**APIService** schema	

Table 2.6. HTTP responses

HTTP code	Reponse body
200 – OK	**APIService** schema
201 – Created	**APIService** schema
202 – Accepted	**APIService** schema
401 – Unauthorized	Empty

2.2.2. /apis/apiregistration.k8s.io/v1/watch/apiservices

HTTP method

GET

Description

watch individual changes to a list of APIService. deprecated: use the 'watch' parameter with a list operation instead.

Table 2.7. HTTP responses

HTTP code	Reponse body
200 - OK	**WatchEvent** schema
401 - Unauthorized	Empty

2.2.3. /apis/apiregistration.k8s.io/v1/apiservices/{name}

Table 2.8. Global path parameters

Parameter	Type	Description
name	**string**	name of the APIService

HTTP method

DELETE

Description

delete an APIService

Table 2.9. Query parameters

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

Table 2.10. HTTP responses

HTTP code	Reponse body
200 - OK	**Status** schema
202 - Accepted	**Status** schema

HTTP code	Reponse body
401 – Unauthorized	Empty

HTTP method

GET

Description

read the specified APIService

Table 2.11. HTTP responses

HTTP code	Reponse body
200 – OK	**APIService** schema
401 – Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified APIService

Table 2.12. Query parameters

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

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

Table 2.13. HTTP responses

HTTP code	Reponse body
200 - OK	**APIService** schema
201 – Created	**APIService** schema
401 – Unauthorized	Empty

HTTP method

> **PUT**

Description

> replace the specified APIService

Table 2.14. Query parameters

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

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

Table 2.15. Body parameters

Parameter	Type	Description
body	**APIService** schema	

Table 2.16. HTTP responses

HTTP code	Reponse body
200 – OK	**APIService** schema
201 – Created	**APIService** schema
401 – Unauthorized	Empty

2.2.4. /apis/apiregistration.k8s.io/v1/watch/apiservices/{name}

Table 2.17. Global path parameters

Parameter	Type	Description
name	**string**	name of the APIService

HTTP method

GET

Description

watch changes to an object of kind APIService. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

Table 2.18. HTTP responses

HTTP code	Reponse body
200 – OK	**WatchEvent** schema
401 – Unauthorized	Empty

2.2.5. /apis/apiregistration.k8s.io/v1/apiservices/{name}/status

Table 2.19. Global path parameters

Parameter	Type	Description
name	**string**	name of the APIService

HTTP method

GET

Description

read status of the specified APIService

Table 2.20. HTTP responses

HTTP code	Reponse body
200 – OK	**APIService** schema
401 – Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified APIService

Table 2.21. Query parameters

Parameter	Type	Description

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

Table 2.22. HTTP responses

HTTP code	Reponse body
200 – OK	**APIService** schema
201 – Created	**APIService** schema
401 – Unauthorized	Empty

HTTP method

    PUT

Description

    replace status of the specified APIService

Table 2.23. Query parameters

Parameter	Type	Description

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

Table 2.24. Body parameters

Parameter	Type	Description
body	**APIService** schema	

Table 2.25. HTTP responses

HTTP code	Reponse body
200 – OK	**APIService** schema
201 – Created	**APIService** schema
401 – Unauthorized	Empty

CHAPTER 3. CUSTOMRESOURCEDEFINITION [APIEXTENSIONS.K8S.IO/V1]

Description

CustomResourceDefinition represents a resource that should be exposed on the API server. Its name MUST be in the format <.spec.name>.<.spec.group>.

Type

object

Required

- **spec**

3.1. SPECIFICATION

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

Property	Type	Description
status	**object**	CustomResourceDefinitionStatus indicates the state of the CustomResourceDefinition

3.1.1. .spec

Description

CustomResourceDefinitionSpec describes how a user wants their resource to appear

Type

object

Required

- **group**

- **names**

- **scope**

- **versions**

Property	Type	Description
conversion	**object**	CustomResourceConversion describes how to convert different versions of a CR.
group	**string**	group is the API group of the defined custom resource. The custom resources are served under **/apis/<group>/…**. Must match the name of the CustomResourceDefinition (in the form **<names.plural>.<group>**).
names	**object**	CustomResourceDefinitionNames indicates the names to serve this CustomResourceDefinition

Property	Type	Description
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 known fields inside metadata are always preserved. This field is deprecated in favor of setting **x-preserve-unknown-fields** to true in **spec.versions[*].schema.ope nAPIV3Schema**. See [https://kubernetes.io/docs/tasks /extend-kubernetes/custom-resources/custom-resource-definitions/#field-pruning](https://kubernetes.io/docs/tasks/extend-kubernetes/custom-resources/custom-resource-definitions/#field-pruning) for details.
scope	**string**	scope indicates whether the defined custom resource is cluster- or namespace-scoped. Allowed values are **Cluster** and **Namespaced**.
versions	**array**	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.
versions[]	**object**	CustomResourceDefinitionVersio n describes a version for CRD.

3.1.2. .spec.conversion

Description

CustomResourceConversion describes how to convert different versions of a CR.

Type

object

Required

- **strategy**

Property	Type	Description
strategy	**string**	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.
webhook	**object**	WebhookConversion describes how to call a conversion webhook

3.1.3. .spec.conversion.webhook

Description

WebhookConversion describes how to call a conversion webhook

Type

object

Required

- **conversionReviewVersions**

Property	Type	Description
clientConfig	**object**	WebhookClientConfig contains the information to make a TLS connection with the webhook.

Property	Type	Description
conversionReviewVersions	**array (string)**	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.

3.1.4. .spec.conversion.webhook.clientConfig

Description

WebhookClientConfig contains the information to make a TLS connection with the webhook.

Type

object

Property	Type	Description
caBundle	**string**	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.
service	**object**	ServiceReference holds a reference to Service.legacy.k8s.io

Property	Type	Description
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.

3.1.5. .spec.conversion.webhook.clientConfig.service

Description

ServiceReference holds a reference to Service.legacy.k8s.io

Type

  object

Required

- **namespace**

- **name**

Property	Type	Description
name	**string**	name is the name of the service. Required
namespace	**string**	namespace is the namespace of the service. Required
path	**string**	path is an optional URL path at which the webhook will be contacted.
port	**integer**	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.

3.1.6. .spec.names

Description

CustomResourceDefinitionNames indicates the names to serve this CustomResourceDefinition

Type

object

Required

- **plural**

- **kind**

Property	Type	Description
categories	**array (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**.

Property	Type	Description
kind	**string**	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.
listKind	**string**	listKind is the serialized kind of the list for this resource. Defaults to "`kind`List".
plural	**string**	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.
shortNames	**array (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.
singular	**string**	singular is the singular name of the resource. It must be all lowercase. Defaults to lowercased **kind**.

3.1.7. .spec.versions

Description

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.

Type

array

3.1.8. .spec.versions[]

Description

CustomResourceDefinitionVersion describes a version for CRD.

Type

object

Required

- **name**

- **served**

- **storage**

Property	Type	Description
additionalPrinterColumns	**array**	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.
additionalPrinterColumns[]	**object**	CustomResourceColumnDefinition specifies a column for server side printing.
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.
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.

Property	Type	Description
name	**string**	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.
schema	**object**	CustomResourceValidation is a list of validation methods for CustomResources.
selectableFields	**array**	selectableFields specifies paths to fields that may be used as field selectors. A maximum of 8 selectable fields are allowed. See https://kubernetes.io/docs/concepts/overview/working-with-objects/field-selectors
selectableFields[]	**object**	SelectableField specifies the JSON path of a field that may be used with field selectors.
served	**boolean**	served is a flag enabling/disabling this version from being served via REST APIs
storage	**boolean**	storage indicates this version should be used when persisting custom resources to storage. There must be exactly one version with storage=true.
subresources	**object**	CustomResourceSubresources defines the status and scale subresources for CustomResources.

3.1.9. .spec.versions[].additionalPrinterColumns

Description

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.

Type

array

3.1.10. .spec.versions[].additionalPrinterColumns[]

Description

CustomResourceColumnDefinition specifies a column for server side printing.

Type

object

Required

- **name**

- **type**

- **jsonPath**

Property	Type	Description
description	**string**	description is a human readable description of this column.
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.
jsonPath	**string**	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.
name	**string**	name is a human readable name for the column.
priority	**integer**	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.

Property	Type	Description
type	**string**	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.

3.1.11. .spec.versions[].schema

Description

CustomResourceValidation is a list of validation methods for CustomResources.

Type

object

Property	Type	Description
openAPIV3Schema	`` `` ``	openAPIV3Schema is the OpenAPI v3 schema to use for validation and pruning.

3.1.12. .spec.versions[].selectableFields

Description

selectableFields specifies paths to fields that may be used as field selectors. A maximum of 8 selectable fields are allowed. See https://kubernetes.io/docs/concepts/overview/working-with-objects/field-selectors

Type

array

3.1.13. .spec.versions[].selectableFields[]

Description

SelectableField specifies the JSON path of a field that may be used with field selectors.

Type

object

Required

- **jsonPath**

Property	Type	Description
jsonPath	**string**	jsonPath is a simple JSON path which is evaluated against each custom resource to produce a field selector value. Only JSON paths without the array notation are allowed. Must point to a field of type string, boolean or integer. Types with enum values and strings with formats are allowed. If jsonPath refers to absent field in a resource, the jsonPath evaluates to an empty string. Must not point to metdata fields. Required.

3.1.14. .spec.versions[].subresources

Description

CustomResourceSubresources defines the status and scale subresources for CustomResources.

Type

object

Property	Type	Description
scale	**object**	CustomResourceSubresourceScale defines how to serve the scale subresource for CustomResources.
status	**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

3.1.15. .spec.versions[].subresources.scale

Description

CustomResourceSubresourceScale defines how to serve the scale subresource for CustomResources.

Type

object

Required

- **specReplicasPath**

- **statusReplicasPath**

Property	Type	Description
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.
specReplicasPath	**string**	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.

Property	Type	Description
statusReplicasPath	**string**	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.

3.1.16. .spec.versions[].subresources.status

Description

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

Type

object

3.1.17. .status

Description

CustomResourceDefinitionStatus indicates the state of the CustomResourceDefinition

Type

object

Property	Type	Description
acceptedNames	**object**	CustomResourceDefinitionNames indicates the names to serve this CustomResourceDefinition
conditions	**array**	conditions indicate state for particular aspects of a CustomResourceDefinition
conditions[]	**object**	CustomResourceDefinitionCondition contains details for the current condition of this pod.

Property	Type	Description
storedVersions	**array (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.

3.1.18. .status.acceptedNames

Description

CustomResourceDefinitionNames indicates the names to serve this CustomResourceDefinition

Type

object

Required

- **plural**

- **kind**

Property	Type	Description
categories	**array (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**.
kind	**string**	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.

Property	Type	Description
listKind	**string**	listKind is the serialized kind of the list for this resource. Defaults to "`kind`List".
plural	**string**	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.
shortNames	**array (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.
singular	**string**	singular is the singular name of the resource. It must be all lowercase. Defaults to lowercased **kind**.

3.1.19. .status.conditions

Description

conditions indicate state for particular aspects of a CustomResourceDefinition

Type

array

3.1.20. .status.conditions[]

Description

CustomResourceDefinitionCondition contains details for the current condition of this pod.

Type

object

Required

- **type**

- **status**

Property	Type	Description
lastTransitionTime	**Time**	lastTransitionTime last time the condition transitioned from one status to another.
message	**string**	message is a human-readable message indicating details about last transition.
reason	**string**	reason is a unique, one-word, CamelCase reason for the condition's last transition.
status	**string**	status is the status of the condition. Can be True, False, Unknown.
type	**string**	type is the type of the condition. Types include Established, NamesAccepted and Terminating.

3.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/apiextensions.k8s.io/v1/customresourcedefinitions**

    - **DELETE**: delete collection of CustomResourceDefinition

    - **GET**: list or watch objects of kind CustomResourceDefinition

    - **POST**: create a CustomResourceDefinition

- **/apis/apiextensions.k8s.io/v1/watch/customresourcedefinitions**

    - **GET**: watch individual changes to a list of CustomResourceDefinition. deprecated: use the 'watch' parameter with a list operation instead.

- **/apis/apiextensions.k8s.io/v1/customresourcedefinitions/{name}**

    - **DELETE**: delete a CustomResourceDefinition

    - **GET**: read the specified CustomResourceDefinition

    - **PATCH**: partially update the specified CustomResourceDefinition

    - **PUT**: replace the specified CustomResourceDefinition

- **/apis/apiextensions.k8s.io/v1/watch/customresourcedefinitions/{name}**

- **GET**: watch changes to an object of kind CustomResourceDefinition. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

- **/apis/apiextensions.k8s.io/v1/customresourcedefinitions/{name}/status**

  - **GET**: read status of the specified CustomResourceDefinition

  - **PATCH**: partially update status of the specified CustomResourceDefinition

  - **PUT**: replace status of the specified CustomResourceDefinition

3.2.1. /apis/apiextensions.k8s.io/v1/customresourcedefinitions

HTTP method

DELETE

Description

delete collection of CustomResourceDefinition

Table 3.1. Query parameters

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

Table 3.2. HTTP responses

HTTP code	Reponse body
200 – OK	**Status** schema
401 – Unauthorized	Empty

HTTP method

GET

Description

list or watch objects of kind CustomResourceDefinition

Table 3.3. HTTP responses

HTTP code	Reponse body
200 – OK	**CustomResourceDefinitionList** schema
401 – Unauthorized	Empty

HTTP method

POST

Description

create a CustomResourceDefinition

Table 3.4. Query parameters

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

Table 3.5. Body parameters

Parameter	Type	Description
body	**CustomResourceDefinition** schema	

Table 3.6. HTTP responses

HTTP code	Reponse body
200 – OK	**CustomResourceDefinition** schema
201 – Created	**CustomResourceDefinition** schema

HTTP code	Reponse body
202 – Accepted	**CustomResourceDefinition** schema
401 – Unauthorized	Empty

3.2.2. /apis/apiextensions.k8s.io/v1/watch/customresourcedefinitions

HTTP method

> **GET**

Description

> watch individual changes to a list of CustomResourceDefinition. deprecated: use the 'watch' parameter with a list operation instead.

Table 3.7. HTTP responses

HTTP code	Reponse body
200 – OK	**WatchEvent** schema
401 – Unauthorized	Empty

3.2.3. /apis/apiextensions.k8s.io/v1/customresourcedefinitions/{name}

Table 3.8. Global path parameters

Parameter	Type	Description
name	**string**	name of the CustomResourceDefinition

HTTP method

> **DELETE**

Description

> delete a CustomResourceDefinition

Table 3.9. Query parameters

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

Table 3.10. HTTP responses

HTTP code	Reponse body
200 – OK	**Status** schema
202 – Accepted	**Status** schema
401 – Unauthorized	Empty

HTTP method

GET

Description

read the specified CustomResourceDefinition

Table 3.11. HTTP responses

HTTP code	Reponse body
200 – OK	**CustomResourceDefinition** schema
401 – Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified CustomResourceDefinition

Table 3.12. Query parameters

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

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

Table 3.13. HTTP responses

HTTP code	Reponse body
200 – OK	**CustomResourceDefinition** schema
201 – Created	**CustomResourceDefinition** schema
401 – Unauthorized	Empty

HTTP method

    PUT

Description

    replace the specified CustomResourceDefinition

Table 3.14. Query parameters

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

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

Table 3.15. Body parameters

Parameter	Type	Description
body	**CustomResourceDefinition** schema	

Table 3.16. HTTP responses

HTTP code	Reponse body
200 – OK	**CustomResourceDefinition** schema
201 – Created	**CustomResourceDefinition** schema
401 – Unauthorized	Empty

3.2.4. /apis/apiextensions.k8s.io/v1/watch/customresourcedefinitions/{name}

Table 3.17. Global path parameters

Parameter	Type	Description
name	**string**	name of the CustomResourceDefinition

HTTP method

GET

Description

watch changes to an object of kind CustomResourceDefinition. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

Table 3.18. HTTP responses

HTTP code	Reponse body
200 – OK	**WatchEvent** schema
401 – Unauthorized	Empty

3.2.5. /apis/apiextensions.k8s.io/v1/customresourcedefinitions/{name}/status

Table 3.19. Global path parameters

Parameter	Type	Description
name	**string**	name of the CustomResourceDefinition

HTTP method

GET

Description

read status of the specified CustomResourceDefinition

Table 3.20. HTTP responses

HTTP code	Reponse body
200 – OK	**CustomResourceDefinition** schema
401 – Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified CustomResourceDefinition

Table 3.21. Query parameters

Parameter	Type	Description

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

Table 3.22. HTTP responses

HTTP code	Reponse body
200 – OK	**CustomResourceDefinition** schema
201 – Created	**CustomResourceDefinition** schema
401 – Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified CustomResourceDefinition

Table 3.23. Query parameters

Parameter	Type	Description

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

Table 3.24. Body parameters

Parameter	Type	Description
body	**CustomResourceDefinition** schema	

Table 3.25. HTTP responses

HTTP code	Reponse body
200 – OK	**CustomResourceDefinition** schema
201 – Created	**CustomResourceDefinition** schema
401 – Unauthorized	Empty

CHAPTER 4. MUTATINGWEBHOOKCONFIGURATION [ADMISSIONREGISTRATION.K8S.IO/V1]

Description

MutatingWebhookConfiguration describes the configuration of and admission webhook that accept or reject and may change the object.

Type

object

4.1. SPECIFICATION

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

Property	Type	Description
webhooks[]	**object**	MutatingWebhook describes an admission webhook and the resources and operations it applies to.

4.1.1. .webhooks

Description

Webhooks is a list of webhooks and the affected resources and operations.

Type

array

4.1.2. .webhooks[]

Description

MutatingWebhook describes an admission webhook and the resources and operations it applies to.

Type

object

Required

- **name**

- **clientConfig**

- **sideEffects**

- **admissionReviewVersions**

Property	Type	Description
admissionReviewVersions	**array (string)**	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.

Property	Type	Description
clientConfig	**object**	WebhookClientConfig contains the information to make a TLS connection with the webhook
failurePolicy	**string**	FailurePolicy defines how unrecognized errors from the admission endpoint are handled – allowed values are Ignore or Fail. Defaults to Fail. Possible enum values: – **"Fail"** means that an error calling the webhook causes the admission to fail. – **"Ignore"** means that an error calling the webhook is ignored.
matchConditions	**array**	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
matchConditions[]	**object**	MatchCondition represents a condition which must by fulfilled for a request to be sent to a webhook.

Property	Type	Description
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" Possible enum values: – **"Equivalent"** means requests should be sent to the webhook if they modify a resource listed in rules via another API group or version. – **"Exact"** means requests should only be sent to the webhook if they exactly match a given rule.

Property	Type	Description
name	**string**	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.
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.

Property	Type	Description
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.

Property	Type	Description
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". Possible enum values: – **"IfNeeded"** indicates that the webhook may 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. – **"Never"** indicates that the webhook must not be called more than once in a single admission evaluation.

Property	Type	Description
rules	**array**	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.
rules[]	**object**	RuleWithOperations is a tuple of Operations and Resources. It is recommended to make sure that all the tuple expansions are valid.

Property	Type	Description
sideEffects	**string**	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. Possible enum values: - **"None"** means that calling the webhook will have no side effects. - **"NoneOnDryRun"** means that calling the webhook will possibly have side effects, but if the request being reviewed has the dry-run attribute, the side effects will be suppressed. - **"Some"** means that calling the webhook will possibly have side effects. If a request with the dry-run attribute would trigger a call to this webhook, the request will instead fail. - **"Unknown"** means that no information is known about the side effects of calling the webhook. If a request with the dry-run attribute would trigger a call to this webhook, the request will instead fail.
timeoutSeconds	**integer**	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.

4.1.3. .webhooks[].clientConfig

Description

WebhookClientConfig contains the information to make a TLS connection with the webhook

Type

    object

Property	Type	Description
caBundle	**string**	**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.
service	**object**	ServiceReference holds a reference to Service.legacy.k8s.io

Property	Type	Description
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.

4.1.4. .webhooks[].clientConfig.service

Description

ServiceReference holds a reference to Service.legacy.k8s.io

Type

object

Required

- **namespace**

namespace

- **name**

Property	Type	Description
name	**string**	**name** is the name of the service. Required
namespace	**string**	**namespace** is the namespace of the service. Required
path	**string**	**path** is an optional URL path which will be sent in any request to this service.
port	**integer**	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).

4.1.5. .webhooks[].matchConditions

Description

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

Type

array

4.1.6. .webhooks[].matchConditions[]

Description

MatchCondition represents a condition which must by fulfilled for a request to be sent to a webhook.

Type

object

Required

- **name**

- **expression**

Property	Type	Description
expression	**string**	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.

Property	Type	Description
name	string	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.

4.1.7. .webhooks[].rules

Description

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.

Type

  array

4.1.8. .webhooks[].rules[]

Description

RuleWithOperations is a tuple of Operations and Resources. It is recommended to make sure that all the tuple expansions are valid.

Type

  object

Property	Type	Description
apiGroups	array (string)	APIGroups is the API groups the resources belong to. '' **is all groups. If** '' is present, the length of the slice must be one. Required.

Property	Type	Description
apiVersions	**array (string)**	APIVersions is the API versions the resources belong to. '' **is all versions. If '** is present, the length of the slice must be one. Required.
operations	**array (string)**	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.
resources	**array (string)**	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.
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 "*".

4.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/admissionregistration.k8s.io/v1/mutatingwebhookconfigurations**

  - **DELETE**: delete collection of MutatingWebhookConfiguration

  - **GET**: list or watch objects of kind MutatingWebhookConfiguration

  - **POST**: create a MutatingWebhookConfiguration

- **/apis/admissionregistration.k8s.io/v1/watch/mutatingwebhookconfigurations**

  - **GET**: watch individual changes to a list of MutatingWebhookConfiguration. deprecated: use the 'watch' parameter with a list operation instead.

- **/apis/admissionregistration.k8s.io/v1/mutatingwebhookconfigurations/{name}**

  - **DELETE**: delete a MutatingWebhookConfiguration

  - **GET**: read the specified MutatingWebhookConfiguration

  - **PATCH**: partially update the specified MutatingWebhookConfiguration

  - **PUT**: replace the specified MutatingWebhookConfiguration

- **/apis/admissionregistration.k8s.io/v1/watch/mutatingwebhookconfigurations/{name}**

  - **GET**: watch changes to an object of kind MutatingWebhookConfiguration. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

4.2.1. /apis/admissionregistration.k8s.io/v1/mutatingwebhookconfigurations

HTTP method
> **DELETE**

Description
> delete collection of MutatingWebhookConfiguration

Table 4.1. Query parameters

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

Table 4.2. HTTP responses

HTTP code	Reponse body
200 – OK	**Status** schema
401 – Unauthorized	Empty

HTTP method

GET

Description

list or watch objects of kind MutatingWebhookConfiguration

Table 4.3. HTTP responses

HTTP code	Reponse body
200 – OK	**MutatingWebhookConfigurationList** schema
401 – Unauthorized	Empty

HTTP method

POST

Description

create a MutatingWebhookConfiguration

Table 4.4. Query parameters

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

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

Table 4.5. Body parameters

Parameter	Type	Description
body	**MutatingWebhookConfiguration** schema	

Table 4.6. HTTP responses

HTTP code	Reponse body
200 – OK	**MutatingWebhookConfiguration** schema
201 – Created	**MutatingWebhookConfiguration** schema
202 – Accepted	**MutatingWebhookConfiguration** schema
401 – Unauthorized	Empty

4.2.2. /apis/admissionregistration.k8s.io/v1/watch/mutatingwebhookconfigurations

HTTP method

    GET

Description

    watch individual changes to a list of MutatingWebhookConfiguration. deprecated: use the 'watch' parameter with a list operation instead.

Table 4.7. HTTP responses

HTTP code	Reponse body
200 – OK	**WatchEvent** schema
401 – Unauthorized	Empty

4.2.3. /apis/admissionregistration.k8s.io/v1/mutatingwebhookconfigurations/{name}

Table 4.8. Global path parameters

Parameter	Type	Description
name	**string**	name of the MutatingWebhookConfiguration

HTTP method

DELETE

Description

delete a MutatingWebhookConfiguration

Table 4.9. Query parameters

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

Table 4.10. HTTP responses

HTTP code	Reponse body
200 – OK	**Status** schema
202 – Accepted	**Status** schema
401 – Unauthorized	Empty

HTTP method

GET

Description

read the specified MutatingWebhookConfiguration

Table 4.11. HTTP responses

HTTP code	Reponse body
200 - OK	**MutatingWebhookConfiguration** schema
401 – Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified MutatingWebhookConfiguration

Table 4.12. Query parameters

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

Table 4.13. HTTP responses

HTTP code	Reponse body
200 - OK	**MutatingWebhookConfiguration** schema
201 – Created	**MutatingWebhookConfiguration** schema

HTTP code	Reponse body
401 – Unauthorized	Empty

HTTP method

PUT

Description

replace the specified MutatingWebhookConfiguration

Table 4.14. Query parameters

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

Table 4.15. Body parameters

Parameter	Type	Description
body	**MutatingWebhookConfiguration** schema	

Table 4.16. HTTP responses

HTTP code	Reponse body
200 – OK	**MutatingWebhookConfiguration** schema
201 – Created	**MutatingWebhookConfiguration** schema
401 – Unauthorized	Empty

4.2.4. /apis/admissionregistration.k8s.io/v1/watch/mutatingwebhookconfigurations/

Table 4.17. Global path parameters

Parameter	Type	Description
name	**string**	name of the MutatingWebhookConfiguration

HTTP method

    GET

Description

    watch changes to an object of kind MutatingWebhookConfiguration. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

Table 4.18. HTTP responses

HTTP code	Reponse body
200 – OK	**WatchEvent** schema
401 – Unauthorized	Empty

CHAPTER 5. VALIDATINGADMISSIONPOLICY [ADMISSIONREGISTRATION.K8S.IO/V1]

Description

ValidatingAdmissionPolicy describes the definition of an admission validation policy that accepts or rejects an object without changing it.

Type

object

5.1. SPECIFICATION

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

5.1.1. .spec

Description

ValidatingAdmissionPolicySpec is the specification of the desired behavior of the AdmissionPolicy.

Type

object

Property	Type	Description
auditAnnotations	**array**	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; a least one of validations or auditAnnotations is required.
auditAnnotations[]	**object**	AuditAnnotation describes how to produce an audit annotation for an API request.
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. Possible enum values: - **"Fail"** means that an error calling the webhook causes the admission to fail. - **"Ignore"** means that an error calling the webhook is ignored.

Property	Type	Description

Property	Type	Description
matchConditions	**array**	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
matchConditions[]	**object**	MatchCondition represents a condition which must by fulfilled for a request to be sent to a webhook.
matchConstraints	**object**	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)
paramKind	**object**	ParamKind is a tuple of Group Kind and Version.

Property	Type	Description
validations	**array**	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.
validations[]	**object**	Validation specifies the CEL expression which is used to apply the validation.
variables	**array**	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.
variables[]	**object**	Variable is the definition of a variable that is used for composition. A variable is defined as a named expression.

5.1.2. .spec.auditAnnotations

Description

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; a least one of validations or auditAnnotations is required.

Type

array

5.1.3. .spec.auditAnnotations[]

Description

AuditAnnotation describes how to produce an audit annotation for an API request.

Type

object

Required

- **key**

- **valueExpression**

Property	Type	Description
key	**string**	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.

Property	Type	Description
valueExpression	**string**	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.

5.1.4. .spec.matchConditions

Description

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

Type

array

5.1.5. .spec.matchConditions[]

Description

MatchCondition represents a condition which must by fulfilled for a request to be sent to a webhook.

Type

object

Required

- **name**

- **expression**

Property	Type	Description
expression	**string**	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.

Property	Type	Description
name	**string**	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.

5.1.6. .spec.matchConstraints

Description

> 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)

Type

> **object**

Property	Type	Description
excludeResourceRules	**array**	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)
excludeResourceRules[]	**object**	NamedRuleWithOperations is a tuple of Operations and Resources with ResourceNames.

Property	Type	Description
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" Possible enum values: – **"Equivalent"** means requests should be sent to the webhook if they modify a resource listed in rules via another API group or version. – **"Exact"** means requests should only be sent to the webhook if they exactly match a given rule.

Property	Type	Description
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.

Property	Type	Description
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.
resourceRules	**array**	ResourceRules describes what operations on what resources/subresources the ValidatingAdmissionPolicy matches. The policy cares about an operation if it matches *any* Rule.
resourceRules[]	**object**	NamedRuleWithOperations is a tuple of Operations and Resources with ResourceNames.

5.1.7. .spec.matchConstraints.excludeResourceRules

Description

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)

Type

array

5.1.8. .spec.matchConstraints.excludeResourceRules[]

Description

NamedRuleWithOperations is a tuple of Operations and Resources with ResourceNames.

Type

object

Property	Type	Description
apiGroups	**array (string)**	APIGroups is the API groups the resources belong to. '' **is all groups. If** '' is present, the length of the slice must be one. Required.
apiVersions	**array (string)**	APIVersions is the API versions the resources belong to. '' **is all versions. If** '' is present, the length of the slice must be one. Required.
operations	**array (string)**	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.
resourceNames	**array (string)**	ResourceNames is an optional white list of names that the rule applies to. An empty set means that everything is allowed.
resources	**array (string)**	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.

Property	Type	Description
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 "*".

5.1.9. .spec.matchConstraints.resourceRules

Description

ResourceRules describes what operations on what resources/subresources the ValidatingAdmissionPolicy matches. The policy cares about an operation if it matches *any* Rule.

Type

array

5.1.10. .spec.matchConstraints.resourceRules[]

Description

NamedRuleWithOperations is a tuple of Operations and Resources with ResourceNames.

Type

object

Property	Type	Description
apiGroups	**array (string)**	APIGroups is the API groups the resources belong to. '' **is all groups. If** '' is present, the length of the slice must be one. Required.
apiVersions	**array (string)**	APIVersions is the API versions the resources belong to. '' **is all versions. If** '' is present, the length of the slice must be one. Required.

Property	Type	Description
operations	**array (string)**	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.
resourceNames	**array (string)**	ResourceNames is an optional white list of names that the rule applies to. An empty set means that everything is allowed.
resources	**array (string)**	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.
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 "*".

5.1.11. .spec.paramKind

Description

ParamKind is a tuple of Group Kind and Version.

Type

object

Property	Type	Description
apiVersion	**string**	APIVersion is the API group version the resources belong to. In format of "group/version". Required.
kind	**string**	Kind is the API kind the resources belong to. Required.

5.1.12. .spec.validations

Description

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.

Type

array

5.1.13. .spec.validations[]

Description

Validation specifies the CEL expression which is used to apply the validation.

Type

object

Required

- **expression**

Property	Type	Description
expression	**string**	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

Property	Type	Description
		null for DELETE requests. – 'oldObject' – The existing object. The value is null for CREATE requests. – 'request' – Attributes of the API request([ref](/pkg/apis/admission/types.go#AdmissionRequest)). – '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*

Property	Type	Description
		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.
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}".

Property	Type	Description
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)")"
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.

5.1.14. .spec.variables

Description

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.

Type

array

5.1.15. .spec.variables[]

Description

Variable is the definition of a variable that is used for composition. A variable is defined as a named expression.

Type

object

Required

- **name**

- **expression**

Property	Type	Description
expression	**string**	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.
name	**string**	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**

5.1.16. .status

Description

ValidatingAdmissionPolicyStatus represents the status of an admission validation policy.

Type

object

Property	Type	Description
conditions	**array (Condition)**	The conditions represent the latest available observations of a policy's current state.
observedGeneration	**integer**	The generation observed by the controller.
typeChecking	**object**	TypeChecking contains results of type checking the expressions in the ValidatingAdmissionPolicy

5.1.17. .status.typeChecking

Description

TypeChecking contains results of type checking the expressions in the ValidatingAdmissionPolicy

Type

object

Property	Type	Description
expressionWarnings	**array**	The type checking warnings for each expression.
expressionWarnings[]	**object**	ExpressionWarning is a warning information that targets a specific expression.

5.1.18. .status.typeChecking.expressionWarnings

Description

The type checking warnings for each expression.

Type

array

5.1.19. .status.typeChecking.expressionWarnings[]

Description

ExpressionWarning is a warning information that targets a specific expression.

Type

object

Required

- **fieldRef**

- **warning**

Property	Type	Description
fieldRef	**string**	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"
warning	**string**	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.

5.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/admissionregistration.k8s.io/v1/validatingadmissionpolicies**

  - **DELETE**: delete collection of ValidatingAdmissionPolicy

  - **GET**: list or watch objects of kind ValidatingAdmissionPolicy

  - **POST**: create a ValidatingAdmissionPolicy

- **/apis/admissionregistration.k8s.io/v1/watch/validatingadmissionpolicies**

  - **GET**: watch individual changes to a list of ValidatingAdmissionPolicy. deprecated: use the 'watch' parameter with a list operation instead.

- **/apis/admissionregistration.k8s.io/v1/validatingadmissionpolicies/{name}**

  - **DELETE**: delete a ValidatingAdmissionPolicy

  - **GET**: read the specified ValidatingAdmissionPolicy

  - **PATCH**: partially update the specified ValidatingAdmissionPolicy

  - **PUT**: replace the specified ValidatingAdmissionPolicy

- **/apis/admissionregistration.k8s.io/v1/watch/validatingadmissionpolicies/{name}**

  - **GET**: watch changes to an object of kind ValidatingAdmissionPolicy. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

- **/apis/admissionregistration.k8s.io/v1/validatingadmissionpolicies/{name}/status**

  - **GET**: read status of the specified ValidatingAdmissionPolicy

  - **PATCH**: partially update status of the specified ValidatingAdmissionPolicy

- **PUT**: replace status of the specified ValidatingAdmissionPolicy

5.2.1. /apis/admissionregistration.k8s.io/v1/validatingadmissionpolicies

HTTP method
    DELETE
Description
    delete collection of ValidatingAdmissionPolicy

Table 5.1. Query parameters

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

Table 5.2. HTTP responses

HTTP code	Reponse body
200 - OK	**Status** schema
401 - Unauthorized	Empty

HTTP method
    GET
Description
    list or watch objects of kind ValidatingAdmissionPolicy

Table 5.3. HTTP responses

HTTP code	Reponse body
200 - OK	**ValidatingAdmissionPolicyList** schema
401 - Unauthorized	Empty

HTTP method
    POST
Description
    create a ValidatingAdmissionPolicy

Table 5.4. Query parameters

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

Table 5.5. Body parameters

Parameter	Type	Description
body	**ValidatingAdmission Policy** schema	

Table 5.6. HTTP responses

HTTP code	Reponse body
200 – OK	**ValidatingAdmissionPolicy** schema
201 – Created	**ValidatingAdmissionPolicy** schema
202 – Accepted	**ValidatingAdmissionPolicy** schema
401 – Unauthorized	Empty

5.2.2. /apis/admissionregistration.k8s.io/v1/watch/validatingadmissionpolicies

HTTP method

GET

Description

watch individual changes to a list of ValidatingAdmissionPolicy. deprecated: use the 'watch' parameter with a list operation instead.

Table 5.7. HTTP responses

HTTP code	Reponse body
200 – OK	**WatchEvent** schema
401 – Unauthorized	Empty

5.2.3. /apis/admissionregistration.k8s.io/v1/validatingadmissionpolicies/{name}

Table 5.8. Global path parameters

Parameter	Type	Description
name	**string**	name of the ValidatingAdmissionPolicy

HTTP method

DELETE

Description

delete a ValidatingAdmissionPolicy

Table 5.9. Query parameters

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

Table 5.10. HTTP responses

HTTP code	Reponse body
200 – OK	**Status** schema
202 – Accepted	**Status** schema
401 – Unauthorized	Empty

HTTP method

> **GET**

Description

> read the specified ValidatingAdmissionPolicy

Table 5.11. HTTP responses

HTTP code	Reponse body
200 – OK	**ValidatingAdmissionPolicy** schema
401 – Unauthorized	Empty

HTTP method

> **PATCH**

Description

> partially update the specified ValidatingAdmissionPolicy

Table 5.12. Query parameters

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

Table 5.13. HTTP responses

HTTP code	Reponse body
200 – OK	**ValidatingAdmissionPolicy** schema
201 – Created	**ValidatingAdmissionPolicy** schema
401 – Unauthorized	Empty

HTTP method

PUT

Description

replace the specified ValidatingAdmissionPolicy

Table 5.14. Query parameters

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

Table 5.15. Body parameters

Parameter	Type	Description
body	**ValidatingAdmission Policy** schema	

Table 5.16. HTTP responses

HTTP code	Reponse body
200 - OK	**ValidatingAdmissionPolicy** schema
201 - Created	**ValidatingAdmissionPolicy** schema
401 - Unauthorized	Empty

5.2.4. /apis/admissionregistration.k8s.io/v1/watch/validatingadmissionpolicies/{name

Table 5.17. Global path parameters

Parameter	Type	Description
name	**string**	name of the ValidatingAdmissionPolicy

HTTP method
GET

Description
watch changes to an object of kind ValidatingAdmissionPolicy. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

Table 5.18. HTTP responses

HTTP code	Reponse body
200 - OK	**WatchEvent** schema
401 - Unauthorized	Empty

5.2.5. /apis/admissionregistration.k8s.io/v1/validatingadmissionpolicies/{name}/statu

Table 5.19. Global path parameters

Parameter	Type	Description
name	**string**	name of the ValidatingAdmissionPolicy

HTTP method
GET

Description
read status of the specified ValidatingAdmissionPolicy

Table 5.20. HTTP responses

HTTP code	Reponse body
200 - OK	**ValidatingAdmissionPolicy** schema
401 – Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified ValidatingAdmissionPolicy

Table 5.21. Query parameters

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

Table 5.22. HTTP responses

HTTP code	Reponse body
200 - OK	**ValidatingAdmissionPolicy** schema
201 – Created	**ValidatingAdmissionPolicy** schema

HTTP code	Reponse body
401 – Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified ValidatingAdmissionPolicy

Table 5.23. Query parameters

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

Table 5.24. Body parameters

Parameter	Type	Description
body	**ValidatingAdmissionPolicy** schema	

Table 5.25. HTTP responses

HTTP code	Reponse body
200 – OK	**ValidatingAdmissionPolicy** schema
201 – Created	**ValidatingAdmissionPolicy** schema
401 – Unauthorized	Empty

CHAPTER 6. VALIDATINGADMISSIONPOLICYBINDING [ADMISSIONREGISTRATION.K8S.IO/V1]

Description

ValidatingAdmissionPolicyBinding binds the ValidatingAdmissionPolicy with paramerized 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.

Type

object

6.1. SPECIFICATION

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

Property	Type	Description
metadata	**ObjectMeta**	Standard object metadata; More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata.
spec	**object**	ValidatingAdmissionPolicyBinding Spec is the specification of the ValidatingAdmissionPolicyBinding.

6.1.1. .spec

Description

ValidatingAdmissionPolicyBindingSpec is the specification of the ValidatingAdmissionPolicyBinding.

Type

object

Property	Type	Description
matchResources	**object**	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)
paramRef	**object**	ParamRef describes how to locate the params to be used as input to expressions of rules applied by a policy binding.
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.
validationActions	**array (string)**	validationActions declares how Validations of the referenced ValidatingAdmissionPolicy are

Property	Type	Description
		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.** |

Property	Type	Description
		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.

6.1.2. .spec.matchResources

Description

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)

Type

object

Property	Type	Description
excludeResourceRules	**array**	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)
excludeResourceRules[]	**object**	NamedRuleWithOperations is a tuple of Operations and Resources with ResourceNames.

Property	Type	Description
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" Possible enum values: – **"Equivalent"** means requests should be sent to the webhook if they modify a resource listed in rules via another API group or version. – **"Exact"** means requests should only be sent to the webhook if they exactly match a given rule.

Property	Type	Description
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.

Property	Type	Description
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.
resourceRules	**array**	ResourceRules describes what operations on what resources/subresources the ValidatingAdmissionPolicy matches. The policy cares about an operation if it matches *any* Rule.
resourceRules[]	**object**	NamedRuleWithOperations is a tuple of Operations and Resources with ResourceNames.

6.1.3. .spec.matchResources.excludeResourceRules

Description

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)

Type

array

6.1.4. .spec.matchResources.excludeResourceRules[]

Description

NamedRuleWithOperations is a tuple of Operations and Resources with ResourceNames.

Type

    object

Property	Type	Description
apiGroups	**array (string)**	APIGroups is the API groups the resources belong to. '**' is all groups. If '**' is present, the length of the slice must be one. Required.
apiVersions	**array (string)**	APIVersions is the API versions the resources belong to. '**' is all versions. If '**' is present, the length of the slice must be one. Required.
operations	**array (string)**	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.
resourceNames	**array (string)**	ResourceNames is an optional white list of names that the rule applies to. An empty set means that everything is allowed.
resources	**array (string)**	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.

Property	Type	Description
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 "*".

6.1.5. .spec.matchResources.resourceRules

Description

ResourceRules describes what operations on what resources/subresources the ValidatingAdmissionPolicy matches. The policy cares about an operation if it matches *any* Rule.

Type

array

6.1.6. .spec.matchResources.resourceRules[]

Description

NamedRuleWithOperations is a tuple of Operations and Resources with ResourceNames.

Type

object

Property	Type	Description
apiGroups	**array (string)**	APIGroups is the API groups the resources belong to. '' **is all groups. If** '' is present, the length of the slice must be one. Required.
apiVersions	**array (string)**	APIVersions is the API versions the resources belong to. '' **is all versions. If** '' is present, the length of the slice must be one. Required.

Property	Type	Description
operations	**array (string)**	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.
resourceNames	**array (string)**	ResourceNames is an optional white list of names that the rule applies to. An empty set means that everything is allowed.
resources	**array (string)**	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.
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 "*".

6.1.7. .spec.paramRef

Description

ParamRef describes how to locate the params to be used as input to expressions of rules applied by a policy binding.

Type

object

Property	Type	Description
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.
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.

Property	Type	Description
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
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.

6.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/admissionregistration.k8s.io/v1/validatingadmissionpolicybindings**

  - **DELETE**: delete collection of ValidatingAdmissionPolicyBinding

  - **GET**: list or watch objects of kind ValidatingAdmissionPolicyBinding

  - **POST**: create a ValidatingAdmissionPolicyBinding

- **/apis/admissionregistration.k8s.io/v1/watch/validatingadmissionpolicybindings**

  - **GET**: watch individual changes to a list of ValidatingAdmissionPolicyBinding. deprecated: use the 'watch' parameter with a list operation instead.

- **/apis/admissionregistration.k8s.io/v1/validatingadmissionpolicybindings/{name}**

  - **DELETE**: delete a ValidatingAdmissionPolicyBinding

  - **GET**: read the specified ValidatingAdmissionPolicyBinding

  - **PATCH**: partially update the specified ValidatingAdmissionPolicyBinding

  - **PUT**: replace the specified ValidatingAdmissionPolicyBinding

- **/apis/admissionregistration.k8s.io/v1/watch/validatingadmissionpolicybindings/{name}**

  - **GET**: watch changes to an object of kind ValidatingAdmissionPolicyBinding. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

6.2.1. /apis/admissionregistration.k8s.io/v1/validatingadmissionpolicybindings

HTTP method

    DELETE

Description

    delete collection of ValidatingAdmissionPolicyBinding

Table 6.1. Query parameters

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

Table 6.2. HTTP responses

HTTP code	Reponse body
200 - OK	**Status** schema
401 - Unauthorized	Empty

HTTP method

    GET

Description

    list or watch objects of kind ValidatingAdmissionPolicyBinding

Table 6.3. HTTP responses

HTTP code	Reponse body
200 - OK	**ValidatingAdmissionPolicyBindingList** schema
401 - Unauthorized	Empty

HTTP method

  POST

Description

  create a ValidatingAdmissionPolicyBinding

Table 6.4. Query parameters

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

Table 6.5. Body parameters

Parameter	Type	Description
body	**ValidatingAdmission PolicyBinding** schema	

Table 6.6. HTTP responses

HTTP code	Reponse body
200 – OK	**ValidatingAdmissionPolicyBinding** schema
201 – Created	**ValidatingAdmissionPolicyBinding** schema
202 – Accepted	**ValidatingAdmissionPolicyBinding** schema
401 – Unauthorized	Empty

6.2.2. /apis/admissionregistration.k8s.io/v1/watch/validatingadmissionpolicybindings

HTTP method

GET

Description

watch individual changes to a list of ValidatingAdmissionPolicyBinding. deprecated: use the 'watch' parameter with a list operation instead.

Table 6.7. HTTP responses

HTTP code	Reponse body
200 – OK	**WatchEvent** schema
401 – Unauthorized	Empty

6.2.3. /apis/admissionregistration.k8s.io/v1/validatingadmissionpolicybindings/{name

Table 6.8. Global path parameters

Parameter	Type	Description
name	**string**	name of the ValidatingAdmissionPolicyBinding

HTTP method

DELETE

Description

delete a ValidatingAdmissionPolicyBinding

Table 6.9. Query parameters

Parameter	Type	Description

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

Table 6.10. HTTP responses

HTTP code	Reponse body
200 – OK	**Status** schema
202 – Accepted	**Status** schema
401 – Unauthorized	Empty

HTTP method

GET

Description

read the specified ValidatingAdmissionPolicyBinding

Table 6.11. HTTP responses

HTTP code	Reponse body
200 – OK	**ValidatingAdmissionPolicyBinding** schema
401 – Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified ValidatingAdmissionPolicyBinding

Table 6.12. Query parameters

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

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

Table 6.13. HTTP responses

HTTP code	Reponse body
200 – OK	**ValidatingAdmissionPolicyBinding** schema
201 – Created	**ValidatingAdmissionPolicyBinding** schema
401 – Unauthorized	Empty

HTTP method
  PUT
Description
  replace the specified ValidatingAdmissionPolicyBinding

Table 6.14. Query parameters

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

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

Table 6.15. Body parameters

Parameter	Type	Description
body	**ValidatingAdmissionPolicyBinding** schema	

Table 6.16. HTTP responses

HTTP code	Reponse body
200 – OK	**ValidatingAdmissionPolicyBinding** schema
201 – Created	**ValidatingAdmissionPolicyBinding** schema
401 – Unauthorized	Empty

6.2.4. /apis/admissionregistration.k8s.io/v1/watch/validatingadmissionpolicybindings/

Table 6.17. Global path parameters

Parameter	Type	Description
name	**string**	name of the ValidatingAdmissionPolicyBinding

HTTP method

GET

Description

watch changes to an object of kind ValidatingAdmissionPolicyBinding. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

Table 6.18. HTTP responses

HTTP code	Reponse body
200 - OK	**WatchEvent** schema
401 - Unauthorized	Empty

CHAPTER 7. VALIDATINGWEBHOOKCONFIGURATION [ADMISSIONREGISTRATION.K8S.IO/V1]

Description

ValidatingWebhookConfiguration describes the configuration of and admission webhook that accept or reject and object without changing it.

Type

object

7.1. SPECIFICATION

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

Property	Type	Description
webhooks[]	**object**	ValidatingWebhook describes an admission webhook and the resources and operations it applies to.

7.1.1. .webhooks

Description

Webhooks is a list of webhooks and the affected resources and operations.

Type

array

7.1.2. .webhooks[]

Description

ValidatingWebhook describes an admission webhook and the resources and operations it applies to.

Type

object

Required

- **name**

- **clientConfig**

- **sideEffects**

- **admissionReviewVersions**

Property	Type	Description
admissionReviewVersions	**array (string)**	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.

Property	Type	Description
clientConfig	**object**	WebhookClientConfig contains the information to make a TLS connection with the webhook
failurePolicy	**string**	FailurePolicy defines how unrecognized errors from the admission endpoint are handled – allowed values are Ignore or Fail. Defaults to Fail. Possible enum values: – **"Fail"** means that an error calling the webhook causes the admission to fail. – **"Ignore"** means that an error calling the webhook is ignored.
matchConditions	**array**	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
matchConditions[]	**object**	MatchCondition represents a condition which must by fulfilled for a request to be sent to a webhook.

Property	Type	Description
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" Possible enum values: – **"Equivalent"** means requests should be sent to the webhook if they modify a resource listed in rules via another API group or version. – **"Exact"** means requests should only be sent to the webhook if they exactly match a given rule.

Property	Type	Description
name	**string**	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.
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.

Property	Type	Description
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.
rules	**array**	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.
rules[]	**object**	RuleWithOperations is a tuple of Operations and Resources. It is recommended to make sure that all the tuple expansions are valid.

Property	Type	Description
sideEffects	**string**	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. Possible enum values: - **"None"** means that calling the webhook will have no side effects. - **"NoneOnDryRun"** means that calling the webhook will possibly have side effects, but if the request being reviewed has the dry-run attribute, the side effects will be suppressed. - **"Some"** means that calling the webhook will possibly have side effects. If a request with the dry-run attribute would trigger a call to this webhook, the request will instead fail. - **"Unknown"** means that no information is known about the side effects of calling the webhook. If a request with the dry-run attribute would trigger a call to this webhook, the request will instead fail.
timeoutSeconds	**integer**	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.

7.1.3. .webhooks[].clientConfig

Description

WebhookClientConfig contains the information to make a TLS connection with the webhook

Type

    object

Property	Type	Description
caBundle	**string**	**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.
service	**object**	ServiceReference holds a reference to Service.legacy.k8s.io

Property	Type	Description
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.

7.1.4. .webhooks[].clientConfig.service

Description

ServiceReference holds a reference to Service.legacy.k8s.io

Type

    object

Required

- **namespace**

- **name**

Property	Type	Description
name	**string**	**name** is the name of the service. Required
namespace	**string**	**namespace** is the namespace of the service. Required
path	**string**	**path** is an optional URL path which will be sent in any request to this service.
port	**integer**	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).

7.1.5. .webhooks[].matchConditions

Description

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

Type

array

7.1.6. .webhooks[].matchConditions[]

Description

MatchCondition represents a condition which must by fulfilled for a request to be sent to a webhook.

Type

object

Required

- **name**

- **expression**

Property	Type	Description
expression	**string**	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.

Property	Type	Description
name	**string**	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.

7.1.7. .webhooks[].rules

Description

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.

Type

array

7.1.8. .webhooks[].rules[]

Description

RuleWithOperations is a tuple of Operations and Resources. It is recommended to make sure that all the tuple expansions are valid.

Type

object

Property	Type	Description

Property	Type	Description
apiGroups	**array (string)**	APIGroups is the API groups the resources belong to. " **is all groups. If** '' is present, the length of the slice must be one. Required.
apiVersions	**array (string)**	APIVersions is the API versions the resources belong to. " **is all versions. If** " is present, the length of the slice must be one. Required.
operations	**array (string)**	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.
resources	**array (string)**	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.

Property	Type	Description
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 "*".

7.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/admissionregistration.k8s.io/v1/validatingwebhookconfigurations**

    - **DELETE**: delete collection of ValidatingWebhookConfiguration

    - **GET**: list or watch objects of kind ValidatingWebhookConfiguration

    - **POST**: create a ValidatingWebhookConfiguration

- **/apis/admissionregistration.k8s.io/v1/watch/validatingwebhookconfigurations**

    - **GET**: watch individual changes to a list of ValidatingWebhookConfiguration. deprecated: use the 'watch' parameter with a list operation instead.

- **/apis/admissionregistration.k8s.io/v1/validatingwebhookconfigurations/{name}**

    - **DELETE**: delete a ValidatingWebhookConfiguration

    - **GET**: read the specified ValidatingWebhookConfiguration

    - **PATCH**: partially update the specified ValidatingWebhookConfiguration

    - **PUT**: replace the specified ValidatingWebhookConfiguration

- **/apis/admissionregistration.k8s.io/v1/watch/validatingwebhookconfigurations/{name}**

    - **GET**: watch changes to an object of kind ValidatingWebhookConfiguration. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

7.2.1. /apis/admissionregistration.k8s.io/v1/validatingwebhookconfigurations

HTTP method
    DELETE
Description

delete collection of ValidatingWebhookConfiguration

Table 7.1. Query parameters

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

Table 7.2. HTTP responses

HTTP code	Reponse body
200 – OK	**Status** schema
401 – Unauthorized	Empty

HTTP method

GET

Description

list or watch objects of kind ValidatingWebhookConfiguration

Table 7.3. HTTP responses

HTTP code	Reponse body
200 – OK	**ValidatingWebhookConfigurationList** schema
401 – Unauthorized	Empty

HTTP method

POST

Description

create a ValidatingWebhookConfiguration

Table 7.4. Query parameters

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

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

Table 7.5. Body parameters

Parameter	Type	Description
body	**ValidatingWebhookConfiguration** schema	

Table 7.6. HTTP responses

HTTP code	Reponse body
200 – OK	**ValidatingWebhookConfiguration** schema
201 – Created	**ValidatingWebhookConfiguration** schema
202 – Accepted	**ValidatingWebhookConfiguration** schema
401 – Unauthorized	Empty

7.2.2. /apis/admissionregistration.k8s.io/v1/watch/validatingwebhookconfigurations

HTTP method
GET
Description

watch individual changes to a list of ValidatingWebhookConfiguration. deprecated: use the 'watch' parameter with a list operation instead.

Table 7.7. HTTP responses

HTTP code	Reponse body
200 - OK	**WatchEvent** schema
401 - Unauthorized	Empty

7.2.3. /apis/admissionregistration.k8s.io/v1/validatingwebhookconfigurations/{name}

Table 7.8. Global path parameters

Parameter	Type	Description
name	**string**	name of the ValidatingWebhookConfiguration

HTTP method
> DELETE

Description
> delete a ValidatingWebhookConfiguration

Table 7.9. Query parameters

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

Table 7.10. HTTP responses

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

HTTP method
> GET

Description

read the specified ValidatingWebhookConfiguration

Table 7.11. HTTP responses

HTTP code	Reponse body
200 - OK	**ValidatingWebhookConfiguration** schema
401 – Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified ValidatingWebhookConfiguration

Table 7.12. Query parameters

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

Table 7.13. HTTP responses

HTTP code	Reponse body
200 – OK	**ValidatingWebhookConfiguration** schema

HTTP code	Reponse body
201 – Created	**ValidatingWebhookConfiguration** schema
401 – Unauthorized	Empty

HTTP method

> **PUT**

Description

> replace the specified ValidatingWebhookConfiguration

Table 7.14. Query parameters

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

Table 7.15. Body parameters

Parameter	Type	Description
body	**ValidatingWebhookConfiguration** schema	

Table 7.16. HTTP responses

HTTP code	Reponse body
200 – OK	**ValidatingWebhookConfiguration** schema
201 – Created	**ValidatingWebhookConfiguration** schema
401 – Unauthorized	Empty

7.2.4. /apis/admissionregistration.k8s.io/v1/watch/validatingwebhookconfigurations/

Table 7.17. Global path parameters

Parameter	Type	Description
name	**string**	name of the ValidatingWebhookConfiguration

HTTP method

    GET

Description

    watch changes to an object of kind ValidatingWebhookConfiguration. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

Table 7.18. HTTP responses

HTTP code	Reponse body
200 – OK	**WatchEvent** schema
401 – Unauthorized	Empty