Operator APIs

Operator [operators.operators.coreos.com/v1]

Operator [operators.operators.coreos.com/v1]

Description

Operator represents a cluster operator.

Type

`object`

Specification

Property	Type	Description
`apiVersion`	`string`	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗

Property	Type	Description
`kind`	`string`	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`metadata`	`ObjectMeta`	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
`spec`	`object`	OperatorSpec defines the desired state of Operator
`status`	`object`	OperatorStatus defines the observed state of an Operator and its components

.spec

Description

OperatorSpec defines the desired state of Operator

Type

`object`

.status

Description

OperatorStatus defines the observed state of an Operator and its components

Type

`object`

Property	Type	Description
`components`	`object`	Components describes resources that compose the operator.

.status.components

Description

Components describes resources that compose the operator.

Type

`object`

Required

`labelSelector`

Property	Type	Description
`labelSelector`	`object`	LabelSelector is a label query over a set of resources used to select the operator's components
`refs`	`array`	Refs are a set of references to the operator's component resources, selected with LabelSelector.

.status.components.labelSelector

Description

LabelSelector is a label query over a set of resources used to select the operator's components

Type

`object`

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

.status.components.labelSelector.matchExpressions

Description

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

Type

`array`

.status.components.labelSelector.matchExpressions[]

Description

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

Type

`object`

Required

`key` `operator`

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

.status.components.labelSelector.matchExpressions[].values

Description

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

Type

`array`

.status.components.labelSelector.matchExpressions[].values[]

Type

`string`

.status.components.labelSelector.matchLabels

Description

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

Type

`object`

.status.components.refs

Description

Refs are a set of references to the operator's component resources, selected with LabelSelector.

Type

`array`

.status.components.refs[]

Description

RichReference is a reference to a resource, enriched with its status conditions.

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`conditions`	`array`	Conditions represents the latest state of the component.
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field

Property	Type	Description
		access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object. TODO: this design is not final and this field is subject to change in the future.
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗

Property	Type	Description
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.status.components.refs[].conditions

Description

Conditions represents the latest state of the component.

Type

`array`

.status.components.refs[].conditions[]

Description

Condition represent the latest available observations of an component's state.

Type

`object`

Required

`status` `type`

Property	Type	Description
`lastTransitionTime`	`string`	Last time the condition transitioned from one status to another.
`lastUpdateTime`	`string`	Last time the condition was probed
`message`	`string`	A human readable message indicating details about the transition.
`reason`	`string`	The reason for the condition's last transition.
`status`	`string`	Status of the condition, one of True, False, Unknown.
`type`	`string`	Type of condition.

API Endpoints

The following API endpoints are available:

- `/kubernetes/{cluster}/apis/operators.coreos.com/v1/operators`
  - `DELETE`: delete collection of Operator
  - `GET`: list objects of kind Operator
  - `POST`: create a new Operator
- `/kubernetes/{cluster}/apis/operators.coreos.com/v1/operators/{name}`

- `DELETE`: delete the specified Operator
- `GET`: read the specified Operator
- `PATCH`: partially update the specified Operator
- `PUT`: replace the specified Operator

- `/kubernetes/{cluster}/apis/operators.coreos.com/v1/operators/{name}/status`
  - `GET`: read status of the specified Operator
  - `PATCH`: partially update status of the specified Operator
  - `PUT`: replace status of the specified Operator

/kubernetes/{cluster}/apis/operators.coreos.com/v1/operators

HTTP method

`DELETE`

Description

delete collection of Operator

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind Operator

HTTP responses

HTTP code	Response body
200 - OK	`OperatorList` schema
401 - Unauthorized	Empty

HTTP method

`POST`

Description

create a new Operator

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.

Body parameters

Parameter	Type	Description
`body`	`Operator` schema	`application/json` formatted

HTTP responses

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

/kubernetes/{cluster}/apis/operators.coreos.com/v1/operators/{name}

HTTP method

`DELETE`

Description

delete the specified Operator

Query parameters

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

HTTP responses

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

HTTP method

`GET`

Description

read the specified Operator

HTTP responses

HTTP code	Response body
200 - OK	`Operator` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified Operator

Query parameters

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

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

HTTP responses

HTTP code	Response body
200 - OK	`Operator` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified Operator

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.

Body parameters

Parameter	Type	Description
`body`	`Operator` schema	`application/json` formatted

HTTP responses

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

/kubernetes/{cluster}/apis/operators.coreos.com/v1/operators/{name}/status

HTTP method

`GET`

Description

read status of the specified Operator

HTTP responses

HTTP code	Response body
200 - OK	`Operator` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified Operator

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

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

HTTP responses

HTTP code	Response body
200 - OK	`Operator` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified Operator

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

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

Body parameters

Parameter	Type	Description
`body`	`Operator` schema	`application/json` formatted

HTTP responses

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