

# EtcD APIs

---

[EtcDCluster \[operator.etcd.io/v1alpha1\]](#)

# EtcdCluster [operator.etcd.io/v1alpha1]

## Description

EtcdCluster is the Schema for the etcdclusters API.

## Type

object

## Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	<p>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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a></p>

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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
metadata	ObjectMeta	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	EtcdClusterSpec defines the desired state of EtcdCluster.
status	object	EtcdClusterStatus defines the observed state of EtcdCluster.

## .spec

### Description

EtcdClusterSpec defines the desired state of EtcdCluster.

### Type

object

### Required

size version

Property	Type	Description
<code>etcdOptions</code>	<code>array</code>	etcd configuration options are passed as command line arguments to the etcd container, refer to etcd documentation for configuration options applicable for the version of etcd being used.
<code>imageRegistry</code>	<code>string</code>	ImageRegistry specifies the container registry that hosts the etcd images. If unset, it defaults to the value provided via the controller's <code>--image-registry</code> flag, which itself defaults to <code>"gcr.io/etcd-development/etcd"</code> .
<code>podTemplate</code>	<code>object</code>	PodTemplate is the pod template to use for the etcd cluster.
<code>size</code>	<code>integer</code>	Size is the expected size of the etcd cluster.
<code>storageSpec</code>	<code>object</code>	StorageSpec is the name of the StorageSpec to use for the etcd cluster. If not provided, then each POD just uses the temporary storage inside the container.
<code>tls</code>	<code>object</code>	TLS is the TLS certificate configuration to use for the etcd cluster and etcd operator.
<code>version</code>	<code>string</code>	Version is the expected version of the etcd container image.

## .spec.etcdOptions

### Description

etcd configuration options are passed as command line arguments to the etcd container, refer to etcd documentation for configuration options applicable for the version of etcd being used.

### Type

array

## .spec.etcdOptions[]

### Type

string

## .spec.podTemplate

### Description

PodTemplate is the pod template to use for the etcd cluster.

### Type

object

Property	Type	Description
metadata	ObjectMeta	Metadata is the metadata to add to the pod.

## .spec.storageSpec

### Description

StorageSpec is the name of the StorageSpec to use for the etcd cluster. If not provided, then each POD just uses the temporary storage inside the container.

### Type

`object`

## Required

`volumeSizeRequest`

Property	Type	Description
<code>accessModes</code>	<code>string</code>	
<code>pvcName</code>	<code>string</code>	
<code>storageClassName</code>	<code>string</code>	
<code>volumeSizeLimit</code>	<code>int64</code>	
<code>volumeSizeRequest</code>	<code>int64</code>	

## .spec.tls

### Description

TLS is the TLS certificate configuration to use for the etcd cluster and etcd operator.

### Type

`object`

Property	Type	Description
<code>provider</code>	<code>string</code>	
<code>providerCfg</code>	<code>object</code>	

## .spec.tls.providerCfg

### Type

`object`

Property	Type	Description
autoCfg	object	
certManagerCfg	object	

## .spec.tls.providerCfg.autoCfg

### Type

object

Property	Type	Description
altNames	object	AltNames contains the domain names and IP addresses that will be added to the x509 certificate SubAltNames fields. The values will be passed directly to the x509.Certificate object.
caBundleSecret	string	CABundleSecret is the expected secret name with CABundle present. It's used by each etcd POD to verify TLS communications with its peers or clients. If isn't provided, the CA included in the secret generated by certificate provider will be used instead if present; otherwise, there is no way to verify TLS communications.
commonName	string	CommonName is the expected common name X509 certificate subject attribute. Should have a length of 64 characters or fewer to avoid generating invalid CSRs.

Property	Type	Description
<code>organizations</code>	<code>array</code>	Organization is the expected array of Organization names to be used on the Certificate.
<code>validityDuration</code>	<code>string</code>	ValidityDuration is the expected duration until which the certificate will be valid, expects in human-readable duration: 100d12h, if empty defaults to 90d

## `.spec.tls.providerCfg.autoCfg.altNames`

### Description

AltNames contains the domain names and IP addresses that will be added to the x509 certificate SubAltNames fields. The values will be passed directly to the x509.Certificate object.

### Type

`object`

Property	Type	Description
<code>dnsNames</code>	<code>array</code>	DNSNames is the expected array of DNS subject alternative names defaults to <code>\$(POD_NAME).\$(ETCD_CLUSTER_NAME).\$(POD_NAMESPACE)</code>
<code>ipAddresses</code>	<code>array</code>	IPs is the expected array of IP address subject alternative names.

## `.spec.tls.providerCfg.autoCfg.altNames.dnsNames`

### Description

DNSNames is the expected array of DNS subject alternative names. if empty defaults to \$(POD\_NAME).\$(ETCD\_CLUSTER\_NAME).\$(POD\_NAMESPACE).svc.cluster.local

**Type**

array

**.spec.tls.providerCfg.autoCfg.altNames.dnsNames[]****Type**

string

**.spec.tls.providerCfg.autoCfg.altNames.ipAddresses****Description**

IPs is the expected array of IP address subject alternative names.

**Type**

array

**.spec.tls.providerCfg.autoCfg.altNames.ipAddresses[]****Type**

string

**.spec.tls.providerCfg.autoCfg.organizations****Description**

Organization is the expected array of Organization names to be used on the Certificate.

**Type**

array

**.spec.tls.providerCfg.autoCfg.organizations[]****Type**

`string`

## .spec.tls.providerCfg.certManagerCfg

### Type

`object`

### Required

`issuerKind``issuerName`

Property	Type	Description
<code>altNames</code>	<code>object</code>	AltNames contains the domain names and IP addresses that will be added to the x509 certificate SubAltNames fields. The values will be passed directly to the x509.Certificate object.
<code>caBundleSecret</code>	<code>string</code>	CABundleSecret is the expected secret name with CABundle present. It's used by each etcd POD to verify TLS communications with its peers or clients. If it isn't provided, the CA included in the secret generated by certificate provider will be used instead if present; otherwise, there is no way to verify TLS communications.
<code>commonName</code>	<code>string</code>	CommonName is the expected common name X509 certificate subject attribute. Should have a length of 64 characters or fewer to avoid generating invalid CSRs.
<code>issuerKind</code>	<code>string</code>	IssuerKind is the expected kind of Issuer, either "ClusterIssuer" or "Issuer"

Property	Type	Description
<code>issuerName</code>	<code>string</code>	IssuerName is the expected name of Issuer required to issue a certificate
<code>organizations</code>	<code>array</code>	Organization is the expected array of Organization names to be used on the Certificate.
<code>validityDuration</code>	<code>string</code>	ValidityDuration is the expected duration until which the certificate will be valid, expects in human-readable duration: 100d12h, if empty defaults to 90d

## `.spec.tls.providerCfg.certManagerCfg.altNames`

### Description

AltNames contains the domain names and IP addresses that will be added to the x509 certificate SubAltNames fields. The values will be passed directly to the x509.Certificate object.

### Type

`object`

Property	Type	Description
<code>dnsNames</code>	<code>array</code>	DNSNames is the expected array of DNS subject alternative names defaults to <code>\$(POD_NAME).\$(ETCD_CLUSTER_NAME).\$(POD_NAMESPACE)</code>
<code>ipAddresses</code>	<code>array</code>	IPs is the expected array of IP address subject alternative names.

## **.spec.tls.providerCfg.certManagerCfg.altNames.dnsName**

**S**

### **Description**

DNSNames is the expected array of DNS subject alternative names. if empty defaults to \$(POD\_NAME).\$(ETCD\_CLUSTER\_NAME).\$(POD\_NAMESPACE).svc.cluster.local

### **Type**

array

## **.spec.tls.providerCfg.certManagerCfg.altNames.dnsName**

**s[]**

### **Type**

string

## **.spec.tls.providerCfg.certManagerCfg.altNames.ipAddresses**

**es**

### **Description**

IPs is the expected array of IP address subject alternative names.

### **Type**

array

## **.spec.tls.providerCfg.certManagerCfg.altNames.ipAddresses**

**es[]**

### **Type**

string

## **.spec.tls.providerCfg.certManagerCfg.organizations**

## Description

Organization is the expected array of Organization names to be used on the Certificate.

## Type

array

## .spec.tls.providerCfg.certManagerCfg.organizations[]

## Type

string

## .status

## Description

EtcdClusterStatus defines the observed state of EtcdCluster.

## Type

object

# API Endpoints

The following API endpoints are available:

- `/apis/operator.etcd.io/v1alpha1/namespaces/{namespace}/etcdclusters`
  - **DELETE** : delete collection of EtcdCluster
  - **GET** : list objects of kind EtcdCluster
  - **POST** : create a new EtcdCluster
- `/apis/operator.etcd.io/v1alpha1/namespaces/{namespace}/etcdclusters/{name}`
  - **DELETE** : delete the specified EtcdCluster
  - **GET** : read the specified EtcdCluster
  - **PATCH** : partially update the specified EtcdCluster
  - **PUT** : replace the specified EtcdCluster

- `/apis/operator.etcd.io/v1alpha1/namespaces/{namespace}/etcdclusters/{name}/status`
  - `GET`: read status of the specified EtcdCluster
  - `PATCH`: partially update status of the specified EtcdCluster
  - `PUT`: replace status of the specified EtcdCluster

## `/apis/operator.etcd.io/v1alpha1/namespaces/{namespace}/etcdclusters`

### HTTP method

`DELETE`

### Description

delete collection of EtcdCluster

### HTTP responses

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

### HTTP method

`GET`

### Description

list objects of kind EtcdCluster

### HTTP responses

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

### HTTP method

POST

## Description

create a new EtcCluster

## Query parameters

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

## Body parameters

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

## HTTP responses

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

## /apis/operator.etcd.io/v1alpha1/namespaces/{namespace}/etcdclusters/{name}

### HTTP method

DELETE

### Description

delete the specified EtcdCluster

### Query parameters

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

### HTTP responses

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

### HTTP method

GET

## Description

read the specified EtcdCluster

## HTTP responses

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

## HTTP method

PATCH

## Description

partially update the specified EtcdCluster

## Query parameters

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

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

## HTTP responses

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

## HTTP method

PUT

## Description

replace the specified EtcCluster

## Query parameters

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

Parameter	Type	Description
		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	EtcCluster schema	application/json formatted

### HTTP responses

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

## /apis/operator.etcd.io/v1alpha1/namespaces/{namespace}/etcdclusters/{name}/status

### HTTP method

GET

### Description

read status of the specified EtcCluster

### HTTP responses

HTTP code	Response body
200 - OK	EtcCluster schema

HTTP code	Response body
401 - Unauthorized	Empty

## HTTP method

PATCH

## Description

partially update status of the specified EtcCluster

## Query parameters

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

## HTTP responses

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

## HTTP method

PUT

## Description

replace status of the specified EtcdCluster

## Query parameters

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

## Body parameters

Parameter	Type	Description
body	EtcCluster schema	application/json formatted

## HTTP responses

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