

Security APIs

EtcdeEncryptionConfig

[cluster.alauda.io/v1alpha1]

Description

EtcdeEncryptionConfig is the Schema for the etcdeencryptionconfig API

Type

object

Specification

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

Property	Type	Description
		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	
status	object	

.spec

Type

object

Required

resources

rotationInterval

type

Property	Type	Description
resources	array	
rotationInterval	string	
type	string	

.spec.resources

Type

array

.spec.resources[]

Type

string

.status

Type

object

Required

revision

Property	Type	Description
deployStatus	object	
migration	object	
revision	integer	

.status.deployStatus

Type

object

.status.migration

Type

object

Required

message

resources

revision

state

Property	Type	Description
completeTimestamp	string	

Property	Type	Description
message	string	
resources	array	
revision	integer	
state	string	

.status.migration.resources

Type

array

.status.migration.resources[]

Type

string

API Endpoints

The following API endpoints are available:

- `/apis/cluster.alauda.io/v1alpha1/etcdencryptionconfigs`
 - **DELETE** : delete collection of EtcdEncryptionConfig
 - **GET** : list objects of kind EtcdEncryptionConfig
 - **POST** : create a new EtcdEncryptionConfig
- `/apis/cluster.alauda.io/v1alpha1/etcdencryptionconfigs/{name}`
 - **DELETE** : delete the specified EtcdEncryptionConfig
 - **GET** : read the specified EtcdEncryptionConfig
 - **PATCH** : partially update the specified EtcdEncryptionConfig

- **PUT** : replace the specified EtcdEncryptionConfig
- `/apis/cluster.alauda.io/v1alpha1/etcdencryptionconfigs/{name}/status`
- **GET** : read status of the specified EtcdEncryptionConfig
- **PATCH** : partially update status of the specified EtcdEncryptionConfig
- **PUT** : replace status of the specified EtcdEncryptionConfig

/apis/cluster.alauda.io/v1alpha1/etcdencryptionconfigs

HTTP method

DELETE

Description

delete collection of EtcdEncryptionConfig

HTTP responses

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

HTTP method

GET

Description

list objects of kind EtcdEncryptionConfig

HTTP responses

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

HTTP method

POST

Description

create a new EtcdeEncryptionConfig

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>EtcdeEncryptionConfig</code> schema	<code>application/json</code> formatted

HTTP responses

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

/apis/cluster.alauda.io/v1alpha1/etcdencryptionconfigs/{name}

HTTP method

DELETE

Description

delete the specified EtcdeEncryptionConfig

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 EtcEncryptionConfig

HTTP responses

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

HTTP method

PATCH

Description

partially update the specified EtcEncryptionConfig

Query parameters

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

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

HTTP responses

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

HTTP method

PUT

Description

replace the specified EtdcEncryptionConfig

Query parameters

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

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

Body parameters

Parameter	Type	Description
body	EtcdEncryptionConfig schema	application/json formatted

HTTP responses

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

/apis/cluster.alauda.io/v1alpha1/etcdencryptionconfigs/{name}/status

HTTP method

GET

Description

read status of the specified EtcdEncryptionConfig

HTTP responses

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

HTTP method

PATCH

Description

partially update status of the specified EtcdEncryptionConfig

Query parameters

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

HTTP responses

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

HTTP method

PUT

Description

replace status of the specified EtcdEncryptionConfig

Query parameters

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

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	<p><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:</p> <ul style="list-style-type: none"> - 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>EtcdeEncryptionConfig</code> schema	<code>application/json</code> formatted

HTTP responses

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