

[Alauda DevOps Connectors Docs](#) > [API Reference](#) > [Kubernetes APIs](#) > Operator APIs

Operator APIs

[ConnectorsCore \\[operator.com](#) [ConnectorsGitHub \[operator.co](#) [Connectors](#)

[ConnectorsGit \\[operator.conne](#) [ConnectorsHarbor \[operator.co](#) [Connectors](#)

[ConnectorsK8S \\[operator.conr](#) [ConnectorsMaven \\[operator.cc](#) [Connectors](#)

[ConnectorsNPM \\[operator.com](#) [ConnectorsOCI \\[operator.conn](#) [Connectors](#)

[ConnectorsSonarQube \\[operator.connectors.alauda.io/v1alpha1\]](#)

ConnectorsCore

[operator.connectors.alauda.io/v1alpha1]

Description

ConnectorsCore is the Schema for the connectorscore 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 ↗
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	ConnectorsCoreSpec defines the desired state of ConnectorsCore.
<code>status</code>	<code>object</code>	ComponentStatus defines the observed state of Connector

.spec

Description

ConnectorsCoreSpec defines the desired state of ConnectorsCore.

Type

`object`

Property	Type	Description
<code>additionalManifests</code>	<code>string</code>	A means to specify the additional manifests to install, eg: http://example.com/manifests.yaml ↗
<code>annotations</code>	<code>object</code>	Annotations overrides labels for the namespace and its template.

Property	Type	Description
<code>featureFlags</code>	<code>object</code>	FeatureFlags overrides feature flag values in the connectors-config ConfigMap. Keys must match the feature flag names defined in connectors-config. Values are string booleans ("true" or "false").
<code>labels</code>	<code>object</code>	Labels overrides labels for the namespace and its template.
<code>registry</code>	<code>string</code>	Registry overrides the default registry for the workloads. The registry is used to replace the prefix of the image. the image reference format is <code>/:@</code> .
<code>workloads</code>	<code>array</code>	Workloads overrides workloads configurations such as resources and replicas.

.spec.annotations

Description

Annotations overrides labels for the namespace and its template.

Type

`object`

.spec.featureFlags

Description

FeatureFlags overrides feature flag values in the connectors-config ConfigMap. Keys must match the feature flag names defined in connectors-config. Values are string booleans ("true" or "false").

Type

object

.spec.labels

Description

Labels overrides labels for the namespace and its template.

Type

object

.spec.workloads

Description

Workloads overrides workloads configurations such as resources and replicas.

Type

array

.spec.workloads[]

Description

WorkloadOverride overrides the workload configurations.

Type

object

Required

name

Property	Type	Description
<code>name</code>	<code>string</code>	Name is the name of the deployment to override.
<code>replicas</code>	<code>integer</code>	Number of desired pods. This is a pointer to distinguish between explicit zero and not specified. Defaults to 1.
<code>template</code>		PodTemplateSpecJSON describes the pods that will be created. The only allowed template.spec.restartPolicy value is "Always".

.status

Description

ComponentStatus defines the observed state of Connector

Type

`object`

Property	Type	Description
<code>annotations</code>	<code>object</code>	Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Property	Type	Description
<code>conditions</code>	<code>array</code>	Conditions the latest available observations of a resource's current state.
<code>observedGeneration</code>	<code>integer</code>	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

.status.annotations

Description

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Type

`object`

.status.conditions

Description

Conditions the latest available observations of a resource's current state.

Type

`array`

.status.conditions[]

Description

Condition defines a readiness condition for a Knative resource. See: <https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties>

Type

object

Required

status

type

Property	Type	Description
lastTransitionTime	string	LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude this from creating equality.Semantic differences (all other things held constant).
message	string	A human readable message indicating details about the transition.
reason	string	The reason for the condition's last transition.
severity	string	Severity with which to treat failures of this type of condition. When this is not specified, it defaults to Error.
status	string	Status of the condition, one of True, False, Unknown.
type	string	Type of condition.

API Endpoints

The following API endpoints are available:

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

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorscores

HTTP method

DELETE

Description

delete collection of ConnectorsCore

HTTP responses

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

HTTP method

GET

Description

list objects of kind ConnectorsCore

HTTP responses

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

HTTP method

POST

Description

create a new ConnectorsCore

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

Parameter	Type	Description
		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	ConnectorsCore schema	application/json formatted

HTTP responses

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

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorscores/{name}

HTTP method

DELETE

Description

delete the specified ConnectorsCore

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 ConnectorsCore

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified ConnectorsCore

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>ConnectorsCore</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified ConnectorsCore

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.

Body parameters

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

HTTP responses

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

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorscores/{name}/status

HTTP method

`GET`

Description

read status of the specified ConnectorsCore

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified ConnectorsCore

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

Parameter	Type	Description
		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>ConnectorsCore</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified ConnectorsCore

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

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsCore</code> schema
201 - Created	<code>ConnectorsCore</code> schema

HTTP code	Response body
401 - Unauthorized	Empty

ConnectorsGitHub

[operator.connectors.alauda.io/v1alpha1]

Description

ConnectorsGitHub is the Schema for the ConnectorsGitHubs 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 ↗
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	ConnectorsGitHubSpec defines the desired state of ConnectorsGitHub.
<code>status</code>	<code>object</code>	ComponentStatus defines the observed state of Connector

.spec

Description

ConnectorsGitHubSpec defines the desired state of ConnectorsGitHub.

Type

`object`

Property	Type	Description
<code>additionalManifests</code>	<code>string</code>	A means to specify the additional manifests to install, eg: http://example.com/manifests.yaml ↗
<code>annotations</code>	<code>object</code>	Annotations overrides labels for the namespace and its template.

Property	Type	Description
<code>labels</code>	<code>object</code>	Labels overrides labels for the namespace and its template.
<code>registry</code>	<code>string</code>	Registry overrides the default registry for the workloads. The registry is used to replace the prefix of the image. the image reference format is <code>!:@</code> .
<code>workloads</code>	<code>array</code>	Workloads overrides workloads configurations such as resources and replicas.

`.spec.annotations`

Description

Annotations overrides labels for the namespace and its template.

Type

`object`

`.spec.labels`

Description

Labels overrides labels for the namespace and its template.

Type

`object`

`.spec.workloads`

Description

Workloads overrides workloads configurations such as resources and replicas.

Type

array

.spec.workloads[]

Description

WorkloadOverride overrides the workload configurations.

Type

object

Required

name

Property	Type	Description
name	string	Name is the name of the deployment to override.
replicas	integer	Number of desired pods. This is a pointer to distinguish between explicit zero and not specified. Defaults to 1.
template		PodTemplateSpecJSON describes the pods that will be created. The only allowed template.spec.restartPolicy value is "Always".

.status

Description

ComponentStatus defines the observed state of Connector

Type

object

Property	Type	Description
<code>annotations</code>	object	Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.
<code>conditions</code>	array	Conditions the latest available observations of a resource's current state.
<code>observedGeneration</code>	integer	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

.status.annotations

Description

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Type

object

.status.conditions

Description

Conditions the latest available observations of a resource's current state.

Type

array

.status.conditions[]

Description

Condition defines a readiness condition for a Knative resource. See: <https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties>

Type

object

Required

status

type

Property	Type	Description
lastTransitionTime	string	LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude this from creating equality.Semantic differences (all other things held constant).
message	string	A human readable message indicating details about the transition.
reason	string	The reason for the condition's last transition.
severity	string	Severity with which to treat failures of this type of condition. When this is not specified, it defaults to

Property	Type	Description
		Error.
status	string	Status of the condition, one of True, False, Unknown.
type	string	Type of condition.

API Endpoints

The following API endpoints are available:

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

- **PATCH** : partially update status of the specified ConnectorsGitHub
- **PUT** : replace status of the specified ConnectorsGitHub

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorgithubs

HTTP method

DELETE

Description

delete collection of ConnectorsGitHub

HTTP responses

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

HTTP method

GET

Description

list objects of kind ConnectorsGitHub

HTTP responses

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

HTTP method

POST

Description

create a new ConnectorsGitHub

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.

Body parameters

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

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsGitHub</code> schema

HTTP code	Response body
201 - Created	ConnectorsGitHub schema
202 - Accepted	ConnectorsGitHub schema
401 - Unauthorized	Empty

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsgithubs/{name}

HTTP method

[DELETE](#)

Description

delete the specified ConnectorsGitHub

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 ConnectorsGitHub

HTTP responses

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

HTTP method

PATCH

Description

partially update the specified ConnectorsGitHub

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

Parameter	Type	Description
		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>ConnectorsGitHub</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified ConnectorsGitHub

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

Parameter	Type	Description
		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	ConnectorsGitHub schema	application/json formatted

HTTP responses

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

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsgithubs/{name}/status

HTTP method

GET

Description

read status of the specified ConnectorsGitHub

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified ConnectorsGitHub

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>ConnectorsGitHub</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified ConnectorsGitHub

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

HTTP responses

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

ConnectorsGitLab

[operator.connectors.alauda.io/v1alpha1]

Description

ConnectorsGitLab is the Schema for the ConnectorsGitLabs 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 ↗
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	ConnectorsGitLabSpec defines the desired state of ConnectorsGitLab.
<code>status</code>	<code>object</code>	ComponentStatus defines the observed state of Connector

.spec

Description

ConnectorsGitLabSpec defines the desired state of ConnectorsGitLab.

Type

`object`

Property	Type	Description
<code>additionalManifests</code>	<code>string</code>	A means to specify the additional manifests to install, eg: http://example.com/manifests.yaml ↗
<code>annotations</code>	<code>object</code>	Annotations overrides labels for the namespace and its template.

Property	Type	Description
<code>labels</code>	<code>object</code>	Labels overrides labels for the namespace and its template.
<code>registry</code>	<code>string</code>	Registry overrides the default registry for the workloads. The registry is used to replace the prefix of the image. the image reference format is <code>!:@</code> .
<code>workloads</code>	<code>array</code>	Workloads overrides workloads configurations such as resources and replicas.

`.spec.annotations`

Description

Annotations overrides labels for the namespace and its template.

Type

`object`

`.spec.labels`

Description

Labels overrides labels for the namespace and its template.

Type

`object`

`.spec.workloads`

Description

Workloads overrides workloads configurations such as resources and replicas.

Type

array

.spec.workloads[]

Description

WorkloadOverride overrides the workload configurations.

Type

object

Required

name

Property	Type	Description
name	string	Name is the name of the deployment to override.
replicas	integer	Number of desired pods. This is a pointer to distinguish between explicit zero and not specified. Defaults to 1.
template		PodTemplateSpecJSON describes the pods that will be created. The only allowed template.spec.restartPolicy value is "Always".

.status

Description

ComponentStatus defines the observed state of Connector

Type

object

Property	Type	Description
<code>annotations</code>	object	Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.
<code>conditions</code>	array	Conditions the latest available observations of a resource's current state.
<code>observedGeneration</code>	integer	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

.status.annotations

Description

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Type

object

.status.conditions

Description

Conditions the latest available observations of a resource's current state.

Type

array

.status.conditions[]

Description

Condition defines a readiness condition for a Knative resource. See: <https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties>

Type

object

Required

status

type

Property	Type	Description
lastTransitionTime	string	LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude this from creating equality.Semantic differences (all other things held constant).
message	string	A human readable message indicating details about the transition.
reason	string	The reason for the condition's last transition.
severity	string	Severity with which to treat failures of this type of condition. When this is not specified, it defaults to

Property	Type	Description
		Error.
<code>status</code>	<code>string</code>	Status of the condition, one of True, False, Unknown.
<code>type</code>	<code>string</code>	Type of condition.

API Endpoints

The following API endpoints are available:

- `/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsgitlabs`
 - `DELETE` : delete collection of ConnectorsGitLab
 - `GET` : list objects of kind ConnectorsGitLab
 - `POST` : create a new ConnectorsGitLab
- `/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsgitlabs/{name}`
 - `DELETE` : delete the specified ConnectorsGitLab
 - `GET` : read the specified ConnectorsGitLab
 - `PATCH` : partially update the specified ConnectorsGitLab
 - `PUT` : replace the specified ConnectorsGitLab
- `/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsgitlabs/{name}/status`
 - `GET` : read status of the specified ConnectorsGitLab

- **PATCH** : partially update status of the specified ConnectorsGitLab
- **PUT** : replace status of the specified ConnectorsGitLab

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsgitlabs

HTTP method

DELETE

Description

delete collection of ConnectorsGitLab

HTTP responses

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

HTTP method

GET

Description

list objects of kind ConnectorsGitLab

HTTP responses

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

HTTP method

POST

Description

create a new ConnectorsGitLab

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

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsGitLab</code> schema

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

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsgitlabs/{name}

HTTP method

`DELETE`

Description

delete the specified ConnectorsGitLab

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 ConnectorsGitLab

HTTP responses

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

HTTP method

PATCH

Description

partially update the specified ConnectorsGitLab

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

Parameter	Type	Description
		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>ConnectorsGitLab</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified `ConnectorsGitLab`

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

Parameter	Type	Description
		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	ConnectorsGitLab schema	application/json formatted

HTTP responses

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

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsgitlabs/{name}/status

HTTP method

GET

Description

read status of the specified ConnectorsGitLab

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified ConnectorsGitLab

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>ConnectorsGitLab</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified ConnectorsGitLab

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

HTTP responses

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

ConnectorsGit

[operator.connectors.alauda.io/v1alpha1]

Description

ConnectorsGit is the Schema for the ConnectorsGits 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 ↗
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	ConnectorsGitSpec defines the desired state of ConnectorsGit.
<code>status</code>	<code>object</code>	ComponentStatus defines the observed state of Connector

.spec

Description

ConnectorsGitSpec defines the desired state of ConnectorsGit.

Type

`object`

Property	Type	Description
<code>additionalManifests</code>	<code>string</code>	A means to specify the additional manifests to install, eg: http://example.com/manifests.yaml ↗
<code>annotations</code>	<code>object</code>	Annotations overrides labels for the namespace and its template.

Property	Type	Description
<code>labels</code>	<code>object</code>	Labels overrides labels for the namespace and its template.
<code>registry</code>	<code>string</code>	Registry overrides the default registry for the workloads. The registry is used to replace the prefix of the image. the image reference format is <code>!:@</code> .
<code>workloads</code>	<code>array</code>	Workloads overrides workloads configurations such as resources and replicas.

`.spec.annotations`

Description

Annotations overrides labels for the namespace and its template.

Type

`object`

`.spec.labels`

Description

Labels overrides labels for the namespace and its template.

Type

`object`

`.spec.workloads`

Description

Workloads overrides workloads configurations such as resources and replicas.

Type

array

.spec.workloads[]

Description

WorkloadOverride overrides the workload configurations.

Type

object

Required

name

Property	Type	Description
name	string	Name is the name of the deployment to override.
replicas	integer	Number of desired pods. This is a pointer to distinguish between explicit zero and not specified. Defaults to 1.
template		PodTemplateSpecJSON describes the pods that will be created. The only allowed template.spec.restartPolicy value is "Always".

.status

Description

ComponentStatus defines the observed state of Connector

Type

object

Property	Type	Description
<code>annotations</code>	object	Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.
<code>conditions</code>	array	Conditions the latest available observations of a resource's current state.
<code>observedGeneration</code>	integer	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

.status.annotations

Description

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Type

object

.status.conditions

Description

Conditions the latest available observations of a resource's current state.

Type

array

.status.conditions[]

Description

Condition defines a readiness condition for a Knative resource. See: <https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties>

Type

object

Required

status

type

Property	Type	Description
lastTransitionTime	string	LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude this from creating equality.Semantic differences (all other things held constant).
message	string	A human readable message indicating details about the transition.
reason	string	The reason for the condition's last transition.
severity	string	Severity with which to treat failures of this type of condition. When this is not specified, it defaults to

Property	Type	Description
		Error.
<code>status</code>	<code>string</code>	Status of the condition, one of True, False, Unknown.
<code>type</code>	<code>string</code>	Type of condition.

API Endpoints

The following API endpoints are available:

- `/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsgits`
 - `DELETE` : delete collection of ConnectorsGit
 - `GET` : list objects of kind ConnectorsGit
 - `POST` : create a new ConnectorsGit
- `/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsgits/{name}`
 - `DELETE` : delete the specified ConnectorsGit
 - `GET` : read the specified ConnectorsGit
 - `PATCH` : partially update the specified ConnectorsGit
 - `PUT` : replace the specified ConnectorsGit
- `/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsgits/{name}/status`
 - `GET` : read status of the specified ConnectorsGit

- **PATCH** : partially update status of the specified ConnectorsGit
- **PUT** : replace status of the specified ConnectorsGit

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsgits

HTTP method

DELETE

Description

delete collection of ConnectorsGit

HTTP responses

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

HTTP method

GET

Description

list objects of kind ConnectorsGit

HTTP responses

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

HTTP method

POST

Description

create a new ConnectorsGit

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

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsGit</code> schema

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

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorgits/{name}

HTTP method

`DELETE`

Description

delete the specified ConnectorsGit

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 ConnectorsGit

HTTP responses

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

HTTP method

PATCH

Description

partially update the specified ConnectorsGit

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

Parameter	Type	Description
		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>ConnectorsGit</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified ConnectorsGit

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

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

HTTP responses

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

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsgits/{name}/status

HTTP method

`GET`

Description

read status of the specified ConnectorsGit

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified ConnectorsGit

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>ConnectorsGit</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified ConnectorsGit

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

HTTP responses

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

ConnectorsHarbor

[operator.connectors.alauda.io/v1alpha1]

Description

ConnectorsHarbor is the Schema for the ConnectorsHarbors 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 ↗
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	ConnectorsHarborSpec defines the desired state of ConnectorsHarbor.
<code>status</code>	<code>object</code>	ComponentStatus defines the observed state of Connector

.spec

Description

ConnectorsHarborSpec defines the desired state of ConnectorsHarbor.

Type

`object`

Property	Type	Description
<code>additionalManifests</code>	<code>string</code>	A means to specify the additional manifests to install, eg: http://example.com/manifests.yaml ↗
<code>annotations</code>	<code>object</code>	Annotations overrides labels for the namespace and its template.

Property	Type	Description
<code>expose</code>	<code>object</code>	Expose defines the expose configuration for the connector
<code>labels</code>	<code>object</code>	Labels overrides labels for the namespace and its template.
<code>registry</code>	<code>string</code>	Registry overrides the default registry for the workloads. The registry is used to replace the prefix of the image. the image reference format is <code>!:@</code> .
<code>workloads</code>	<code>array</code>	Workloads overrides workloads configurations such as resources and replicas.

.spec.annotations

Description

Annotations overrides labels for the namespace and its template.

Type

`object`

.spec.expose

Description

Expose defines the expose configuration for the connector

Type

object

Required

type

Property	Type	Description
		Domain specifies the access domain for the connector service. Required when Type is Ingress or NodePort:
domain	string	- For Ingress: Set to your ingress domain (e.g., "connectors.example.com") - For NodePort: Set to your cluster node IP (e.g., "192.168.1.100")
ingress	object	Ingress configuration for external access via ingress controller. Required when Type is Ingress.
nodePort	object	NodePort configuration for external access via node port. Required when Type is NodePort.
		Type specifies the exposure method for the OCI connector service.
type	string	- ClusterIP: Internal cluster access only (default) - NodePort: Access via node IP and port from outside the cluster - Ingress: Access via ingress controller with domain name

.spec.expose.ingress

Description

Ingress configuration for external access via ingress controller. Required when Type is Ingress.

Type

object

Property	Type	Description
<code>ingressClassName</code>	<code>string</code>	IngressClassName specifies the ingress controller class to use. Leave empty to use the default ingress controller. eg: "nginx"
<code>secretName</code>	<code>string</code>	SecretName specifies the TLS secret for HTTPS access. Example: "connectors-tls-secret"

.spec.expose.nodePort

Description

NodePort configuration for external access via node port. Required when Type is NodePort.

Type

object

Property	Type	Description
<code>port</code>	<code>integer</code>	Port specifies the node port number for external access. Must be between 30000-32767 (Kubernetes node port range). Recommended: Use ports 30000-31000 to avoid conflicts. Example: 30080 for HTTP, 30443 for HTTPS

.spec.labels

Description

Labels overrides labels for the namespace and its template.

Type

object

.spec.workloads

Description

Workloads overrides workloads configurations such as resources and replicas.

Type

array

.spec.workloads[]

Description

WorkloadOverride overrides the workload configurations.

Type

object

Required

name

Property	Type	Description
name	string	Name is the name of the deployment to override.
replicas	integer	Number of desired pods. This is a pointer to distinguish between explicit zero and not specified. Defaults to 1.

Property	Type	Description
<code>template</code>	<code>object</code>	PodTemplateSpecJSON describes the pods that will be created. The only allowed template.spec.restartPolicy value is "Always".

`.status`

Description

ComponentStatus defines the observed state of Connector

Type

`object`

Property	Type	Description
<code>annotations</code>	<code>object</code>	Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.
<code>conditions</code>	<code>array</code>	Conditions the latest available observations of a resource's current state.
<code>observedGeneration</code>	<code>integer</code>	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

`.status.annotations`

Description

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Type

object

.status.conditions

Description

Conditions the latest available observations of a resource's current state.

Type

array

.status.conditions[]

Description

Condition defines a readiness condition for a Knative resource. See:
<https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties>

Type

object

Required

status

type

Property	Type	Description
<code>lastTransitionTime</code>	<code>string</code>	LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude this from creating equality.Semantic differences (all other things held constant).
<code>message</code>	<code>string</code>	A human readable message indicating details about the transition.
<code>reason</code>	<code>string</code>	The reason for the condition's last transition.
<code>severity</code>	<code>string</code>	Severity with which to treat failures of this type of condition. When this is not specified, it defaults to Error.
<code>status</code>	<code>string</code>	Status of the condition, one of True, False, Unknown.
<code>type</code>	<code>string</code>	Type of condition.

API Endpoints

The following API endpoints are available:

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

`/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsharbors`

HTTP method

DELETE

Description

delete collection of ConnectorsHarbor

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind ConnectorsHarbor

HTTP responses

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

HTTP method

POST

Description

create a new ConnectorsHarbor

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

Parameter	Type	Description
		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	ConnectorsHarbor schema	application/json formatted

HTTP responses

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

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsharbors/{name}

HTTP method

DELETE

Description

delete the specified ConnectorsHarbor

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	Status <code>schema</code>
202 - Accepted	Status <code>schema</code>
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified ConnectorsHarbor

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified ConnectorsHarbor

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>ConnectorsHarbor</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified ConnectorsHarbor

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.

Body parameters

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

HTTP responses

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

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsharbors/{name}/status

HTTP method

`GET`

Description

read status of the specified ConnectorsHarbor

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified ConnectorsHarbor

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

Parameter	Type	Description
		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>ConnectorsHarbor</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified ConnectorsHarbor

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

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsHarbor</code> schema
201 - Created	<code>ConnectorsHarbor</code> schema

HTTP code	Response body
401 - Unauthorized	Empty

ConnectorsJFrog

[operator.connectors.alauda.io/v1alpha1]

Description

ConnectorsJFrog is the Schema for the ConnectorsJFrogs 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 ↗
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	ConnectorsJFrogSpec defines the desired state of ConnectorsJFrog.
<code>status</code>	<code>object</code>	ComponentStatus defines the observed state of Connector

.spec

Description

ConnectorsJFrogSpec defines the desired state of ConnectorsJFrog.

Type

`object`

Property	Type	Description
<code>additionalManifests</code>	<code>string</code>	A means to specify the additional manifests to install, eg: http://example.com/manifests.yaml ↗
<code>annotations</code>	<code>object</code>	Annotations overrides labels for the namespace and its template.

Property	Type	Description
<code>labels</code>	<code>object</code>	Labels overrides labels for the namespace and its template.
<code>registry</code>	<code>string</code>	Registry overrides the default registry for the workloads. The registry is used to replace the prefix of the image. the image reference format is <code>!:@</code> .
<code>workloads</code>	<code>array</code>	Workloads overrides workloads configurations such as resources and replicas.

`.spec.annotations`

Description

Annotations overrides labels for the namespace and its template.

Type

`object`

`.spec.labels`

Description

Labels overrides labels for the namespace and its template.

Type

`object`

`.spec.workloads`

Description

Workloads overrides workloads configurations such as resources and replicas.

Type

array

.spec.workloads[]

Description

WorkloadOverride overrides the workload configurations.

Type

object

Required

name

Property	Type	Description
name	string	Name is the name of the deployment to override.
replicas	integer	Number of desired pods. This is a pointer to distinguish between explicit zero and not specified. Defaults to 1.
template		PodTemplateSpecJSON describes the pods that will be created. The only allowed template.spec.restartPolicy value is "Always".

.status

Description

ComponentStatus defines the observed state of Connector

Type

object

Property	Type	Description
<code>annotations</code>	object	Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.
<code>conditions</code>	array	Conditions the latest available observations of a resource's current state.
<code>observedGeneration</code>	integer	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

.status.annotations

Description

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Type

object

.status.conditions

Description

Conditions the latest available observations of a resource's current state.

Type

array

.status.conditions[]

Description

Condition defines a readiness condition for a Knative resource. See: <https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties>

Type

object

Required

status

type

Property	Type	Description
lastTransitionTime	string	LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude this from creating equality.Semantic differences (all other things held constant).
message	string	A human readable message indicating details about the transition.
reason	string	The reason for the condition's last transition.
severity	string	Severity with which to treat failures of this type of condition. When this is not specified, it defaults to

Property	Type	Description
		Error.
status	string	Status of the condition, one of True, False, Unknown.
type	string	Type of condition.

API Endpoints

The following API endpoints are available:

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

- **PATCH** : partially update status of the specified ConnectorsJFrog
- **PUT** : replace status of the specified ConnectorsJFrog

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsjfrogs

HTTP method

DELETE

Description

delete collection of ConnectorsJFrog

HTTP responses

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

HTTP method

GET

Description

list objects of kind ConnectorsJFrog

HTTP responses

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

HTTP method

POST

Description

create a new ConnectorsJFrog

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

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsJFrog</code> schema

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

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsjfrogs/{name}

HTTP method

`DELETE`

Description

delete the specified ConnectorsJFrog

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 ConnectorsJFrog

HTTP responses

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

HTTP method

PATCH

Description

partially update the specified ConnectorsJFrog

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

Parameter	Type	Description
		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>ConnectorsJFrog</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified ConnectorsJFrog

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

Parameter	Type	Description
		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	ConnectorsJFrog schema	application/json formatted

HTTP responses

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

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsjfrogs/{name}/status

HTTP method

GET

Description

read status of the specified ConnectorsJFrog

HTTP responses

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

HTTP method

PATCH

Description

partially update status of the specified ConnectorsJFrog

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>ConnectorsJFrog</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified ConnectorsJFrog

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

Parameter	Type	Description
<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>ConnectorsJFrog</code> schema	<code>application/json</code> formatted

HTTP responses

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

ConnectorsK8S

[operator.connectors.alauda.io/v1alpha1]

Description

ConnectorsK8S is the Schema for the ConnectorsK8Ss 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 ↗
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	ConnectorsK8SSpec defines the desired state of ConnectorsK8S.
<code>status</code>	<code>object</code>	ComponentStatus defines the observed state of Connector

.spec

Description

ConnectorsK8SSpec defines the desired state of ConnectorsK8S.

Type

`object`

Property	Type	Description
<code>additionalManifests</code>	<code>string</code>	A means to specify the additional manifests to install, eg: http://example.com/manifests.yaml ↗
<code>annotations</code>	<code>object</code>	Annotations overrides labels for the namespace and its template.

Property	Type	Description
<code>labels</code>	<code>object</code>	Labels overrides labels for the namespace and its template.
<code>registry</code>	<code>string</code>	Registry overrides the default registry for the workloads. The registry is used to replace the prefix of the image. the image reference format is <code>!:@</code> .
<code>workloads</code>	<code>array</code>	Workloads overrides workloads configurations such as resources and replicas.

.spec.annotations

Description

Annotations overrides labels for the namespace and its template.

Type

`object`

.spec.labels

Description

Labels overrides labels for the namespace and its template.

Type

`object`

.spec.workloads

Description

Workloads overrides workloads configurations such as resources and replicas.

Type

array

.spec.workloads[]

Description

WorkloadOverride overrides the workload configurations.

Type

object

Required

name

Property	Type	Description
name	string	Name is the name of the deployment to override.
replicas	integer	Number of desired pods. This is a pointer to distinguish between explicit zero and not specified. Defaults to 1.
template		PodTemplateSpecJSON describes the pods that will be created. The only allowed template.spec.restartPolicy value is "Always".

.status

Description

ComponentStatus defines the observed state of Connector

Type

object

Property	Type	Description
<code>annotations</code>	object	Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.
<code>conditions</code>	array	Conditions the latest available observations of a resource's current state.
<code>observedGeneration</code>	integer	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

.status.annotations

Description

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Type

object

.status.conditions

Description

Conditions the latest available observations of a resource's current state.

Type

array

.status.conditions[]

Description

Condition defines a readiness condition for a Knative resource. See: <https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties>

Type

object

Required

status

type

Property	Type	Description
lastTransitionTime	string	LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude this from creating equality.Semantic differences (all other things held constant).
message	string	A human readable message indicating details about the transition.
reason	string	The reason for the condition's last transition.
severity	string	Severity with which to treat failures of this type of condition. When this is not specified, it defaults to

Property	Type	Description
		Error.
status	string	Status of the condition, one of True, False, Unknown.
type	string	Type of condition.

API Endpoints

The following API endpoints are available:

- `/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsK8S`
 - **DELETE** : delete collection of ConnectorsK8S
 - **GET** : list objects of kind ConnectorsK8S
 - **POST** : create a new ConnectorsK8S
- `/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsK8S/{name}`
 - **DELETE** : delete the specified ConnectorsK8S
 - **GET** : read the specified ConnectorsK8S
 - **PATCH** : partially update the specified ConnectorsK8S
 - **PUT** : replace the specified ConnectorsK8S
- `/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsK8S/{name}/status`
 - **GET** : read status of the specified ConnectorsK8S

- **PATCH** : partially update status of the specified ConnectorsK8S
- **PUT** : replace status of the specified ConnectorsK8S

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsk8s

HTTP method

DELETE

Description

delete collection of ConnectorsK8S

HTTP responses

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

HTTP method

GET

Description

list objects of kind ConnectorsK8S

HTTP responses

HTTP code	Response body
200 - OK	ConnectorsK8SList schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new ConnectorsK8S

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.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>ConnectorsK8S</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsK8S</code> schema

HTTP code	Response body
201 - Created	<code>ConnectorsK8S</code> schema
202 - Accepted	<code>ConnectorsK8S</code> schema
401 - Unauthorized	Empty

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsk8s/{name}

HTTP method

`DELETE`

Description

delete the specified ConnectorsK8S

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 ConnectorsK8S

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsK8S</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified ConnectorsK8S

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

Parameter	Type	Description
		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>ConnectorsK8S</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified ConnectorsK8S

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

Parameter	Type	Description
		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	ConnectorsK8S schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	ConnectorsK8S schema
201 - Created	ConnectorsK8S schema
401 - Unauthorized	Empty

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsk8s/{name}/status

HTTP method

GET

Description

read status of the specified ConnectorsK8S

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsK8S</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified ConnectorsK8S

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>ConnectorsK8S</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified ConnectorsK8S

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>	<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>ConnectorsK8S</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsK8S</code> schema
201 - Created	<code>ConnectorsK8S</code> schema
401 - Unauthorized	Empty

ConnectorsMaven

[operator.connectors.alauda.io/v1alpha1]

Description

ConnectorsMaven is the Schema for the ConnectorsMavens 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 ↗
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	ConnectorsMavenSpec defines the desired state of ConnectorsMaven.
<code>status</code>	<code>object</code>	ComponentStatus defines the observed state of Connector

.spec

Description

ConnectorsMavenSpec defines the desired state of ConnectorsMaven.

Type

`object`

Property	Type	Description
<code>additionalManifests</code>	<code>string</code>	A means to specify the additional manifests to install, eg: http://example.com/manifests.yaml ↗
<code>annotations</code>	<code>object</code>	Annotations overrides labels for the namespace and its template.

Property	Type	Description
<code>labels</code>	<code>object</code>	Labels overrides labels for the namespace and its template.
<code>registry</code>	<code>string</code>	Registry overrides the default registry for the workloads. The registry is used to replace the prefix of the image. the image reference format is <code>!:@</code> .
<code>workloads</code>	<code>array</code>	Workloads overrides workloads configurations such as resources and replicas.

`.spec.annotations`

Description

Annotations overrides labels for the namespace and its template.

Type

`object`

`.spec.labels`

Description

Labels overrides labels for the namespace and its template.

Type

`object`

`.spec.workloads`

Description

Workloads overrides workloads configurations such as resources and replicas.

Type

array

.spec.workloads[]

Description

WorkloadOverride overrides the workload configurations.

Type

object

Required

name

Property	Type	Description
name	string	Name is the name of the deployment to override.
replicas	integer	Number of desired pods. This is a pointer to distinguish between explicit zero and not specified. Defaults to 1.
template		PodTemplateSpecJSON describes the pods that will be created. The only allowed template.spec.restartPolicy value is "Always".

.status

Description

ComponentStatus defines the observed state of Connector

Type

object

Property	Type	Description
<code>annotations</code>	object	Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.
<code>conditions</code>	array	Conditions the latest available observations of a resource's current state.
<code>observedGeneration</code>	integer	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

.status.annotations

Description

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Type

object

.status.conditions

Description

Conditions the latest available observations of a resource's current state.

Type

array

.status.conditions[]

Description

Condition defines a readiness condition for a Knative resource. See: <https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties>

Type

object

Required

status

type

Property	Type	Description
lastTransitionTime	string	LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude this from creating equality.Semantic differences (all other things held constant).
message	string	A human readable message indicating details about the transition.
reason	string	The reason for the condition's last transition.
severity	string	Severity with which to treat failures of this type of condition. When this is not specified, it defaults to

Property	Type	Description
		Error.
status	string	Status of the condition, one of True, False, Unknown.
type	string	Type of condition.

API Endpoints

The following API endpoints are available:

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

- **PATCH** : partially update status of the specified ConnectorsMaven
- **PUT** : replace status of the specified ConnectorsMaven

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsmavens

HTTP method

DELETE

Description

delete collection of ConnectorsMaven

HTTP responses

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

HTTP method

GET

Description

list objects of kind ConnectorsMaven

HTTP responses

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

HTTP method

POST

Description

create a new ConnectorsMaven

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.

Body parameters

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

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsMaven</code> schema

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

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsmavens/{name}

HTTP method

`DELETE`

Description

delete the specified ConnectorsMaven

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 ConnectorsMaven

HTTP responses

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

HTTP method

PATCH

Description

partially update the specified ConnectorsMaven

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

Parameter	Type	Description
		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>ConnectorsMaven</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified ConnectorsMaven

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

Parameter	Type	Description
		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	ConnectorsMaven schema	application/json formatted

HTTP responses

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

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsmavens/{name}/status

HTTP method

GET

Description

read status of the specified ConnectorsMaven

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified ConnectorsMaven

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>ConnectorsMaven</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified ConnectorsMaven

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

HTTP responses

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

ConnectorsNexus

[operator.connectors.alauda.io/v1alpha1]

Description

ConnectorsNexus is the Schema for the ConnectorsNexuses 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 ↗
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	ConnectorsNexusSpec defines the desired state of ConnectorsNexus.
<code>status</code>	<code>object</code>	ComponentStatus defines the observed state of Connector

.spec

Description

ConnectorsNexusSpec defines the desired state of ConnectorsNexus.

Type

`object`

Property	Type	Description
<code>additionalManifests</code>	<code>string</code>	A means to specify the additional manifests to install, eg: http://example.com/manifests.yaml ↗
<code>annotations</code>	<code>object</code>	Annotations overrides labels for the namespace and its template.

Property	Type	Description
<code>labels</code>	<code>object</code>	Labels overrides labels for the namespace and its template.
<code>registry</code>	<code>string</code>	Registry overrides the default registry for the workloads. The registry is used to replace the prefix of the image. the image reference format is <code>!:@</code> .
<code>workloads</code>	<code>array</code>	Workloads overrides workloads configurations such as resources and replicas.

.spec.annotations

Description

Annotations overrides labels for the namespace and its template.

Type

`object`

.spec.labels

Description

Labels overrides labels for the namespace and its template.

Type

`object`

.spec.workloads

Description

Workloads overrides workloads configurations such as resources and replicas.

Type

array

.spec.workloads[]

Description

WorkloadOverride overrides the workload configurations.

Type

object

Required

name

Property	Type	Description
name	string	Name is the name of the deployment to override.
replicas	integer	Number of desired pods. This is a pointer to distinguish between explicit zero and not specified. Defaults to 1.
template		PodTemplateSpecJSON describes the pods that will be created. The only allowed template.spec.restartPolicy value is "Always".

.status

Description

ComponentStatus defines the observed state of Connector

Type

object

Property	Type	Description
<code>annotations</code>	object	Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.
<code>conditions</code>	array	Conditions the latest available observations of a resource's current state.
<code>observedGeneration</code>	integer	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

.status.annotations

Description

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Type

object

.status.conditions

Description

Conditions the latest available observations of a resource's current state.

Type

array

.status.conditions[]

Description

Condition defines a readiness condition for a Knative resource. See: <https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties>

Type

object

Required

status

type

Property	Type	Description
lastTransitionTime	string	LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude this from creating equality.Semantic differences (all other things held constant).
message	string	A human readable message indicating details about the transition.
reason	string	The reason for the condition's last transition.
severity	string	Severity with which to treat failures of this type of condition. When this is not specified, it defaults to

Property	Type	Description
		Error.
status	string	Status of the condition, one of True, False, Unknown.
type	string	Type of condition.

API Endpoints

The following API endpoints are available:

- `/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsnexus`
 - `DELETE` : delete collection of ConnectorsNexus
 - `GET` : list objects of kind ConnectorsNexus
 - `POST` : create a new ConnectorsNexus
- `/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsnexus/{name}`
 - `DELETE` : delete the specified ConnectorsNexus
 - `GET` : read the specified ConnectorsNexus
 - `PATCH` : partially update the specified ConnectorsNexus
 - `PUT` : replace the specified ConnectorsNexus
- `/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsnexus/{name}/status`
 - `GET` : read status of the specified ConnectorsNexus

- **PATCH** : partially update status of the specified ConnectorsNexus
- **PUT** : replace status of the specified ConnectorsNexus

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectornexuses

HTTP method

DELETE

Description

delete collection of ConnectorsNexus

HTTP responses

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

HTTP method

GET

Description

list objects of kind ConnectorsNexus

HTTP responses

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

HTTP method

POST

Description

create a new ConnectorsNexus

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

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsNexus</code> schema

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

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsnexususes/{name}

HTTP method

`DELETE`

Description

delete the specified ConnectorsNexus

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 ConnectorsNexus

HTTP responses

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

HTTP method

PATCH

Description

partially update the specified ConnectorsNexus

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

Parameter	Type	Description
		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>ConnectorsNexus</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified ConnectorsNexus

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

Parameter	Type	Description
		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	ConnectorsNexus schema	application/json formatted

HTTP responses

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

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsnexus/{name}/status

HTTP method

GET

Description

read status of the specified ConnectorsNexus

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified ConnectorsNexus

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>ConnectorsNexus</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified ConnectorsNexus

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

HTTP responses

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

ConnectorsNPM

[operator.connectors.alauda.io/v1alpha1]

Description

ConnectorsNPM is the Schema for the ConnectorsNPMs 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 ↗
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	ConnectorsNPMSpec defines the desired state of ConnectorsNPM.
<code>status</code>	<code>object</code>	ComponentStatus defines the observed state of Connector

.spec

Description

ConnectorsNPMSpec defines the desired state of ConnectorsNPM.

Type

`object`

Property	Type	Description
<code>additionalManifests</code>	<code>string</code>	A means to specify the additional manifests to install, eg: http://example.com/manifests.yaml ↗
<code>annotations</code>	<code>object</code>	Annotations overrides labels for the namespace and its template.

Property	Type	Description
<code>labels</code>	<code>object</code>	Labels overrides labels for the namespace and its template.
<code>registry</code>	<code>string</code>	Registry overrides the default registry for the workloads. The registry is used to replace the prefix of the image. the image reference format is <code>!:@</code> .
<code>workloads</code>	<code>array</code>	Workloads overrides workloads configurations such as resources and replicas.

`.spec.annotations`

Description

Annotations overrides labels for the namespace and its template.

Type

`object`

`.spec.labels`

Description

Labels overrides labels for the namespace and its template.

Type

`object`

`.spec.workloads`

Description

Workloads overrides workloads configurations such as resources and replicas.

Type

array

.spec.workloads[]

Description

WorkloadOverride overrides the workload configurations.

Type

object

Required

name

Property	Type	Description
name	string	Name is the name of the deployment to override.
replicas	integer	Number of desired pods. This is a pointer to distinguish between explicit zero and not specified. Defaults to 1.
template		PodTemplateSpecJSON describes the pods that will be created. The only allowed template.spec.restartPolicy value is "Always".

.status

Description

ComponentStatus defines the observed state of Connector

Type

object

Property	Type	Description
annotations	object	Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.
conditions	array	Conditions the latest available observations of a resource's current state.
observedGeneration	integer	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

.status.annotations

Description

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Type

object

.status.conditions

Description

Conditions the latest available observations of a resource's current state.

Type

array

.status.conditions[]

Description

Condition defines a readiness condition for a Knative resource. See: <https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties>

Type

object

Required

status

type

Property	Type	Description
lastTransitionTime	string	LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude this from creating equality.Semantic differences (all other things held constant).
message	string	A human readable message indicating details about the transition.
reason	string	The reason for the condition's last transition.
severity	string	Severity with which to treat failures of this type of condition. When this is not specified, it defaults to

Property	Type	Description
		Error.
<code>status</code>	<code>string</code>	Status of the condition, one of True, False, Unknown.
<code>type</code>	<code>string</code>	Type of condition.

API Endpoints

The following API endpoints are available:

- `/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsnps`
 - `DELETE` : delete collection of ConnectorsNPM
 - `GET` : list objects of kind ConnectorsNPM
 - `POST` : create a new ConnectorsNPM
- `/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsnps/{name}`
 - `DELETE` : delete the specified ConnectorsNPM
 - `GET` : read the specified ConnectorsNPM
 - `PATCH` : partially update the specified ConnectorsNPM
 - `PUT` : replace the specified ConnectorsNPM
- `/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsnps/{name}/status`
 - `GET` : read status of the specified ConnectorsNPM

- **PATCH** : partially update status of the specified ConnectorsNPM
- **PUT** : replace status of the specified ConnectorsNPM

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsnpms

HTTP method

DELETE

Description

delete collection of ConnectorsNPM

HTTP responses

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

HTTP method

GET

Description

list objects of kind ConnectorsNPM

HTTP responses

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

HTTP method

POST

Description

create a new ConnectorsNPM

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

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsNPM</code> schema

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

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsnpms/{name}

HTTP method

DELETE

Description

delete the specified ConnectorsNPM

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 ConnectorsNPM

HTTP responses

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

HTTP method

PATCH

Description

partially update the specified ConnectorsNPM

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

Parameter	Type	Description
		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>ConnectorsNPM</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified ConnectorsNPM

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

Parameter	Type	Description
		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	ConnectorsNPM schema	application/json formatted

HTTP responses

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

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsnpms/{name}/status

HTTP method

GET

Description

read status of the specified ConnectorsNPM

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified ConnectorsNPM

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.

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

HTTP responses

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

ConnectorsOCI

[operator.connectors.alauda.io/v1alpha1]

Description

ConnectorsOCI is the Schema for the ConnectorsOCIs 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 ↗
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	ConnectorsOCISpec defines the desired state of ConnectorsOCI.
<code>status</code>	<code>object</code>	ComponentStatus defines the observed state of Connector

.spec

Description

ConnectorsOCISpec defines the desired state of ConnectorsOCI.

Type

`object`

Property	Type	Description
<code>additionalManifests</code>	<code>string</code>	A means to specify the additional manifests to install, eg: http://example.com/manifests.yaml ↗
<code>annotations</code>	<code>object</code>	Annotations overrides labels for the namespace and its template.

Property	Type	Description
<code>expose</code>	<code>object</code>	Expose defines the expose configuration for the connector
<code>labels</code>	<code>object</code>	Labels overrides labels for the namespace and its template.
<code>registry</code>	<code>string</code>	Registry overrides the default registry for the workloads. The registry is used to replace the prefix of the image. the image reference format is <code>!:@</code> .
<code>workloads</code>	<code>array</code>	Workloads overrides workloads configurations such as resources and replicas.

.spec.annotations

Description

Annotations overrides labels for the namespace and its template.

Type

`object`

.spec.expose

Description

Expose defines the expose configuration for the connector

Type

object

Required

type

Property	Type	Description
		Domain specifies the access domain for the connector service. Required when Type is Ingress or NodePort:
domain	string	- For Ingress: Set to your ingress domain (e.g., "connectors.example.com") - For NodePort: Set to your cluster node IP (e.g., "192.168.1.100")
ingress	object	Ingress configuration for external access via ingress controller. Required when Type is Ingress.
nodePort	object	NodePort configuration for external access via node port. Required when Type is NodePort.
		Type specifies the exposure method for the OCI connector service.
type	string	- ClusterIP: Internal cluster access only (default) - NodePort: Access via node IP and port from outside the cluster - Ingress: Access via ingress controller with domain name

.spec.expose.ingress

Description

Ingress configuration for external access via ingress controller. Required when Type is Ingress.

Type

object

Property	Type	Description
<code>ingressClassName</code>	<code>string</code>	IngressClassName specifies the ingress controller class to use. Leave empty to use the default ingress controller. eg: "nginx"
<code>secretName</code>	<code>string</code>	SecretName specifies the TLS secret for HTTPS access. Example: "connectors-tls-secret"

.spec.expose.nodePort

Description

NodePort configuration for external access via node port. Required when Type is NodePort.

Type

object

Property	Type	Description
<code>port</code>	<code>integer</code>	Port specifies the node port number for external access. Must be between 30000-32767 (Kubernetes node port range). Recommended: Use ports 30000-31000 to avoid conflicts. Example: 30080 for HTTP, 30443 for HTTPS

.spec.labels

Description

Labels overrides labels for the namespace and its template.

Type

object

.spec.workloads

Description

Workloads overrides workloads configurations such as resources and replicas.

Type

array

.spec.workloads[]

Description

WorkloadOverride overrides the workload configurations.

Type

object

Required

name

Property	Type	Description
name	string	Name is the name of the deployment to override.
replicas	integer	Number of desired pods. This is a pointer to distinguish between explicit zero and not specified. Defaults to 1.

Property	Type	Description
<code>template</code>	<code>PodTemplateSpecJSON</code>	PodTemplateSpecJSON describes the pods that will be created. The only allowed template.spec.restartPolicy value is "Always".

`.status`

Description

ComponentStatus defines the observed state of Connector

Type

`object`

Property	Type	Description
<code>annotations</code>	<code>object</code>	Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.
<code>conditions</code>	<code>array</code>	Conditions the latest available observations of a resource's current state.
<code>observedGeneration</code>	<code>integer</code>	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

`.status.annotations`

Description

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Type

object

.status.conditions

Description

Conditions the latest available observations of a resource's current state.

Type

array

.status.conditions[]

Description

Condition defines a readiness condition for a Knative resource. See:
<https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties>

Type

object

Required

status

type

Property	Type	Description
<code>lastTransitionTime</code>	<code>string</code>	LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude this from creating equality.Semantic differences (all other things held constant).
<code>message</code>	<code>string</code>	A human readable message indicating details about the transition.
<code>reason</code>	<code>string</code>	The reason for the condition's last transition.
<code>severity</code>	<code>string</code>	Severity with which to treat failures of this type of condition. When this is not specified, it defaults to Error.
<code>status</code>	<code>string</code>	Status of the condition, one of True, False, Unknown.
<code>type</code>	<code>string</code>	Type of condition.

API Endpoints

The following API endpoints are available:

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

`/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsoci`

HTTP method

DELETE

Description

delete collection of ConnectorsOCI

HTTP responses

HTTP code	Response body
200 - OK	Status  schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind ConnectorsOCI

HTTP responses

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

HTTP method

POST

Description

create a new ConnectorsOCI

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

Parameter	Type	Description
		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	ConnectorsOCI schema	application/json formatted

HTTP responses

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

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsocis/{name}

HTTP method

DELETE

Description

delete the specified ConnectorsOCI

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 ConnectorsOCI

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified ConnectorsOCI

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>ConnectorsOCI</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified ConnectorsOCI

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.

Body parameters

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

HTTP responses

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

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorsocis/{name}/status

HTTP method

`GET`

Description

read status of the specified ConnectorsOCI

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified ConnectorsOCI

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

Parameter	Type	Description
		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>ConnectorsOCI</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified ConnectorsOCI

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.

Body parameters

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

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsOCI</code> schema
201 - Created	<code>ConnectorsOCI</code> schema

HTTP code	Response body
401 - Unauthorized	Empty

ConnectorsPyPI

[operator.connectors.alauda.io/v1alpha1]

Description

ConnectorsPyPI is the Schema for the ConnectorsPyPIs 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 ↗
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	ConnectorsPyPISpec defines the desired state of ConnectorsPyPI.
<code>status</code>	<code>object</code>	ComponentStatus defines the observed state of Connector

.spec

Description

ConnectorsPyPISpec defines the desired state of ConnectorsPyPI.

Type

`object`

Property	Type	Description
<code>additionalManifests</code>	<code>string</code>	A means to specify the additional manifests to install, eg: http://example.com/manifests.yaml ↗
<code>annotations</code>	<code>object</code>	Annotations overrides labels for the namespace and its template.

Property	Type	Description
<code>labels</code>	<code>object</code>	Labels overrides labels for the namespace and its template.
<code>registry</code>	<code>string</code>	Registry overrides the default registry for the workloads. The registry is used to replace the prefix of the image. the image reference format is <code>!:@</code> .
<code>workloads</code>	<code>array</code>	Workloads overrides workloads configurations such as resources and replicas.

.spec.annotations

Description

Annotations overrides labels for the namespace and its template.

Type

`object`

.spec.labels

Description

Labels overrides labels for the namespace and its template.

Type

`object`

.spec.workloads

Description

Workloads overrides workloads configurations such as resources and replicas.

Type

array

.spec.workloads[]

Description

WorkloadOverride overrides the workload configurations.

Type

object

Required

name

Property	Type	Description
name	string	Name is the name of the deployment to override.
replicas	integer	Number of desired pods. This is a pointer to distinguish between explicit zero and not specified. Defaults to 1.
template		PodTemplateSpecJSON describes the pods that will be created. The only allowed template.spec.restartPolicy value is "Always".

.status

Description

ComponentStatus defines the observed state of Connector

Type

object

Property	Type	Description
<code>annotations</code>	object	Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.
<code>conditions</code>	array	Conditions the latest available observations of a resource's current state.
<code>observedGeneration</code>	integer	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

.status.annotations

Description

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Type

object

.status.conditions

Description

Conditions the latest available observations of a resource's current state.

Type

array

.status.conditions[]

Description

Condition defines a readiness condition for a Knative resource. See: <https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties>

Type

object

Required

status

type

Property	Type	Description
lastTransitionTime	string	LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude this from creating equality.Semantic differences (all other things held constant).
message	string	A human readable message indicating details about the transition.
reason	string	The reason for the condition's last transition.
severity	string	Severity with which to treat failures of this type of condition. When this is not specified, it defaults to

Property	Type	Description
		Error.
status	string	Status of the condition, one of True, False, Unknown.
type	string	Type of condition.

API Endpoints

The following API endpoints are available:

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

- **PATCH** : partially update status of the specified ConnectorsPyPI
- **PUT** : replace status of the specified ConnectorsPyPI

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorspypis

HTTP method

DELETE

Description

delete collection of ConnectorsPyPI

HTTP responses

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

HTTP method

GET

Description

list objects of kind ConnectorsPyPI

HTTP responses

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

HTTP method

POST

Description

create a new ConnectorsPyPI

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

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsPyPI</code> schema

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

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorspypis/{name}

HTTP method

`DELETE`

Description

delete the specified ConnectorsPyPI

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 ConnectorsPyPI

HTTP responses

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

HTTP method

PATCH

Description

partially update the specified ConnectorsPyPI

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

Parameter	Type	Description
		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>ConnectorsPyPI</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified ConnectorsPyPI

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

Parameter	Type	Description
		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	ConnectorsPyPI schema	application/json formatted

HTTP responses

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

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorspypis/{name}/status

HTTP method

GET

Description

read status of the specified ConnectorsPyPI

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified ConnectorsPyPI

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.

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

HTTP responses

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

ConnectorsSonarQube

[operator.connectors.alauda.io/v1alpha1]

Description

ConnectorsSonarQube is the Schema for the ConnectorsSonarQubes 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 ↗
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	ConnectorsSonarQubeSpec defines the desired state of ConnectorsSonarQube.
<code>status</code>	<code>object</code>	ComponentStatus defines the observed state of Connector

.spec

Description

ConnectorsSonarQubeSpec defines the desired state of ConnectorsSonarQube.

Type

`object`

Property	Type	Description
<code>additionalManifests</code>	<code>string</code>	A means to specify the additional manifests to install, eg: http://example.com/manifests.yaml ↗
<code>annotations</code>	<code>object</code>	Annotations overrides labels for the namespace and its template.

Property	Type	Description
<code>labels</code>	<code>object</code>	Labels overrides labels for the namespace and its template.
<code>registry</code>	<code>string</code>	Registry overrides the default registry for the workloads. The registry is used to replace the prefix of the image. the image reference format is <code>!:@</code> .
<code>workloads</code>	<code>array</code>	Workloads overrides workloads configurations such as resources and replicas.

`.spec.annotations`

Description

Annotations overrides labels for the namespace and its template.

Type

`object`

`.spec.labels`

Description

Labels overrides labels for the namespace and its template.

Type

`object`

`.spec.workloads`

Description

Workloads overrides workloads configurations such as resources and replicas.

Type

array

.spec.workloads[]

Description

WorkloadOverride overrides the workload configurations.

Type

object

Required

name

Property	Type	Description
name	string	Name is the name of the deployment to override.
replicas	integer	Number of desired pods. This is a pointer to distinguish between explicit zero and not specified. Defaults to 1.
template		PodTemplateSpecJSON describes the pods that will be created. The only allowed template.spec.restartPolicy value is "Always".

.status

Description

ComponentStatus defines the observed state of Connector

Type

object

Property	Type	Description
<code>annotations</code>	object	Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.
<code>conditions</code>	array	Conditions the latest available observations of a resource's current state.
<code>observedGeneration</code>	integer	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

.status.annotations

Description

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Type

object

.status.conditions

Description

Conditions the latest available observations of a resource's current state.

Type

array

.status.conditions[]

Description

Condition defines a readiness condition for a Knative resource. See: <https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties>

Type

object

Required

status

type

Property	Type	Description
lastTransitionTime	string	LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude this from creating equality.Semantic differences (all other things held constant).
message	string	A human readable message indicating details about the transition.
reason	string	The reason for the condition's last transition.
severity	string	Severity with which to treat failures of this type of condition. When this is not specified, it defaults to

Property	Type	Description
		Error.
status	string	Status of the condition, one of True, False, Unknown.
type	string	Type of condition.

API Endpoints

The following API endpoints are available:

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

- **PATCH** : partially update status of the specified ConnectorsSonarQube
- **PUT** : replace status of the specified ConnectorsSonarQube

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorssonarqubes

HTTP method

DELETE

Description

delete collection of ConnectorsSonarQube

HTTP responses

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

HTTP method

GET

Description

list objects of kind ConnectorsSonarQube

HTTP responses

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

HTTP method

POST

Description

create a new ConnectorsSonarQube

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>	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
<code>body</code>	<code>ConnectorsSonarQube</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ConnectorsSonarQube</code> schema

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

/apis/operator.connectors.alauda.io/v1.alpha1/namespaces/{namespace}/connectorssonarqubes/{name}

HTTP method

`DELETE`

Description

delete the specified ConnectorsSonarQube

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 ConnectorsSonarQube

HTTP responses

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

HTTP method

PATCH

Description

partially update the specified ConnectorsSonarQube

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

Parameter	Type	Description
		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>ConnectorsSonarQube</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified ConnectorsSonarQube

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

Parameter	Type	Description
		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	ConnectorsSonarQube	schema
	application/json	formatted

HTTP responses

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

/apis/operator.connectors.alauda.io/v1alpha1/namespaces/{namespace}/connectorssonarqubes/{name}/status

HTTP method

GET

Description

read status of the specified ConnectorsSonarQube

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified ConnectorsSonarQube

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>ConnectorsSonarQube</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified ConnectorsSonarQube

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

HTTP responses

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