Alauda DevOps Connectors Docs

Operator APIs

ConnectorsCore [operator.conn

ConnectorsGitLab [operator.co

Connectors

ConnectorsHarbor [operator.co

ConnectorsK8S [operator.conn

Connectors

ConnectorsNPM [operator.conn

ConnectorsOCI [operator.conn

Connectors

ConnectorsSonarQube [operator.connectors.alauda.io/v1alpha1]

ConnectorsCore [operator.connectors.alauda.io/v1alpha1]

`operator.connectors.alauda.io` group

ConnectorsCore is the Schema for the connectorscore API.

`v1alpha1` version

▼ spec `object`

ConnectorsCoreSpec defines the desired state of ConnectorsCore.

▼ additionalManifests `string`

A means to specify the additional manifests to install, eg: http://example.com/manifests.yaml ↗

▼ annotations `object`

Annotations overrides labels for the namespace and its template.

▼ labels `object`

Labels overrides labels for the namespace and its template.

▼ registry `string`

Registry overrides the default registry for the workloads. The registry is used to replace the prefix of the image. the image reference format is /:@.

workloads `[]object`

WorkloadOverride overrides the workload configurations.

name `string` required

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 `object`

ComponentStatus defines the observed state of Connector

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 `[]object`

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 ↗

▼ 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` required

Status of the condition, one of True, False, Unknown.

▼ type `string` required

Type of condition.

▼ observedGeneration `integer`

ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

ConnectorsGitLab [operator.connectors.alauda.io/v1alpha1]

`operator.connectors.alauda.io` group

ConnectorsGitLab is the Schema for the ConnectorsGitLabs API.

`v1alpha1` version

▼ spec `object`

ConnectorsGitLabSpec defines the desired state of ConnectorsGitLab.

▼ additionalManifests `string`

A means to specify the additional manifests to install, eg: http://example.com/manifests.yaml ↗

▼ annotations `object`

Annotations overrides labels for the namespace and its template.

▼ labels `object`

Labels overrides labels for the namespace and its template.

▼ registry `string`

Registry overrides the default registry for the workloads. The registry is used to replace the prefix of the image. the image reference format is /:@.

▼ workloads `[]object`

WorkloadOverride overrides the workload configurations.

▼ name `string` required

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 `object`

ComponentStatus defines the observed state of Connector

▼ 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 `[]object`

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 ↗

▼ 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` required

Status of the condition, one of True, False, Unknown.

▼ type `string` required

Type of condition.

▼ **observedGeneration** `integer`

ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

ConnectorsGit [operator.connectors.alauda.io/v1alpha1]

`operator.connectors.alauda.io` group

ConnectorsGit is the Schema for the ConnectorsGits API.

`v1alpha1` version

▼ spec `object`

ConnectorsGitSpec defines the desired state of ConnectorsGit.

▼ additionalManifests `string`

A means to specify the additional manifests to install, eg: http://example.com/manifests.yaml

▼ annotations `object`

Annotations overrides labels for the namespace and its template.

▼ labels `object`

Labels overrides labels for the namespace and its template.

▼ registry `string`

Registry overrides the default registry for the workloads. The registry is used to replace the prefix of the image. the image reference format is /:@.

▼ **workloads** `[]object`

WorkloadOverride overrides the workload configurations.

▼ **name** `string` required

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** `object`

ComponentStatus defines the observed state of Connector

▼ **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 `[]object`

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

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` required

Status of the condition, one of True, False, Unknown.

type `string` required

Type of condition.

▼ observedGeneration `integer`

ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

ConnectorsHarbor [operator.connectors.alauda.io/v1alpha1]

`operator.connectors.alauda.io` group

ConnectorsHarbor is the Schema for the ConnectorsHarbors API.

`v1alpha1` version

▼ **spec** `object`

ConnectorsHarborSpec defines the desired state of ConnectorsHarbor.

▼ **additionalManifests** `string`

A means to specify the additional manifests to install, eg: http://example.com/manifests.yaml ↗

▼ **annotations** `object`

Annotations overrides labels for the namespace and its template.

▼ **expose** `object`

Expose defines the expose configuration for the connector

▼ **domain** `string`

Domain specifies the access domain for the connector service. Required when Type is Ingress or NodePort:

- 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.

ingressClassName `string`

IngressClassName specifies the ingress controller class to use. Leave empty to use the default ingress controller. eg: "nginx"

secretName `string`

SecretName specifies the TLS secret for HTTPS access. Example: "connectors-tls-secret"

nodePort `object`

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

port `integer`

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

type `string` required

Type specifies the exposure method for the OCI connector service.

- 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

▼ labels `object`

Labels overrides labels for the namespace and its template.

▼ registry `string`

Registry overrides the default registry for the workloads. The registry is used to replace the prefix of the image. the image reference format is /:@.

▼ workloads `[]object`

WorkloadOverride overrides the workload configurations.

▼ name `string` required

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 `object`

ComponentStatus defines the observed state of Connector

▼ 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 `[]object`

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 ↗

▼ 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` required

Status of the condition, one of True, False, Unknown.

▼ **type** `string` required

Type of condition.

▼ **observedGeneration** `integer`

ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

ConnectorsK8S [operator.connectors.alauda.io/v1alpha1]

`operator.connectors.alauda.io` group

ConnectorsK8S is the Schema for the ConnectorsK8Ss API.

`v1alpha1` version

▼ **spec** `object`

ConnectorsK8SSpec defines the desired state of ConnectorsK8S.

▼ **additionalManifests** `string`

A means to specify the additional manifests to install, eg: http://example.com/manifests.yaml ↗

▼ **annotations** `object`

Annotations overrides labels for the namespace and its template.

▼ **labels** `object`

Labels overrides labels for the namespace and its template.

▼ **registry** `string`

Registry overrides the default registry for the workloads. The registry is used to replace the prefix of the image. the image reference format is /:@.

▼ workloads `[]object`

WorkloadOverride overrides the workload configurations.

▼ name `string` required

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 `object`

ComponentStatus defines the observed state of Connector

▼ 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 `[]object`

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 ↗

▼ 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` required

Status of the condition, one of True, False, Unknown.

▼ type `string` required

Type of condition.

▼ **observedGeneration** `integer`

ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

ConnectorsMaven [operator.connectors.alauda.io/v1alpha1]

`operator.connectors.alauda.io` group

ConnectorsMaven is the Schema for the ConnectorsMavens API.

`v1alpha1` version

▼ spec `object`

ConnectorsMavenSpec defines the desired state of ConnectorsMaven.

▼ additionalManifests `string`

A means to specify the additional manifests to install, eg: http://example.com/manifests.yaml ↗

▼ annotations `object`

Annotations overrides labels for the namespace and its template.

▼ labels `object`

Labels overrides labels for the namespace and its template.

▼ registry `string`

Registry overrides the default registry for the workloads. The registry is used to replace the prefix of the image. the image reference format is /:@.

▼ workloads `[]object`

WorkloadOverride overrides the workload configurations.

▼ name `string` required

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 `object`

ComponentStatus defines the observed state of Connector

▼ 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 `[]object`

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 ↗

▼ 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` required

Status of the condition, one of True, False, Unknown.

▼ type `string` required

Type of condition.

▼ observedGeneration `integer`

ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

ConnectorsNPM [operator.connectors.alauda.io/v1alpha1]

`operator.connectors.alauda.io` group

ConnectorsNPM is the Schema for the ConnectorsNPMs API.

`v1alpha1` version

▼ spec `object`

ConnectorsNPMSpec defines the desired state of ConnectorsNPM.

▼ additionalManifests `string`

A means to specify the additional manifests to install, eg: http://example.com/manifests.yaml ↗

▼ annotations `object`

Annotations overrides labels for the namespace and its template.

▼ labels `object`

Labels overrides labels for the namespace and its template.

▼ registry `string`

Registry overrides the default registry for the workloads. The registry is used to replace the prefix of the image. the image reference format is /:@.

▼ **workloads** `[]object`

WorkloadOverride overrides the workload configurations.

▼ **name** `string` required

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** `object`

ComponentStatus defines the observed state of Connector

▼ **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 `[]object`

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 ↗

▼ 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` required

Status of the condition, one of True, False, Unknown.

▼ type `string` required

Type of condition.

▼ **observedGeneration** `integer`

ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

ConnectorsOCI [operator.connectors.alauda.io/v1alpha1]

`operator.connectors.alauda.io` group

ConnectorsOCI is the Schema for the ConnectorsOCIs API.

`v1alpha1` version

▼ **spec** `object`

ConnectorsOCISpec defines the desired state of ConnectorsOCI.

▼ **additionalManifests** `string`

A means to specify the additional manifests to install, eg: http://example.com/manifests.yaml ↗

▼ **annotations** `object`

Annotations overrides labels for the namespace and its template.

▼ **expose** `object`

Expose defines the expose configuration for the connector

▼ **domain** `string`

Domain specifies the access domain for the connector service. Required when Type is Ingress or NodePort:

- 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.

▼ ingressClassName `string`

IngressClassName specifies the ingress controller class to use. Leave empty to use the default ingress controller. eg: "nginx"

▼ secretName `string`

SecretName specifies the TLS secret for HTTPS access. Example: "connectors-tls-secret"

▼ nodePort `object`

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

▼ port `integer`

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

▼ type `string` required

Type specifies the exposure method for the OCI connector service.

- 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

▼ labels `object`

Labels overrides labels for the namespace and its template.

▼ registry `string`

Registry overrides the default registry for the workloads. The registry is used to replace the prefix of the image. the image reference format is /:@.

▼ workloads `[]object`

WorkloadOverride overrides the workload configurations.

▼ name `string` required

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 `object`

ComponentStatus defines the observed state of Connector

▼ 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 `[]object`

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 ↗

▼ 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` required

Status of the condition, one of True, False, Unknown.

▼ **type** `string` required

Type of condition.

▼ **observedGeneration** `integer`

ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

ConnectorsPyPI [operator.connectors.alauda.io/v1alpha1]

`operator.connectors.alauda.io` group

ConnectorsPyPI is the Schema for the ConnectorsPyPIs API.

`v1alpha1` version

▼ spec `object`

ConnectorsPyPISpec defines the desired state of ConnectorsPyPI.

▼ additionalManifests `string`

A means to specify the additional manifests to install, eg: http://example.com/manifests.yaml ↗

▼ annotations `object`

Annotations overrides labels for the namespace and its template.

▼ labels `object`

Labels overrides labels for the namespace and its template.

▼ registry `string`

Registry overrides the default registry for the workloads. The registry is used to replace the prefix of the image. the image reference format is /:@.

▼ workloads `[]object`

WorkloadOverride overrides the workload configurations.

▼ name `string` required

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 `object`

ComponentStatus defines the observed state of Connector

▼ 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 `[]object`

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 ↗

▼ 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` required

Status of the condition, one of True, False, Unknown.

▼ type `string` required

Type of condition.

▼ observedGeneration `integer`

ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

ConnectorsSonarQube [operator.connectors.alauda.io/v1alpha1]

`operator.connectors.alauda.io` group

ConnectorsSonarQube is the Schema for the ConnectorsSonarQubes API.

`v1alpha1` version

▼ **spec** `object`

ConnectorsSonarQubeSpec defines the desired state of ConnectorsSonarQube.

▼ **additionalManifests** `string`

A means to specify the additional manifests to install, eg: http://example.com/manifests.yaml ↗

▼ **annotations** `object`

Annotations overrides labels for the namespace and its template.

▼ **labels** `object`

Labels overrides labels for the namespace and its template.

▼ **registry** `string`

Registry overrides the default registry for the workloads. The registry is used to replace the prefix of the image. the image reference format is /:@.

▼ workloads `[]object`

WorkloadOverride overrides the workload configurations.

▼ name `string` required

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 `object`

ComponentStatus defines the observed state of Connector

▼ 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 `[]object`

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 ↗

▼ 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` required

Status of the condition, one of True, False, Unknown.

▼ type `string` required

Type of condition.

▼ observedGeneration `integer`

ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.