

Release Notes

Date published: 2024-06-11

Date modified: 2026-07-31



Legal Notice

© Cloudera Inc. 2026. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

Contents

- Release notes.....4**
 - What's New.....4
 - Fixed Issues.....6
 - Known Issues.....7
 - Deprecations.....7
 - Removals.....8
 - Behavioral changes.....9
- Unsupported features..... 10**
- Supported Kafka versions..... 11**
- System requirements..... 12**
- Kafka Connect plugins.....12**

Release notes

Learn about the new features, known and fixed issues, as well as deprecations and removals in this release of Cloudera Streams Messaging Operator for Kubernetes.

What's New

Learn about the new features and notable changes in this release.

Rebase to Strimzi 1.0.1 and Kafka 4.2.0

This release of Cloudera Streams Messaging Operator for Kubernetes is based on Strimzi 1.0.1 (previously 0.49.1) and Kafka 4.2.0 (previously 4.1.1), bringing new features including per-listener Kafka options, advertised port templates, environment variable-based rack awareness, and on-demand `KafkaUser` certificate renewal.

See the following upstream resources for more information on these versions:

- [Strimzi 0.50.0 Release notes](#)
- [Strimzi 0.51.0 Release notes](#)
- [Strimzi 1.0.0 Release notes](#)
- [Strimzi 1.0.1 Release notes](#)
- [Kafka 4.2.0 Release notes](#)
- [Kafka 4.2.0 Notable changes](#)

Strimzi v1 API is now the only supported API version

The v1alpha1, v1beta1, and v1beta2 API versions for Strimzi Custom Resource Definitions (CRDs) are removed in this release. Only the v1 API (`kafka.strimzi.io/v1`) is supported.

Custom resources still using old API versions will stop working after upgrading to this release. Before upgrading, convert all Strimzi custom resources to the v1 API using the Strimzi API conversion tool shipped with Cloudera Streams Messaging Operator for Kubernetes. Additionally, run the `crd-upgrade` command to finalize the API conversion at the CRD level before upgrading Strimzi.

For conversion instructions, see [Converting Strimzi custom resource to the v1 API](#)

Strimzi Drain Cleaner

Cloudera Streams Messaging Operator for Kubernetes now ships Strimzi Drain Cleaner as an optional, separately installable component. Strimzi Drain Cleaner helps you safely drain Kubernetes nodes that run Strimzi-managed Kafka Pods during cluster maintenance or upgrades. It uses a validating admission webhook to coordinate rolling Pod restarts through the Strimzi Cluster Operator, helping keep Kafka partitions in sync while nodes are drained.

For more information, see the following resources:

- [Deployment architecture](#)
- [Installing Strimzi Drain Cleaner with Helm](#)
- [Strimzi Drain Cleaner overview](#)
- [Strimzi Drain Cleaner Helm chart configuration reference](#)

OperatorHub installation on OpenShift is discontinued

Cloudera Streams Messaging Operator for Kubernetes no longer ships an Operator Lifecycle Manager (OLM) bundle. New releases are not published to OperatorHub in OpenShift. Use Helm for installation instead.

For installation instructions, see [Installation overview](#).

Minimum Kubernetes version is now 1.30

The minimum supported Kubernetes version is raised to 1.30. Kubernetes 1.27, 1.28, and 1.29 are no longer supported. The minimum supported OpenShift version is raised to 4.17.

For the full list of system requirements, see [System requirements](#).

report.sh captures Cloudera Surveyor data

The report.sh diagnostic tool now discovers and includes diagnostic data for Cloudera Surveyor instances by default. The bundle captures the full set of Kubernetes resources associated with each Cloudera Surveyor instance, including Deployments, ConfigMaps, Secrets, Services, ReplicaSets, Pods, and pod logs.

For more information, see [Diagnostics](#).

Kafka options are now configurable per listener

The following Kafka options can now be configured individually for each listener in the `Kafka` resource, rather than only at the broker level via `spec.kafka.config`:

- `connections.max.reauth.ms`
- `max.connections`
- `max.connections.per.ip`
- `max.connections.per.ip.overrides`
- `max.connection.creation.rate`

Per-listener configuration enables more granular control. For example, session re-authentication can be enforced on a specific external listener without affecting internal listeners. For more information, see [Listener configuration options](#).

Advertised port template for listener configuration

A new `advertisedPortTemplate` field in the listener configuration controls how Strimzi generates the advertised ports for individual brokers. The field accepts a simple mathematical expression using the `{nodeId}` placeholder and the `+`, `-`, and `*` operators. This is useful for external listener types such as `nodeport` and `loadbalancer`, where clients connect directly to specific brokers and need predictable per-broker port numbers. For more information, see [Listener configuration options](#).

Environment variable-based rack awareness

A new type: environment-variable option is available for rack awareness configuration in the `Kafka` resource. Instead of reading rack information from Kubernetes node labels, which requires `ClusterRoleBindings` for the Strimzi Cluster Operator, the rack value is derived from an environment variable present in the broker pod.

This makes rack awareness available in environments where granting `ClusterRoleBindings` to the Strimzi Cluster Operator is restricted. For more information, see [Rack awareness](#).

Force-renew KafkaUser certificates on demand

A new `strimzi.io/force-renew` annotation on the user `Secret` triggers immediate renewal of a `KafkaUser` certificate. This is useful when a certificate needs to be rotated ahead of its normal renewal schedule, for example after a security incident or when a certificate must be rotated for operational reasons. For more information, see [Renewing KafkaUser certificates](#).

Client Configurations download in Cloudera Surveyor

Cloudera Surveyor adds a **Client Configurations** tab on the **Cluster Details** page for each monitored Kafka cluster. From this tab, end users can preview, copy, and download files that you configure for that cluster. The typical use case is exposing connection material for external Kafka clients, such as client configuration properties and CA certificates.

The tab is empty by default and no files are available for download. You configure what is available by storing file content in Kubernetes **Secrets** and referencing them using `clusterConfigs.clusters[*].clientConfigs` properties in your values file.

- For configuration instructions, see [Exposing client configurations for download](#).
- For a full list of available properties, see [Cloudera Surveyor Helm chart configuration reference](#).
- For an overview of the **Client Configurations** tab and its download options, see [Downloading client configurations](#).

Controller column visibility and metadata store display in Cloudera Surveyor

The **Controller** column on the **Brokers** page and in the Cluster Details **Brokers** tab is not displayed for KRaft-based clusters. A new **Metadata Store** property is now visible in the cluster details drawer, explicitly indicating whether the cluster uses KRaft or ZooKeeper.

Configurable instance name in Cloudera Surveyor

You can now configure an instance name for Cloudera Surveyor using the new `instanceName` property. When set, the name is displayed on the UI. This is useful when running multiple Cloudera Surveyor instances, as the name allows for quick differentiation between instances.

Read-only broker configuration in Cloudera Surveyor

A new `clusterConfigs.clusters[*].readOnlyBrokerConfigs` property allows you to disable broker configuration editing for individual registered Kafka clusters. When set to true, the Edit Properties button on **Broker Details** pages is disabled and broker configuration properties cannot be modified through the Cloudera Surveyor UI or API.

This is particularly useful for Kafka clusters managed by the Strimzi Cluster Operator, where any configuration changes applied through Cloudera Surveyor would be overwritten during the next Strimzi reconciliation cycle. Setting this property to true for Strimzi-managed clusters is highly recommended. For more information, see [Disabling broker configuration editing](#).

Redesigned Cloudera Surveyor user interface

The Cloudera Surveyor user interface has been redesigned. All pages, tables, navigation elements, filters, controls, and icons have a new look and feel. All existing functionality, navigation structure, and workflows remain unchanged.

Updated Entity Details drawer in Cloudera Surveyor

The **Entity Details** drawer has been updated with the following improvements:

- Attributes are now organized into named sections that vary by entity type, replacing the previous flat attribute list.
- After the drawer is opened, a collapsible strip persists on the right edge of the page until you navigate away, so you can reopen the drawer without reselecting an entity.

For more information, see [Entity Details drawer](#).

Fixed Issues

Learn what issues are fixed in this release.

Strimzi

None.

Kafka

None.

Cloudera Surveyor

CSMDS-1697: Snapshot errors when Connect or MirrorMaker 2 groups are present on a Kafka cluster

Fixed a bug where Cloudera Surveyor incorrectly attempted to describe consumer groups that use the connect protocol on Kafka clusters running Kafka Connect or MirrorMaker 2, causing snapshot failures.

CSMDS-1825: Topic name field not reset or focused after topic creation

Fixed a bug where the Topic Name field was not cleared or focused after creating a topic. The field is now reset and automatically focused after each successful topic creation.

CSMDS-1821: Browser back navigation looping on detail pages

Fixed a bug where the browser Back button would loop instead of navigating back on Topic, Broker, Cluster, Consumer Group, and Producer Details pages.

CSMDS-1820: Missing placeholder for unset broker rack

Fixed a bug where the Rack field on the Broker Details page showed nothing when a broker had no rack configured. The field now displays a placeholder value.

CSMDS-1780: Stuck time counter and incorrect tooltips on Configuration tabs

Fixed a bug where the time-since-last-update counter on Configuration tabs was frozen when auto-update was disabled. Additionally corrected the tooltip texts on Configuration tabs to reflect that configuration data is fetched live and is not based on a Kafka snapshot.

Schema Registry

None.

Known Issues

Learn about the known issues in this release.

Strimzi

None.

Kafka

None.

Cloudera Surveyor

None.

Schema Registry

None.

Deprecations

Learn what is deprecated in this release.

Strimzi

The ingress listener type

The ingress listener type is deprecated following the archiving of the Ingress NGINX Controller project. Plan migration to an alternative external listener type: nodeport, loadbalancer, or route (OpenShift only).

Kafka

None.

Cloudera Surveyor

None.

Schema Registry

None.

Removals

Learn what is removed in this release.

Strimzi

Strimzi API versions v1alpha1, v1beta1, and v1beta2

The v1alpha1, v1beta1, and v1beta2 API versions for Strimzi Custom Resources are removed. Only the v1 API (kafka.strimzi.io/v1) is supported. Custom resources still using old API versions will stop working after upgrading. Before upgrading, convert all custom resources to the v1 API using the Strimzi API conversion tool shipped with Cloudera Streams Messaging Operator for Kubernetes.

The type: oauth authentication in Kafka

The type: oauth authentication type for Kafka listeners and Kafka Connect is removed. It was only available in the v1beta2 API and is not present in the v1 API. Use type: custom instead to configure OAuth authentication.

Mounting additional Secrets in the custom authentication type

Support for the spec.kafka.listeners[*].authentication.secrets property in the custom authentication type is removed. Use the additional volumes and volume mounts instead to mount additional Secrets. For more information, see [Configuring additional volumes and volume](#).

Storage overrides

Support for storage overrides for persistent volumes is removed. The spec.storage.overrides property in the KafkaNodePool resource is no longer supported. If you want to configure storage classes on a per-broker basis, deploy multiple KafkaNodePool resources with a different storage class each.

CPU and memory configuration in the Kafka resource

The spec.kafka.resources property is removed from the v1 API. Define CPU and memory requests and limits in KafkaNodePool resources instead.

spec.externalConfiguration in KafkaConnect resources

The spec.externalConfiguration property is removed from the v1 API. Use spec.template to configure additional environment variables or volumes in Connect worker pods instead. The Strimzi API conversion tool does not migrate this property automatically. For more information, see [Adding external configuration to Kafka Connect worker pods](#).

Kafka

None.

Cloudera Surveyor

None.

Schema Registry

None.

Behavioral changes

Learn about the behavioral changes in this release.

Strimzi

Buildah is now the default builder for Kafka Connect custom images

Previous Behavior: Kaniko was used to build custom Kafka Connect images when `spec.build` was configured in a `KafkaConnect` resource.

New Behavior: Buildah is now the default builder. The `spec.build` configuration syntax in `KafkaConnect` resources does not change. Air-gapped users who use the Kafka Connect image build who use `spec.build` in `KafkaConnect` resources to automatically build new Kafka images must mirror the Buildah image before upgrading. If the Buildah image is not available in the self-hosted registry after upgrading, the image build will fail. The Buildah image is available at `container.repository.cloudera.com/cloudera/buildah:1.0.1-csmop-1.7.0-b248`.



Note: If you configured advanced build or push options using `additionalBuildOptions` or `additionalPushOptions` in your `KafkaConnect` resource, you might need to update your configuration as the accepted CLI flags differ between Kaniko and Buildah. Review the Buildah options listed in the [Strimzi API reference](#) and update any Kaniko-specific flags.



Note: Kaniko remains available. If needed, you can switch back by disabling the `UseConnectBuildWithBuildah` Strimzi feature gate. For more information, see [Feature gates in Strimzi](#).

Server-side apply is now enabled by default

Previous Behavior: Strimzi used client-side apply when managing Kubernetes resources.

New Behavior: The `ServerSideApplyPhase1` feature gate is now enabled by default. Strimzi uses server-side apply when managing Kubernetes resources. For standard deployments this change is transparent. Users who manually modify Strimzi-managed resources or use GitOps tools that also apply these resources may encounter field manager conflicts.

Entity Operator cross-namespace watching is disabled by default

Previous Behavior: In previous versions, when you configured `watchedNamespace` in your Entity Operator (Topic Operator or User Operator) to watch a different namespace than the Kafka cluster namespace, cross-namespace watching was enabled automatically.

New Behavior: Cross-namespace watching is now disabled by default. To watch a namespace different from the Kafka cluster namespace, you must explicitly enable it by setting the `STRIMZI_ENTITY_OPERATOR_WATCHED_NAMESPACE_ENABLED=true` environment variable in your Cluster Operator `Deployment`.

Enabling cross-namespace watching extends the Entity Operator's access beyond the Kafka cluster namespace, including full create, read, update, and delete access to `Secret` resources in the watched namespace. Enable it only when you need this configuration and accept that cross-namespace trust boundary.

If you are upgrading from a previous version and have `watchedNamespace` configured to a different namespace than your Kafka cluster, you must set `STRIMZI_ENTITY_OPERATOR_WATCHED_NAMESPACE_ENABLED=true` during the upgrade. If you upgrade without enabling this environment variable, the Entity Operator `Deployment` and all related resources will be deleted during reconciliation, and your topics and users will no longer be managed until you enable cross-namespace watching. For upgrade instructions, see [Upgrading and rolling back Cloudera Streams](#)

[Messaging Operator for Kubernetes](#). For configuration instructions, see [Deploying and configuring the Strimzi Entity Operator](#).

Kafka

None.

Cloudera Surveyor

HTTP listener is disabled and default service port changes when TLS is enabled

Previous Behavior: When `tlsConfigs.enabled` was set to true, the HTTP listener on port 8080 remained active alongside the HTTPS listener. The default value of `service.port` was 8080.

New Behavior: The HTTP listener is disabled when TLS is enabled. Cloudera Surveyor accepts only HTTPS traffic on the configured TLS port. The default value of `service.port` is now 8443 to match the default TLS configuration. The liveness and readiness probes in the Helm-managed `Deployment` are updated automatically to use HTTPS. The HTTP listener was only reachable from within the Kubernetes cluster, so external traffic was not affected by its presence. If you have TLS enabled and have not explicitly set `service.port` in your values file, verify that any in-cluster tooling such as custom health checks or monitoring agents targeting port 8080 over HTTP are updated to use port 8443 over HTTPS.

Default service type is now ClusterIP

Previous Behavior: The Helm chart created a `NodePort` type `Service` by default, making the Cloudera Surveyor UI reachable from outside the Kubernetes cluster without additional configuration.

New Behavior: The Helm chart now creates a `ClusterIP` type `Service` by default. The UI is not reachable externally without configuring an `Ingress` or a `LoadBalancer` type `Service`. If you are upgrading and relied on the `NodePort` default, configure external access using `Ingress` or `LoadBalancer`, or explicitly set `service.type: NodePort` in your values file.

`service.tlsPort` and `ingress.rules.port` properties removed

Previous Behavior: Two separate port properties existed: `service.tlsPort` configured a dedicated TLS port on the `Service`, and `ingress.rules.port` configured which `Service` port the `Ingress` forwarded traffic to.

New Behavior: Both properties are removed. Only `service.port` configures the `Service` port, and the `Ingress` forwards to `service.port` automatically. Remove `service.tlsPort` and `ingress.rules.port` from any custom values files.

LDAP authentication properties are now nested under the `ldap` key

Previous Behavior: LDAP authentication properties were configured directly under `surveyorConfig.ig.surveyor.authentication`, for example `surveyorConfig.surveyor.authentication.enabled`.

New Behavior: All LDAP authentication properties are now nested one level deeper under a `ldap` key, for example `surveyorConfig.surveyor.authentication.ldap.enabled`. Update any custom values files or automation scripts that reference the old property paths.

Schema Registry

None.

Unsupported features

Learn what features are unsupported in this release.

Strimzi

- Kafka MirrorMaker 2
- Kafka Bridge

Kafka

None.

Cloudera Surveyor

- The REST API

Calling the REST API directly using any kind of tooling or using it programmatically is not supported. At this time, the API is evolving and is subject to major, backward incompatible changes.

Schema Registry

- The Schema Registry UI

The Schema Registry UI is available for unauthenticated technical previews or development environments only. Because the UI does not currently support OAuth login flows, it is inaccessible when authentication is enabled. For production environments where security is required, use the Schema Registry REST API.

Supported Kafka versions

Kafka version strings supported by Cloudera Streams Messaging Operator for Kubernetes for use in `Kafka` and `KafkaConnect` resources, and the associated Kafka metadata versions.

Kafka versions shipped in Cloudera Streams Messaging Operator for Kubernetes are specific to Cloudera. You specify them in the `spec.version` property of cluster resources like `Kafka` and `KafkaConnect` resources. When deploying a cluster, you must use the versions listed here. Specifying upstream Apache Kafka versions is not supported.

Table 1: Supported Kafka versions

Version	Component/Resource	Metadata version
4.2.0-csmop-1.7 (latest and default)	• <code>Kafka</code> – <code>Kafka</code> resources • <code>Kafka Connect</code> – <code>KafkaConnect</code> resources	4.2-IV1
4.1.1.1.6	• <code>Kafka</code> – <code>Kafka</code> resources • <code>Kafka Connect</code> – <code>KafkaConnect</code> resources	4.1-IV1

The latest version is the current and latest supported version. This version is used by default to deploy clusters if an explicit version is not provided in your resource configuration. This version is also the version recommended by Cloudera. All other versions listed here are Kafka versions shipped in previous releases of Cloudera Streams Messaging Operator for Kubernetes that are also supported.

The Kafka metadata version is relevant for upgrades. Depending on your specific upgrade path, explicitly specifying the metadata version might be necessary during an upgrade.

System requirements

Cloudera Streams Messaging Operator for Kubernetes requires a Kubernetes cluster environment that meets the following system requirements and prerequisites. Ensure that you meet these requirements, otherwise, you will not be able to install and use Cloudera Streams Messaging Operator for Kubernetes or its components.

- A Kubernetes 1.30 or later cluster.
 - OpenShift 4.17 or later.
 - Any Cloud Native Computing Foundation (CNCF) certified Kubernetes distribution. For more information, see cncf.io.
- Administrative rights on the Kubernetes cluster.
- Access to kubectl or oc. These command line tools must be configured to connect to your running cluster.
- Access to helm version 3.14.0 or higher.
- Log collection is enabled for the Kubernetes cluster.

Cloudera requires that the logs of Cloudera Streams Messaging Operator for Kubernetes components are stored long term for diagnostic and supportability purposes. Collect logs using the log collector feature of your Kubernetes platform. Cloudera recommends at least one week of retention time for the collected logs.

- A persistent storage class configured on the Kubernetes cluster that satisfies the durability and low-latency requirements for operating Kafka.

The ideal storage class configuration can vary per environment and use-case and is determined by the Kubernetes platform where Cloudera Streams Messaging Operator for Kubernetes is deployed.

Additionally, for Kafka brokers, Cloudera recommends a `StorageClass` that has volume expansion enabled (`allowvolumeexpansion` set to `true`).

- A [Prometheus](#) installation running in the same Kubernetes cluster where you install Cloudera Streams Messaging Operator for Kubernetes is recommended. Prometheus is used for collecting and monitoring Kafka and Strimzi metrics.

Kafka Connect plugins

Learn about the third-party connectors that Cloudera certified for use and the various default connectors, transforms, and converters that are included in Cloudera Streams Messaging Operator for Kubernetes.

Certified third-party connectors

Certified connectors are validated to work with Cloudera Streams Messaging Operator for Kubernetes but do not come packaged with it or are installed to Kafka Connect by default.

Table 2: Certified connectors in Cloudera Streams Messaging Operator for Kubernetes

Connector	Certified Version	Download location	Documentation
Apache Iceberg Sink Connector for Kafka Connect	1.10	Maven Central	Kafka Connect Apache Iceberg

Plugins included in Cloudera Streams Messaging Operator for Kubernetes Connectors

Cloudera Streams Messaging Operator for Kubernetes ships and supports all Kafka Connect connectors included in Apache Kafka.

The full list is as follows.

- `org.apache.kafka.connect.mirror.MirrorCheckpointConnector`
- `org.apache.kafka.connect.mirror.MirrorSourceConnector`
- `org.apache.kafka.connect.mirror.MirrorHeartBeatConnector`
- `org.apache.kafka.connect.file.FileStreamSourceConnector`
- `org.apache.kafka.connect.file.FileStreamSinkConnector`

**Note:**

Although both `FileStreamSourceConnector` and `FileStreamSinkConnector` are shipped with Cloudera Streams Messaging Operator for Kubernetes, neither connector is installed, and you cannot deploy them by default. To deploy instances of these connectors, you must first install them as any other third-party connector. Cloudera also does not recommend that you use these connectors in production.

Single Message Transforms plugins (transformations and predicates)

Single Message Transforms (SMT) plugins (transformations and predicates) are deployed on top of Kafka Connect connectors. They enable you to apply message transformations and filtering on a single message basis. Cloudera Streams Messaging Operator for Kubernetes ships and supports all transformation and predicates plugins included in Apache Kafka as well as the `ConvertFromBytes` and `ConvertToBytes` plugins, which are Cloudera specific plugins.

The full list is as follows.

Transformations

- `com.cloudera.dim.kafka.connect.transforms.ConvertFromBytes`
- `com.cloudera.dim.kafka.connect.transforms.ConvertToBytes`
- `org.apache.kafka.connect.transforms.Cast`
- `org.apache.kafka.connect.transforms.DropHeaders`
- `org.apache.kafka.connect.transforms.ExtractField`
- `org.apache.kafka.connect.transforms.Filter`
- `org.apache.kafka.connect.transforms.Flatten`
- `org.apache.kafka.connect.transforms.HeaderFrom`
- `org.apache.kafka.connect.transforms.HoistField`
- `org.apache.kafka.connect.transforms.InsertField`
- `org.apache.kafka.connect.transforms.InsertHeader`
- `org.apache.kafka.connect.transforms.MaskField`
- `org.apache.kafka.connect.transforms.RegexRouter`
- `org.apache.kafka.connect.transforms.ReplaceField`
- `org.apache.kafka.connect.transforms.SetSchemaMetadata`
- `org.apache.kafka.connect.transforms.TimestampConverter`
- `org.apache.kafka.connect.transforms.TimestampRouter`
- `org.apache.kafka.connect.transforms.ValueToKey`

Predicates

- `org.apache.kafka.connect.transforms.predicates.HasHeaderKey`
- `org.apache.kafka.connect.transforms.predicates.RecordIsTombstone`
- `org.apache.kafka.connect.transforms.predicates.TopicNameMatches`

Converters

Converters can be used to transform Kafka record keys and values between bytes and a specific format. In addition to the `JsonConverter`, there are converters for most often used primitive types as well.

The full list is as follows.

- `org.apache.kafka.connect.json.JsonConverter`
- `org.apache.kafka.connect.converters.ByteArrayConverter`
- `org.apache.kafka.connect.converters.BooleanConverter`
- `org.apache.kafka.connect.converters.DoubleConverter`
- `org.apache.kafka.connect.converters.FloatConverter`
- `org.apache.kafka.connect.converters.IntegerConverter`
- `org.apache.kafka.connect.converters.LongConverter`
- `org.apache.kafka.connect.converters.ShortConverter`
- `org.apache.kafka.connect.storage.StringConverter`

Header converters

Header converters can be used to transform Kafka record headers between bytes and a specific format. Cloudera Streams Messaging Operator for Kubernetes and Kafka includes a single dedicated header converter, which is the `org.apache.kafka.connect.storage.SimpleHeaderConverter`.

The `SimpleHeaderConverter` is the default header converter and is adequate for the majority of use cases. In case your headers are of a specific format, like JSON, you can use any other converter listed in the [Converters](#) on page 13 section as a header converter as well.

Replication policies

A replication policy defines the basic rules of how topics are replicated from source to target clusters when using Kafka Connect-based replication to replicate Kafka data between Kafka clusters.

The full list is as follows.

- `org.apache.kafka.connect.mirror.DefaultReplicationPolicy`
- `org.apache.kafka.connect.mirror.IdentityReplicationPolicy`

Related Information

[Installing Kafka Connect connector plugins](#)

[ConvertFromBytes](#)

[ConvertToBytes](#)

[Included transformations | Kafka](#)

[Predicates | Kafka](#)