

Release Notes

Date published: 2024-06-11

Date modified: 2025-10-30



Legal Notice

© Cloudera Inc. 2025. 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.....5
 - Known Issues.....5
 - Unsupported features.....5
 - Deprecations and removals.....5
 - Behavioral changes.....6
- Component versions..... 7**
- System requirements..... 8**
- Kafka Connect plugins.....9**

Release notes

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

What's New

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

Rebase to Strimzi 0.47.0 and Kafka 4.0.1

This release of Cloudera Streams Messaging - Kubernetes Operator is based on Strimzi 0.47.0 (previously 0.45.0) and Kafka 4.0.1 (previously 3.9.0).

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

- [Strimzi 0.46.0 Release notes](#)
- [Strimzi 0.47.0 Release notes](#)
- [Kafka 4.0.0 Release notes](#)
- [Kafka 4.0.0 Notable changes](#)
- [Kafka 4.0.1 Notable changes](#)
- [Kafka 4.0.1 Release notes](#)

ZooKeeper is removed

ZooKeeper and support for ZooKeeper-based Kafka clusters is removed. You can now only run Kafka clusters in KRaft mode. In addition, migrating ZooKeeper-based Kafka clusters to KRaft is no longer supported.



Important: Before you upgrade, migrate all your ZooKeeper-based Kafka clusters to KRaft. Upgrading a ZooKeeper-based Kafka cluster to this version is not supported. For more information, see [Migrating Kafka clusters from ZooKeeper to KRaft](#).

FIPS 140-2/3 compatibility and compliance

Cloudera Streams Messaging - Kubernetes Operator components are FIPS 140-2/3 compliant and can run in FIPS mode to meet strict security requirements.

- Strimzi and its managed components automatically enable FIPS mode on FIPS-enabled hosts.
- Cloudera Surveyor is FIPS compliant, but requires you to manually enable FIPS mode through configuration. Cloudera Surveyor also supports the use of custom security providers.

For more information, see [FIPS mode](#).

Cloudera Surveyor improvements

- Added two new configuration properties that you can use to specify a regular expression filter to hide topics.

This feature is useful to hide technical or internal topics that are not relevant for most Cloudera Surveyor users. Set a global filter with `surveyorConfig.surveyor.globalHiddenTopicNamePattern` or set per-cluster overrides with `clusterConfigs.clusters[*].hiddenTopicNamePattern`. Additionally, the Topics page on the UI now includes a filter to show either visible or hidden topics.

- Added new filtering options to the Edit Committed Offsets modal.

The filtering options allow you to toggle between filtering for Subscribed Topics or All Topics. Subscribed Topics is selected by default. You can access the modal on the Consumer Group Details Partition Assignment tab. The filtering options are available for the Topics and Topics & Partitions scopes.

- Improved how unsaved changes are handled on the `Topic Details Configuration` tab.

Unsaved changes are now restored automatically when returning to the tab, and a warning dialog is displayed when navigating away from the tab. A visual indicator is also displayed next to the tab name to show when you have unsaved changes.

- Improved status and alert displays throughout the UI.

Status formatting is more consistent, warning and alert icons are clearer, Unknown and Unreachable states are more expressive, and redundant filters have been removed.

- Improved tooltips for all data refresh controls including the refresh, pause, and stop buttons, the refresh interval selector, and the Data as of pill.
- Added a tooltip on the `Topic Details Configuration` tab explaining what the Overridden label means.
- Multiple selection drop-down lists now allow selecting multiple items without closing.
- Operation names in filtering dialogs are now spelled out rather than abbreviated.

Fixed Issues

Learn what issues are fixed in this release.

CSMDS-334: ZooKeeper pods are running but Kafka pods are not created

No longer applicable. ZooKeeper is removed.

CSMDS-953: Kafka and ZooKeeper might experience downtime during upgrades

No longer applicable. ZooKeeper is removed.

Known Issues

There are no known issues in this release.

Unsupported features

Learn what features are unsupported in this release.

Strimzi

- Kafka MirrorMaker
- Kafka MirrorMaker 2
- Kafka Bridge
- Kafka cluster creation without using `KafkaNodePool` resources

Cloudera Surveyor

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.

Deprecations and removals

Learn what is deprecated or removed in this release.

Deprecations

Mounting additional Secrets in the custom authentication type

Support for the `spec.kafka.listeners[*].authentication.secrets` property in the custom authentication type is deprecated and will be removed in a future release. Use the additional volumes and volume

mounts instead to mount additional `Secrets`. For more information, see [Configuring additional volumes and volume](#).

Storage overrides

Configuring storage overrides for persistent volumes is deprecated. The `spec.storage.overrides` property in the `KafkaNodePool` resource is now ignored. If you want to configure storage classes on a per-broker basis, deploy multiple `KafkaNodePool` resources with a different storage class each. You can find more details about migrating from storage overrides in [Configuring Kafka storage with ZooKeeper | Strimzi](#)

Removals

ZooKeeper

ZooKeeper and support for ZooKeeper-based Kafka clusters is removed. You can now only run Kafka clusters in KRaft mode. In addition, migrating ZooKeeper-based Kafka clusters to KRaft is no longer supported.



Important: Before you upgrade, migrate all your ZooKeeper-based Kafka clusters to KRaft. Upgrading a ZooKeeper-based Kafka cluster to this version is not supported. For more information, see [Migrating Kafka clusters from ZooKeeper to KRaft](#).

Behavioral changes

Learn about the behavioral changes in this release.

Strimzi

Truststore location and format change for Kafka Connect

Previous Behavior: For deployments where both Kafka Connect and the Kafka cluster it connects to are managed by Strimzi, Kafka Connect used a PKCS12 truststore file for trusted certificates. The truststore was loaded from disk and configured using the following properties:

- `ssl.truststore.location`
- `ssl.truststore.password`
- `ssl.truststore.type=PKCS12`

New Behavior: Kafka Connect now uses PEM-encoded certificates loaded from a Kubernetes `Secret` for trusted certificates. The truststore is configured with the following properties:

- `ssl.truststore.certificates`
- `ssl.truststore.type=PEM`
- No password property is required for PEM truststores.

As a result of this behavioral change, the process for deploying a replication flow has also changed in the following ways:

- Truststore references must use the `*.ssl.truststore.certificates` property rather than `*.ssl.truststore.location`.
- Mounting the source `Secret` as a volume is no longer required.
- Configuring truststore passwords is no longer required.
- Truststore certificates are referenced directly from `Secrets`, not from files or `Configmaps`.

For updated instructions see [Deploying a replication flow](#).

Kafka

Kafka uses log4j2 instead of Reload4j/Log4j1 for logging

Previous Behavior: Kafka 3.9 and lower used Reload4j/Log4j1 for logging.

New Behavior: Kafka 4.0 later uses Log4j2 for logging. If you have any custom logging configuration, you might need to update it during the upgrade to Kafka 4.0.

Explicitly configuring OAuth and LDAP allowed URLs is now required

Previous Behavior: The `org.apache.kafka.sasl.oauthbearer.allowed.urls` and `com.cloudera.kafka.ldap.allowed.urls` variables were used to restrict access to specific authentication servers only. If the variables were left empty, Kafka could connect to any LDAP or OAuth server.

New Behavior: If these variables are left empty, Kafka is not allowed to connect to any LDAP or OAuth server URL. If you use LDAP or OAuth authentication, update your configuration and add your server URLs. Configure these variables in your `KafkaNodePool` resource using the `spec.jvmOptions.javaSystemProperties` property.

```
#...
kind: KafkaNodePool
spec:
  jvmOptions:
    javaSystemProperties:
      - name: com.cloudera.kafka.ldap.allowed.urls
        value: http://www.ldap-example-1.com,http://www.ldap-example-2.com
      - name: org.apache.kafka.sasl.oauthbearer.allowed.urls
        value: http://www.oauth-example-1.com,http://www.oauth-example-2.com
```

Cloudera Surveyor

Ingress annotations are no longer automatically added

Previous Behavior: The `nginx.ingress.kubernetes.io/backend-protocol` and `cert-manager.io/issuer` annotations were automatically added to the `Ingress` resource. The `cert-manager.io/issuer` annotation was only added if the `tls.enabled` and `tls.issuer` properties were configured.

New Behavior: The `nginx.ingress.kubernetes.io/backend-protocol` and `cert-manager.io/issuer` annotations are no longer added automatically. Use the `ingress.extraAnnotations` property to configure custom annotations.

LDAP entries are now cached by default for five minutes

Previous Behavior: LDAP caching was disabled by default. LDAP entries were not cached.

New Behavior: LDAP caching is enabled by default. Entries are cached for five minutes.

Component versions

A list of components and their versions shipped in this release of Cloudera Streams Messaging - Kubernetes Operator.

Table 1: Cloudera Streams Messaging - Kubernetes Operator component versions

Component	Version
Cloudera Surveyor for Apache Kafka	0.1.0.1.5.0-b123
Cruise Control	2.5.145.1.5.0-b123
Kafka	4.0.1.1.5.0-b123
Strimzi	0.47.0.1.5.0-b123

Supported Kafka versions

Cloudera Streams Messaging - Kubernetes Operator supports the following Kafka versions:

Table 2: Supported Kafka versions

Version	Component/Resource	Metadata version
4.0.1.1.5 (latest and default)	- Kafka – Kafka resources - Kafka Connect – KafkaConnect resources	4.0-IV3
3.9.0.1.4	- Kafka – Kafka resources - Kafka Connect – KafkaConnect resources	3.9-IV0

Kafka versions shipped in Cloudera Streams Messaging - Kubernetes Operator are specific to Cloudera. You specify them in the `spec.version` property of cluster resources like `Kafka` and `KafkaConnect` resources.

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 - Kubernetes Operator that are also supported.

The Kafka version is made up of two parts. The first three digits specify the Apache Kafka version. The digits following the Apache Kafka version specify the major and minor version of Cloudera Streams Messaging - Kubernetes Operator. When deploying a cluster, you must use the versions listed here. Specifying upstream versions is not 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 - Kubernetes Operator 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 - Kubernetes Operator or its components.

- A Kubernetes 1.25 or later cluster.
 - OpenShift 4.12 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 - Kubernetes Operator 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 - Kubernetes Operator 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 - Kubernetes Operator is recommended. Prometheus is used for collecting and monitoring Kafka and Strimzi metrics.

Kafka Connect plugins

Learn what Kafka Connect plugins are shipped with and supported in Cloudera Streams Messaging - Kubernetes Operator.

Connectors

Cloudera Streams Messaging - Kubernetes Operator 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 - Kubernetes Operator, 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 - Kubernetes Operator 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 - Kubernetes Operator 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 10 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](#)

[Transformations | Kafka](#)

[Predicates | Kafka](#)