Cloudera Streams Messaging Operator for Kubernetes 1.6.0

Release Notes

Date published: 2024-06-11
Date modified: 2026-01-27

https://docs.cloudera.com/

Legal Notice

Contents

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 0.49.1 and Kafka 4.1.1

This release of Cloudera Streams Messaging Operator for Kubernetes is based on Strimzi 0.49.1 (previously 0.47.0) and Kafka 4.1.1 (previously 4.0.1).

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

- Strimzi 0.48.0 Release notes
- Strimzi 0.49.0 Release notes
- Strimzi 0.49.1 Release notes
- Kafka 4.1.0 Release notes
- Kafka 4.1.0 Notable changes
- Kafka 4.1.1 Release notes
- Kafka 4.1.1 Notable changes

Schema Registry

Schema Registry is a standalone application that allows you to efficiently store and manage schemas for your streaming data. It supports Avro and JSON schema formats, and provides schema evolution capabilities with configurable compatibility modes. Schema Registry helps ensure data consistency across your streaming applications by providing a central repository of schemas that applications use to validate their data. It provides a REST API and client libraries for programmatic access.

Schema Registry is a new component introduced in this release of Cloudera Streams Messaging Operator for Kubernetes. It is proprietary Cloudera software based on Cloudera Runtime.

See the following resources for more information:

- Installing Schema Registry with Helm
- What is Schema Registry?
- Configuring Schema Registry
- Securing Schema Registry
- Schema Registry REST API reference

Apache Iceberg Sink Connector for Kafka Connect

The Apache Iceberg Sink Connector is now certified for use with Cloudera Streams Messaging Operator for Kubernetes.

The connector must be installed from an external repository (for example, Maven Central) as it is not packaged or shipped by Cloudera.

To get started, follow the End-to-end example deployment with Nessie and AWS. In addition, you can review the following resources for more information:

- Apache Iceberg Kafka Connect documentation
- iceberg-kafka-connect on Maven Central

- Installing Kafka Connect plugins

New Brokers and Broker Details pages in Cloudera Surveyor for Apache Kafka

New **Brokers** and **Broker Details** pages are introduced in Cloudera Surveyor. The **Brokers** page displays all monitored brokers in a selected cluster, while individual **Broker Details** pages provide comprehensive information for each broker.

You can access these pages from the new Brokers menu item in the left navigation. The pages provide you with the following monitoring and management capabilities:

Brokers

- View overall status, health, disk usage, alert information, and more for all brokers within a selected cluster.
- Filter and sort brokers.
- Navigate to any individual broker to open its **Broker Details** page.

Broker Details

- View in-depth information about the status and health of an individual broker.
- View all active alerts specific to the selected broker, with filtering and sorting capabilities.
- View and edit broker configurations.
- Access detailed information about log directories, including disk usage and replica distribution.
- Track all topic partitions and replicas hosted on the broker.

Authentication keys are now generated automatically in Cloudera Surveyor

Authentication setup for Cloudera Surveyor is now simplified with automatic key generation. The surveyorConfig.surveyor.authentication.keys.active property now defaults to a cryptographically secure random 128-byte value when not explicitly configured. This eliminates the need to manually generate and configure a key when enabling authentication. For more information, see Authentication in Cloudera Surveyor for Apache Kafka.

Note: Automatic key generation is not available when FIPS mode is enabled (fipsMode: true). When both authentication and FIPS mode are enabled, the key must still be generated and set manually.

Strimzi v1 API introduced for custom resources

The v1 API version is now available for all Strimzi Custom Resource Definitions (CRDs). This introduces a stable API for managing Kafka resources.

All example manifests in Cloudera documentation have been updated to use the v1 API version (kafka.strimzi.io/v1).

The v1alpha1, v1beta1, and v1beta2 API versions for Strimzi Custom Resources are deprecated and will be removed in a future release. Convert to the v1 API version (kafka.strimzi.io/v1) using the Strimzi v1 API Conversion Tool. For conversion instructions, see Converting Strimzi custom resources to the v1 API.

Fixed Issues

There are no fixed issues in this release.

Known Issues

Learn about the known issues in this release.

Cloudera Surveyor

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

On Kafka clusters running Kafka Connect or MirrorMaker 2, Cloudera Surveyor incorrectly attempts to describe consumer groups that use the connect protocol, causing snapshot failures. Error messages such as the following might appear in the logs:

```
ERROR [KafkaSnapshotFetcher] Snapshot request describe consumer
groups failed, affected entities: secondary-mm2, connect-cluster:
 java.lang.IllegalArgumentException: GroupId secondary-mm2 is not
 a consumer group (connect).
```

None.

Deprecations

Learn what is deprecated in this release.

Strimzi

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

The type: oauth authentication in Kafka

The type: oauth authentication type is deprecated for both Kafka listeners (brokers) and clients (Kafka Connect) and will be removed in a future release. Use type: custom instead to configure OAuth authentication.

Strimzi API versions v1alpha1, v1beta1, and v1beta2

The v1alpha1, v1beta1, and v1beta2 API versions for Strimzi Custom Resources are deprecated and will be removed in a future release. Convert to the v1 API version (kafka.strimzi.io/v1) using the Strimzi v1 API Conversion Tool. For conversion instructions, see Converting Strimzi custom resources to the v1 API.

Kafka

None.

Cloudera Surveyor

None.

Schema Registry

None.

Removals

Learn what is removed in this release.

Strimzi

None.

Kafka

None.

Cloudera Surveyor

Target ports

The service.targetPort and service.tlsTargetPort properties are removed.

Schema Registry

None.

Behavioral changes

Learn about the behaviroal changes in this release.

Strimzi

None.

Kafka

None.

Cloudera Surveyor

Authentication keys used to generate authentication tokens are now generated automatically

Previous Behavior: The surveyorConfig.surveyor.authentication.keys.active property was not set by default. When configuring authentication, an authentication key of at least 32 bytes had to be generated manually.

New Behavior: The surveyorConfig.surveyor.authentication.keys.active property now defaults to a cryptographically secure random 128-byte value when the property is not explicitly configured and FIPS mode is disabled. When both authentication and FIPS mode are enabled, the key must still be generated and set manually.

Schema Registry

None.

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

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.

Component versions

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

Table 1: Cloudera Streams Messaging Operator for Kubernetes component versions

Component	Version
Cloudera Surveyor for Apache Kafka	0.1.0.1.6.0-b99
Cruise Control	2.5.146.1.6.0-b99
Kafka	4.1.1.1.6.0-b99
Strimzi	0.49.1.1.6.0-b99
Schema Registry	0.10.0.1.6.0-b99

Note: The underlying Schema Registry component version is 7.3.1.600-337 from Cloudera Runtime.

Supported Kafka versions

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

Table 2: Supported Kafka versions

Version	Component/Resource	Metadata version
4.1.1.1.6 (latest and default)	• Kafka – `Kafka` resources • Kafka Connect – `KafkaConnect` resources	4.1-IV1
4.0.1.1.5	• Kafka – `Kafka` resources • Kafka Connect – `KafkaConnect` resources	4.0-IV3

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.

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 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 Operator for Kubernetes. 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 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 1.27 or later cluster.
  - OpenShift 4.14 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 3: 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 11 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