Cloudera Streams Messaging Operator 1.1.0

Release Notes

Date published: 2024-06-11
Date modified: 2024-09-04

https://docs.cloudera.com/

Legal Notice

Contents

Release notes

Learn about the new features, improvements, known and fixed issues, limitations, and unsupported features in this release of CSM Operator.

What's New

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

Rebase on Strimzi 0.41.0

This release of CSM Operator is based on Strimzi 0.41.0. For more information, see the Strimzi 0.41.0 Release Notes.

Kafka Connect support

You can now deploy Kafka Connect clusters and Kafka connectors using `KafkaConnect` and `KafkaConnector` resources. For more information, see Deploying Kafka Connect clusters .

Kafka replication support

You can now set up data replication between Kafka clusters using CSM Operator.

Replication of Kafka data is achieved by manually deploying Kafka Connect clusters and various replication connectors. This type replication setup is referred to as **Kafka Connect-based** replication. Kafka Connect-based replication does not involve the use of MirrorMaker 2 or the `KafkaMirrorMaker2` resource shipped as part of Strimzi.

For more information, see Replication overview .

Fixed Issues

Learn what issues are fixed in this release of CSM Operator.

CSMDS-644: Kafka CLI commands fail with a TLS handshake error when using kafka_shell.sh

The kafka_shell.sh tool no longer fails with a TLS handshake error when an mTLS listener is configured for the Kafka cluster targeted by the tool.

CSMDS-721: Some alerts are triggered when deploying the prometheus-rules.yaml example

False positive results are no longer triggered when deploying the prometheus-rules.yaml example. The pod selector from the KafkaBrokerContainersDown rule is removed. The BridgeContainers Down,

EntityOperatorTlsSidecarContainerDown, and MirrorMakerContainerDown alerts are deleted.

Known Issues

Learn about the known issues in this release of CSM Operator.

CSMDS-933: The log-dir-describe.txt and topic-describe.txt files generated by report.sh are empty

The log-dir-describe.txt and topic-describe.txt files generated by report.sh might be empty for some namespaces or some clusters. These files contain log directory and topic information collected by report.sh.

Run a single instance of report.sh at a time for a single Kubernetes cluster. Do not run multiple instances concurrently.

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

Under certain circumstances, ZooKeeper pods might not be able to form a quorum. In a case like this, the creation of the Kafka cluster gets stuck in a state where ZooKeeper pods are running, but Kafka pods are not created.

If you encounter this issue, at least one of the ZooKeeper pods logs a WARN entry similar to the following:

```
2024-02-23 18:45:00,311 WARN Unexpected exception (org.apache.zo
okeeper.server.quorum.QuorumPeer) [QuorumPeer[myid=3](plain=127.
0.0.1:12181)(secure=[0:0:0:0:0:0:0:0]:2181)]
java.lang.InterruptedException: Timeout while waiting for epoch
 from quorum
 at org.apache.zookeeper.server.quorum.Leader.getEpochToPropose
(Leader.java:1443)
 at org.apache.zookeeper.server.quorum.Leader.lead(Leader.java:60
6)
 at org.apache.zookeeper.server.quorum.QuorumPeer.run(QuorumPeer.
java:1552)
```

This is caused by a race condition issue in ZooKeeper. ZooKeeper is unable to recover from this state automatically.

Delete the ZooKeeper pods that are unable to form a quorum.

```
kubectl delete pod [***ZOOKEEPER POD***] -n [***NAMESPACE***]
```

The Strimzi Cluster Operator automatically recreates the ZooKeeper pods that are deleted. The newly created ZooKeeper pods are less likely to encounter the issue.

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

Under certain circumstances, ZooKeeper pods might not be able to form a quorum during an upgrade. This results in both ZooKeeper and Kafka becoming unavailable for several minutes during an upgrade.

After a certain amount of time, failed ZooKeeper pods are automatically recreated by the Strimzi Cluster Operator, and the upgrade succeeds.

If you encounter this issue, at least one of the ZooKeeper pods will log the following error:

```
java.net.BindException: Cannot assign requested address
```

This issue affects ZooKeeper-based deployments only.

CSMDS-805: The kafka_shell.sh and connect_shell.sh tools do not propagate command return code

The kafka_shell.sh and connect_shell.sh tools do not propagate the return code of the last command which ran inside the shell.

None.

Unsupported features

Learn what features are unsupported in this release of CSM Operator.

The following Strimzi features are unsupported in CSM Operator:

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

Component versions

CSM Operator components and their versions shipped in this release of CSM Operator.

Table 1: CSM Operator component versions

Component	Version
Cruise Control	2.5.137.1.1.0-b79
Kafka	3.7.0.1.1.0-b79
Strimzi	0.41.0.1.1.0-b79
ZooKeeper	3.8.1.7.2.18.200-37

Supported Kafka versions

CSM Operator supports the following Kafka versions:

Table 2: Supported Kafka versions

Version	Component/Resource	Kafka Protocol version
3.7.0.1.1 (latest and default)	• Kafka – `Kafka` resources • Kafka Connect – `KafkaConnect` resources	3.7
3.7.0.1.0	• Kafka – `Kafka` resources	3.7

Kafka versions shipped in CSM 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 CSM 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 CSM Operator. When deploying a cluster, you must use the versions listed here. Specifying upstream versions is not supported.

The Kafka protocol version is relevant for upgrades. Depending on your specific upgrade path, explicitly setting the protocol version might be necessary.

Note: Kafka Connect also uses the Kafka version. However, Kafka Connect is only supported starting with CSM Operator 1.1.

System requirements

CSM 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 CSM Operator or its components.

- A Kubernetes 1.23 or later cluster.
  - OpenShift 4.10 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.
- Log collection is enabled for the Kubernetes cluster.

  Cloudera requires that the logs of 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 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 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 CSM Operator.

Connectors

CSM 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 CSM 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. CSM Operator ships and supports all transformation and predicates plugins included in Apache Kafka. Additionally, CSM Operator ships and supports 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. CSM 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 8 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