

## Overview

Date published: 2024-06-11

Date modified: 2024-06-11



# Legal Notice

© Cloudera Inc. 2024. 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.

This content is modified and adapted from [Strimzi Documentation](#) by Strimzi Authors, which is licensed under [CC BY 4.0](#).

# Contents

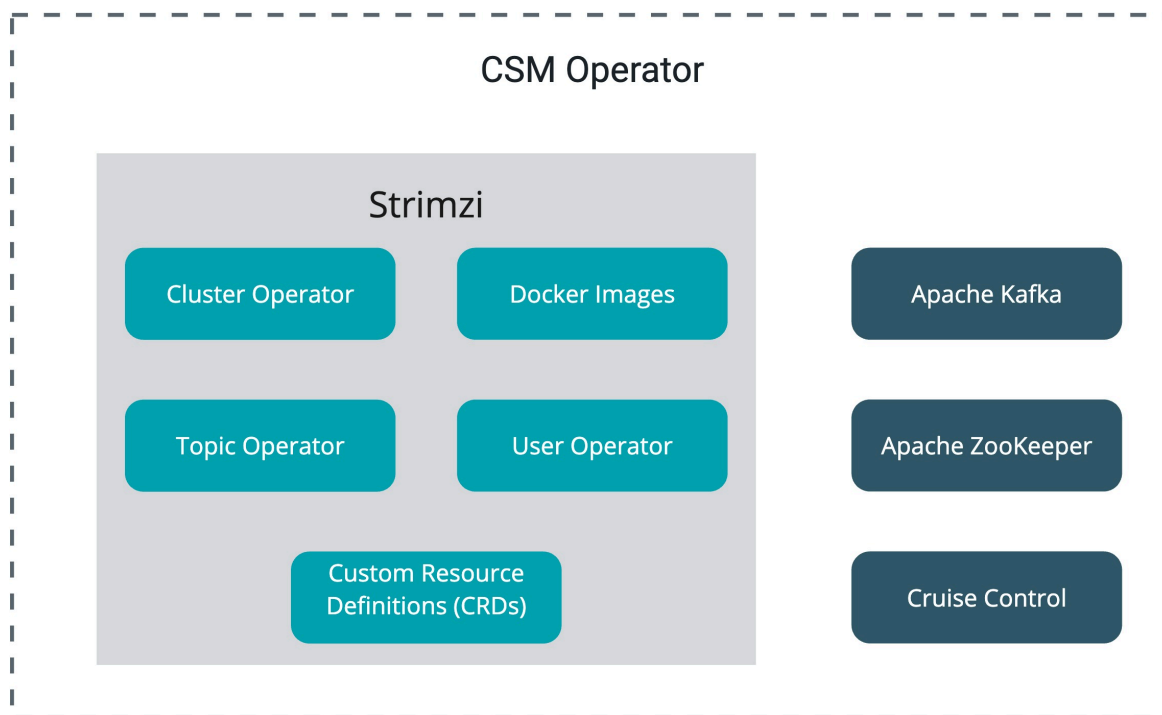
**What is CSM Operator?.....4**

**Deployment architecture.....5**

**Licensing.....7**

## What is CSM Operator?

CSM (Cloudera Streams Messaging) Operator allows you to deploy and manage Streams Messaging components as container applications on Kubernetes. CSM Operator simplifies the process of creating, managing, and troubleshooting Kafka deployments in a Kubernetes environment. CSM Operator includes several services and components such as Strimzi, Kafka, Cruise Control, and others.



### Strimzi

Strimzi is an open-source project that provides a way to run an Apache Kafka cluster on Kubernetes. Within CSM Operator, Strimzi is the component that makes it possible to deploy and manage Kafka workloads in a Kubernetes environment using Kubernetes-native tooling and processes.

Strimzi itself is made up of multiple components and includes various operator applications, Custom Resource Definitions (CRD) as well as Docker (container) images.

Operator applications are purpose built Kubernetes applications that act as an extension to Kubernetes. These applications provide an easy way for you to deploy, manage and configure Kafka and related components.

The CRDs created by Strimzi define the APIs to interface with Kafka-related custom resources on Kubernetes as, for example, `KafkaCluster`, `KafkaNodePools`, and `KafkaTopic`. The custom resources are created as instances of these APIs by providing an associated set of configurations to be applied to the resource. CRDs and custom resources are defined as YAML files.

### Kafka

Apache Kafka is an open-source, high performance, highly available, and redundant streams messaging platform. It supports millions of messages per second with low latency and high throughput, scaling elastically and transparently without downtime. Kafka addresses a wide range

of streaming data initiatives, enabling enterprises to keep up with customer demand, provide better services, and proactively manage risk.

### Apache ZooKeeper

Apache ZooKeeper is an open-source centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services. Kafka uses Zookeeper for broker coordination as well as to store broker, topic, and partition metadata.

### Cruise Control

Cruise Control acts as a load balancing component in large Kafka installations. It provides automatic data balancing of Kafka partitions across Kafka clusters based on user specified parameters (goals) as well as workload data.

Components shipped in CSM Operator are based on open source projects and might contain additional changes or fixes to guarantee that they work in Cloudera supported environments.

Additionally, not all Kafka and Strimzi features are supported by CSM Operator. See the Release Notes for a comprehensive list of unsupported features and components.



**Note:** Cloudera will replace ZooKeeper with KRaft in a future release.

### Key features and benefits

A Kafka deployment with CSM Operator provides the following key features and benefits.

- Flexible, agile, and rapid deployment as well as scaling for variable workloads
- Standardization of deployments on existing Kubernetes infrastructure
- Operational efficiency with simple upgrades, and swift creation of new clusters.
- Ability to deploy Kafka and related components on existing, shared Kubernetes infrastructure. No need for dedicated infrastructure.
- Lightweight dependencies and system requirements for Kafka-centric deployments.

### Related Information

[Strimzi](#)

[Apache Kafka](#)

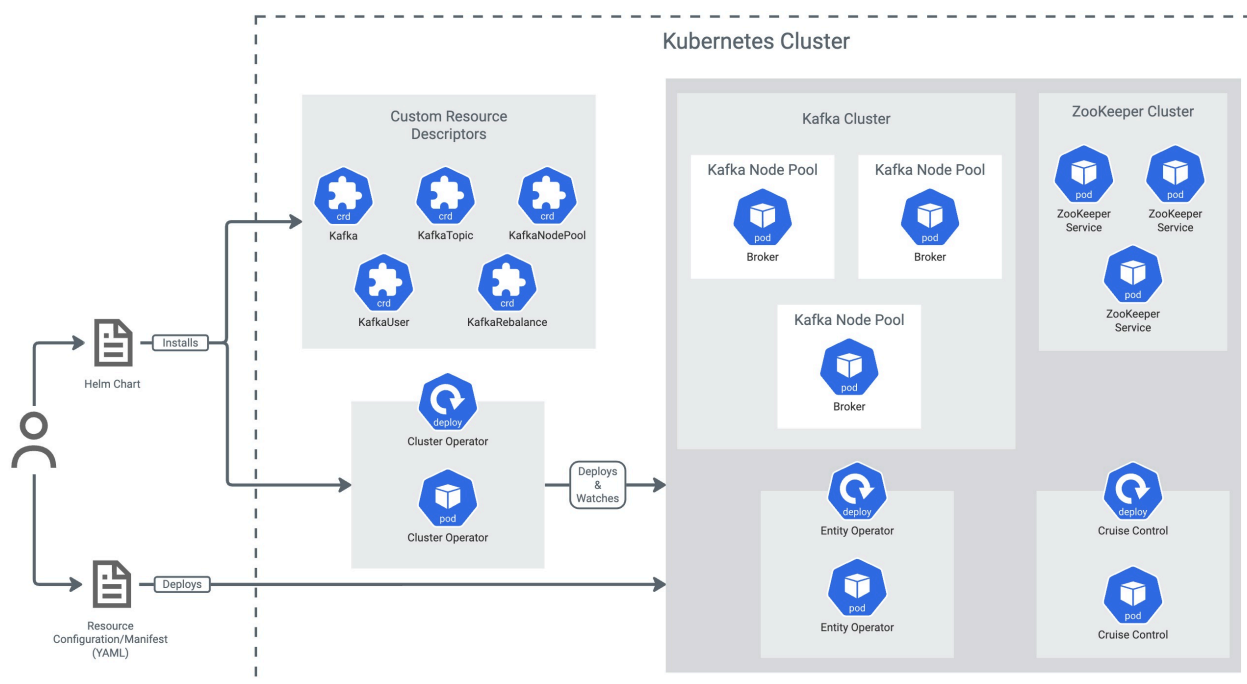
[Apache ZooKeeper](#)

[Cruise Control](#)

[Operator pattern](#) | [Kubernetes](#)

## Deployment architecture

Learn the architecture of a typical CSM Operator deployment.



CSM Operator deployment starts with first installing a Helm chart.

The Helm chart installs various Custom Resource Definitions (CRDs) to the Kuberentes environment. Resources described by these CRDs are managed by the Kubernetes operator applications. Strimzi defines CRDs such as:

- `Kafka` – represents a Kafka cluster consisting of Kafka brokers, ZooKeeper servers, Cruise Control and Strimzi Entity Operators.
- `KafkaNodePool` – represents a group of Kafka brokers from the cluster that have the same configuration.
- `KafkaTopic` – represents a Kafka topic.
- `KafkaUser` – represents a user that is external to the Kafka cluster.
- `KafkaRebalance` – represents a broker rebalance action for the Kafka cluster.

When installing the Helm chart, a Strimzi Cluster Operator Kubernetes deployment is created with a Strimzi Cluster Operator pod running within the deployment. This application is responsible for monitoring the Kafka clusters in the environment and reconciling the Kafka clusters when their configuration changes.

When the `Kafka` and `KafkaNodePool` custom resources are deployed, based on their resource configuration, the Strimzi Cluster Operator will create pods for ZooKeeper and Kafka and will manage the lifecycle of these pods accordingly. The Strimzi Cluster Operator also creates an Entity Operator pod (in an Entity Operator deployment) that is responsible for managing other resources inside the particular Kafka cluster (topics, users, and so on). Optionally a Cruise Control pod is also created if deployed.



**Warning:** Strimzi allows creating Kafka brokers by creating only a single `Kafka` resource. However, CSM Operator only supports creating Kafka brokers by creating `KafkaNodePool` resources. Node pools allow for more flexible deployments with easier scaling options. Moreover, certain features like rack awareness and scaling are limited without node pools. Broker creation using the `Kafka` resource only is deprecated, and results in unnecessary effort of migrating the deployment to use node pools.

## Related Information

[Overview | Strimzi](#)

# Licensing

CSM Operator requires a valid license to function. Licenses are made available to you together with your Cloudera credentials as part of your license and subscription agreement with Cloudera.

Licenses are registered during CSM Operator installation. They are stored in a Kubernetes secret. Licenses can be updated at any time.

Licenses are valid for a set period of time. Once the license expires, the cluster resources you deployed will continue to run. However, reconciliation of resources will be blocked. For example, failed pods will not be restarted, scaling your clusters will not be possible. In general, the control mechanisms in place that keep resources healthy will be blocked. This leads to deployed resources breaking down over time.

CSM Operator publishes various log entries and Kubernetes events related to your licenses.

For example, if your license expires or becomes invalid due to any reason, appropriate logs and events are published notifying you that there are issues with your license.

These logs and events are published for the Strimzi Cluster Operator deployment. You can check these logs and events with the following commands.

```
kubect1 events deployments/strimzi-cluster-operator --namesp  
ace [***NAMESPACE***]
```

```
kubect1 logs deployment/strimzi-cluster-operator --names  
pace [***NAMESPACE***]
```

## Related Information

[Updating a license](#)