

## Installation & Upgrade

Date published: 2024-06-11

Date modified: 2024-09-04



# 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.

# Contents

- Installation..... 4**
  - Installing Strimzi in an internet environment..... 5
  - Installing Strimzi in an air-gapped environment..... 7
- Upgrade..... 10**
  - Upgrading CSM Operator..... 10

## Installation overview

Get started with installing CSM Operator. Learn about the concept of installing CSM Operator, the installation artifacts, and where these artifacts are hosted.

CSM Operator is installed by installing the various components shipped in CSM Operator. In this release of CSM Operator, only a single component, Strimzi, requires installation. Strimzi is installed by installing the Strimzi Cluster Operator with a Helm chart.

### Installation artifacts and artifact locations

CSM Operator ships with various installation artifacts. These artifacts are hosted at two locations, the Cloudera Docker registry and the Cloudera Archive.

Both the Cloudera Docker registry and the Cloudera Archive require Cloudera credentials (username and password) for access. Credentials are provided to you as part of your license and subscription agreement. You can access both the registry and the archive using the same credentials.

#### Cloudera Docker registry – [container.repository.cloudera.com](https://container.repository.cloudera.com)

The Docker registry hosts the Helm chart as well as all Docker images used for installation.

**Table 1: CSM Operator artifacts on the Cloudera Docker registry**

Artifact	Location	Description
Strimzi Docker image	<a href="https://container.repository.cloudera.com/cloudera/kafka-operator:0.41.0.1.1.0-b79">container.repository.cloudera.com/cloudera/kafka-operator:0.41.0.1.1.0-b79</a>	Docker image used for deploying Strimzi and its components.
Kafka Docker image	<a href="https://container.repository.cloudera.com/cloudera/kafka:0.41.0.1.1.0-b79-kafka-3.7.0.1.1">container.repository.cloudera.com/cloudera/kafka:0.41.0.1.1.0-b79-kafka-3.7.0.1.1</a>	Docker image used for deploying Kafka and related components.
Strimzi Cluster Operator Helm chart	<a href="https://container.repository.cloudera.com/cloudera-helm/csm-operator/strimzi-kafka-operator:1.1.0-b79">oci://container.repository.cloudera.com/cloudera-helm/csm-operator/strimzi-kafka-operator:1.1.0-b79</a>	Helm chart used to install the Strimzi Cluster Operator with helm install.

#### Cloudera Archive – [archive.cloudera.com/p/csm-operator/](https://archive.cloudera.com/p/csm-operator/)

The Cloudera Archive hosts various installation artifacts including the Helm chart, configuration examples, a YAML file containing all CRDs, diagnostic tools, and the maven artifacts.

Accessing the Cloudera Archive and the artifacts it hosts is not necessary to complete installation. All artifacts on the archive are supplemental resources. The following table collects the CSM Operator directories located in the archive with an overview of what artifacts they contain and how you can use them.

**Table 2: CSM Operator directories on the Cloudera Archive**

Archive Directory	Description
<a href="https://archive.cloudera.com/p/csm-operator/1.1/charts/">https://archive.cloudera.com/p/csm-operator/1.1/charts/</a>	The charts directory contains the Helm chart. This is the same chart that is available on the Docker registry. Cloudera recommends that whenever possible you install with the chart hosted on the registry. The chart on the archive is provided in case you cannot access the registry or want to download the chart using a browser.
<a href="https://archive.cloudera.com/p/csm-operator/1.1/examples/">https://archive.cloudera.com/p/csm-operator/1.1/examples/</a>	The examples directory includes various examples of resource configuration files. You can use these to quickly deploy Kafka and other components in Kubernetes following installation

Archive Directory	Description
<a href="https://archive.cloudera.com/p/csm-operator/1.1/install/">https://archive.cloudera.com/p/csm-operator/1.1/install/</a>	<p>The install directory contains a single YAML file that collects all Strimzi Cluster Operator CRDs. The purpose of this file is twofold.</p> <p>One, the CRDs are rich in comments. Reviewing them can help you better understand how Kafka is deployed and managed with Strimzi in CSM Operator. It is a supplemental resource to the documentation.</p> <p>Two, this file is used to upgrade CRDs during CSM Operator upgrades.</p> <p>The CRDs are also included in the Strimzi Cluster Operator Helm chart, and Helm will automatically install the necessary CRDs to Kubernetes. You do not need to install them separately with the file hosted on the archive.</p>
<a href="https://archive.cloudera.com/p/csm-operator/1.1/maven-repository/">https://archive.cloudera.com/p/csm-operator/1.1/maven-repository/</a>	The maven artifacts can be used to develop your own applications or tools for use with CSM Operator.
<a href="https://archive.cloudera.com/p/csm-operator/1.1/tools/">https://archive.cloudera.com/p/csm-operator/1.1/tools/</a>	The tools directory contains command line tools that you use to collect diagnostic information and to troubleshoot cluster issues.

## Installing Strimzi in an internet environment

Complete these steps to install Strimzi if your Kubernetes cluster has internet access. Installing Strimzi installs the applications and resources that enable you to deploy and manage Kafka in Kubernetes.

### About this task

Strimzi is installed in your Kubernetes cluster with the Strimzi Cluster Operator Helm chart using the `helm install` command. When you install the chart, Helm installs the Strimzi Custom Resource Definitions (CRDs) included in CSM Operator and deploys the Strimzi Cluster Operator, which is an operator application that manages and monitors Kafka and related components. Additionally, other cluster resources and applications required for managing Kafka are also installed.

Installing Strimzi does not create or deploy a Kafka cluster. Kafka clusters are created following the installation by deploying `Kafka` and `KafkaNodePool` resources in the Kubernetes cluster with `kubectl` or `oc`.

Cloudera recommends that you install Strimzi once per Kubernetes cluster. Some resources are cluster-wide, which can cause issues if Strimzi is installed multiple times on the same cluster.

By default the Strimzi Cluster Operator (deployed with installation) watches and manages the Kafka clusters that are deployed in the same namespace as the Strimzi Cluster Operator. However, you can configure it to watch any namespace. This allows you to manage multiple Kafka clusters deployed in different namespaces using a single installation.

### Before you begin

- Ensure that your Kubernetes environment meets requirements listed in [System requirements](#).
- Your Kubernetes cluster requires internet connectivity to complete these steps. It must be able to reach the Cloudera Docker registry.
- Ensure that you have access to your Cloudera credentials (username and password). Credentials are required to access the Cloudera Archive and Cloudera Docker registry where installation artifacts are hosted.
- Ensure that you have access to a valid Cloudera license.
- Review the [Helm chart reference](#) before installation.

The Helm chart accepts various configuration properties that you can set during installation. Using these properties you can customize your installation.

- If you are planning to watch and manage more than 20 Kafka clusters with a single installation, you must increase the memory and heap allocated to the Strimzi Cluster Operator. You can specify memory configuration in your helm install command. For more information, see [Increasing Cluster Operator memory](#).

## Procedure

1. Create a namespace in your Kubernetes cluster.

```
kubectl create namespace [***NAMESPACE***]
```

This is the namespace where you install Strimzi. Use the namespace you create in all installation steps that follow.

2. Create a Kubernetes secret containing your Cloudera credentials.

```
kubectl create secret docker-registry [***SECRET NAME***] \
  --docker-server container.repository.cloudera.com \
  --docker-username [***USERNAME***] \
  --docker-password [***PASSWORD***] \
  --namespace [***NAMESPACE***]
```

Replace [\*\*\*USERNAME\*\*\*] and [\*\*\*PASSWORD\*\*\*] with your Cloudera credentials.



**Important:** The Secret containing your Cloudera credentials must exist in the namespace where you install Strimzi as well as all namespaces where you deploy Kafka or Kafka Connect clusters. Cloudera recommends that you create the Secret in all required namespaces now if you know what namespaces you will be using to deploy Kafka or Kafka Connect.

3. Log in to the Cloudera Docker registry with helm.

```
helm registry login container.repository.cloudera.com
```

Enter your Cloudera credentials when prompted.

4. Install Strimzi with helm install.

```
helm install strimzi-cluster-operator \
  --namespace [***NAMESPACE***] \
  --set 'image.imagePullSecrets[0].name=[***SECRET NAME***]' \
  --set-file clouderaLicense.fileContent=[***PATH TO LICENSE FILE***] \
  --set watchAnyNamespace=true \
  oci://container.repository.cloudera.com/cloudera-helm/csm-operator/strimzi-kafka-operator \
  --version 1.1.0-b79
```

- The string strimzi-cluster-operator is the Helm release name of the chart installation. This is an arbitrary, user defined name. Cloudera recommends that you use a unique and easily identifiable name.
- imagePullSecrets specifies what secret is used to pull images from the Cloudera registry. Setting this property is mandatory, otherwise, Helm cannot pull the necessary images from the Cloudera Docker registry. Ensure that you replace [\*\*\*SECRET NAME\*\*\*] with the name of the secret you created in [Step 2](#).
- clouderaLicense.fileContent is used to register your license. If this property is set, a secret is generated that contains the license you specify. Setting this property is mandatory. The Strimzi Cluster Operator will not function without a valid license. Ensure that you replace [\*\*\*PATH TO LICENSE FILE\*\*\*] with the full path to your Cloudera license file.
- You can use --set to set various other properties of the Helm chart. This enables you to customize your installation. For example, Cloudera recommends that you configure the Cluster Operator to watch all namespaces, this is configured by setting watchAnyNamespace to true. Alternatively, you can configure a list of specific namespaces to watch using watchNamespaces.

## 5. Verify your installation

This is done by listing the deployments and pods in your namespace. If installation is successful, you should see a `strimzi-cluster-operator` deployment and pod present in the cluster.

```
kubectl get deployments --namespace [***NAMESPACE***]
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
#...				
strimzi-cluster-operator	1/1	1	1	13m

```
kubectl get pods --namespace [***NAMESPACE***]
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
#...				
strimzi-cluster-operator	1/1	1	1	13m

## 6. Access supplemental resources available on the Cloudera Archive.

Supplemental resources available on the Cloudera Archive include various example files, diagnostic tools, and more. You can use these resources to quickly deploy Kafka clusters and to gain a better understanding of Strimzi and CSM Operator.

### What to do next

- Deploy a Kafka cluster, see [Deploying Kafka](#).
- Set up Prometheus for monitoring, see [Configuring Kafka for Prometheus monitoring](#) and [Monitoring with Prometheus](#).

## Installing Strimzi in an air-gapped environment

Complete these steps to install Strimzi if your Kubernetes cluster does not have internet access, or if you want to install from a self-hosted registry. Installing Strimzi installs the applications and resources that enable you to deploy and manage Kafka in Kubernetes.

### About this task

Strimzi is installed in your Kubernetes cluster with the Strimzi Cluster Operator Helm chart using the `helm install` command. When you install the chart, Helm installs the Strimzi Custom Resource Definitions (CRDs) included in CSM Operator and deploys the Strimzi Cluster Operator, which is an operator application that manages and monitors Kafka and related components. Additionally, other cluster resources and applications required for managing Kafka are also installed.

Installing Strimzi does not create or deploy a Kafka cluster. Kafka clusters are created following the installation by deploying `Kafka` and `KafkaNodePool` resources in the Kubernetes cluster with `kubectl` or `oc`.

Cloudera recommends that you install Strimzi once per Kubernetes cluster. Some resources are cluster-wide, which can cause issues if Strimzi is installed multiple times on the same cluster.

By default the Strimzi Cluster Operator (deployed with installation) watches and manages the Kafka clusters that are deployed in the same namespace as the Strimzi Cluster Operator. However, you can configure it to watch any namespace. This allows you to manage multiple Kafka clusters deployed in different namespaces using a single installation.

### Before you begin

- Ensure that your Kubernetes environment meets requirements listed in [System requirements](#).
- A self-hosted Docker registry is required. Your registry must be accessible by your Kubernetes cluster.

- A machine with Internet connectivity is required. While the Kubernetes cluster does not need internet access, you will need a machine to pull the images from the Cloudera Docker registry.
- Access to docker or equivalent utility that you can use to pull and push images is required. The following steps use docker. Replace commands where necessary.
- Ensure that you have access to your Cloudera credentials (username and password). Credentials are required to access the Cloudera Archive and Cloudera Docker registry where installation artifacts are hosted.
- Ensure that you have access to a valid Cloudera license.
- Review the [Helm chart reference](#) before installation.

The Helm chart accepts various configuration properties that you can set during installation. Using these properties you can customize your installation.

- If you are planning to watch and manage more than 20 Kafka clusters with a single installation, you must increase the memory and heap allocated to the Strimzi Cluster Operator. You can specify memory configuration in your helm install command. For more information, see [Increasing Cluster Operator memory](#).

## Procedure

1. Copy the following installation artifacts to your self-hosted registry.

**Table 3: CSM Operator artifacts on the Cloudera Docker registry**

Artifact	Location	Description
Strimzi Docker image	container.repository.cloudera.com/cloudera/kafka-operator:0.41.0.1.1.0-b79	Docker image used for deploying Strimzi and its components.
Kafka Docker image	container.repository.cloudera.com/cloudera/kafka:0.41.0.1.1.0-b79-kafka-3.7.0.1.1	Docker image used for deploying Kafka and related components.
Strimzi Cluster Operator Helm chart	oci://container.repository.cloudera.com/cloudera-helm/csm-operator/strimzi-kafka-operator:1.1.0-b79	Helm chart used to install the Strimzi Cluster Operator with helm install.

This step involves pulling the artifacts from the Cloudera Docker registry, retagging them, and then pushing them to your self-hosted registry. The exact steps you need to carry it out depend on your environment and how your registry is set up. The following substeps demonstrate the basic workflow using docker and helm.

- a) Log in to the Cloudera Docker registry with both docker and helm.

Provide your Cloudera credentials when prompted.

```
docker login container.repository.cloudera.com
```

```
helm registry login container.repository.cloudera.com
```

- b) Pull the Docker images from the Cloudera Docker registry.

```
docker pull \
  container.repository.cloudera.com/cloudera/[***IMAGE
  NAME***]:[***VERSION***]
```

- c) Pull the Strimzi Cluster Operator Helm chart.

```
helm pull \
  oci://container.repository.cloudera.com/cloudera-helm/csm-operator/str
  imzi-kafka-operator \
  --version 1.1.0-b79
```

- d) Retag the Docker images you pulled so that they contain the address of your registry.

```
docker tag \
  [***ORIGINAL IMAGE TAG***] \
```



```
[**YOUR_REGISTRY**]/cloudera/[**IMAGE_NAME**]:[**VERSION**]
```

- e) Push the images and chart to your self-hosted registry.

```
docker push \
  [**YOUR_REGISTRY**]/cloudera/[**IMAGE_NAME**]:[**VERSION**]

helm push \
  strimzi-kafka-operator-1.1.0-b79.tgz \
  oci://[**REGISTRY_HOSTNAME**]:[**PORT**]/cloudera-helm/csm-opera
tor/
```

2. Create a namespace in your Kubernetes cluster.

```
kubectl create namespace [**NAMESPACE**]
```

This is the namespace where you install Strimzi. Use the namespace you create in all installation steps that follow.

3. Create a Kubernetes secret containing credentials for your self-hosted registry.

```
kubectl create secret docker-registry [**SECRET_NAME**] \
  --docker-server [**REGISTRY_HOSTNAME**]:[**PORT**] \
  --docker-username [**USERNAME**] \
  --docker-password [**PASSWORD**] \
  --namespace [**NAMESPACE**]
```

Replace [\*\*USERNAME\*\*] and [\*\*PASSWORD\*\*] with your credentials.



**Important:** The Secret containing your Cloudera credentials must exist in the namespace where you install Strimzi as well as all namespaces where you deploy Kafka or Kafka Connect clusters. Cloudera recommends that you create the Secret in all required namespaces now if you know what namespaces you will be using to deploy Kafka or Kafka Connect.

4. Log in to your self-hosted registry with helm.

```
helm registry login [**REGISTRY_HOSTNAME**]:[**PORT**]
```

Enter your credentials when prompted.

5. Install Strimzi with helm install.

```
helm install strimzi-cluster-operator \
  --namespace [**NAMESPACE**] \
  --set 'image.imagePullSecrets[0].name=[**SECRET_NAME**]' \
  --set defaultImageRegistry=[**REGISTRY_HOSTNAME**]:[**PORT**] \
  --set-file clouderaLicense.fileContent=[**PATH TO LICENSE FILE**] \
  oci://[**YOUR_REGISTRY**]/cloudera-helm/csm-operator/strimzi-kafka-op
erator \
  --version 1.1.0-b79 \
  --set watchAnyNamespace=true
```

- The string strimzi-cluster-operator is the Helm release name of the chart installation. This is an arbitrary, user defined name. Cloudera recommends that you use a unique and easily identifiable name.
- imagePullSecrets specifies what secret is used to pull images from the specified registry. Ensure that you replace [\*\*SECRET\_NAME\*\*] with the name of the secret you created in [Step 3](#).
- clouderaLicense.fileContent is used to register your license. If this property is set, a secret is generated that contains the license you specify. Setting this property is mandatory. The Strimzi Cluster Operator will not function without a valid license. Ensure that you replace [\*\*PATH TO LICENSE FILE\*\*] with the full path to your Cloudera license file.
- You can use --set to set various other properties of the Helm chart. This enables you to customize your installation. For example, Cloudera recommends that you configure the Cluster Operator to watch all

namespaces, this is configured by setting `watchAnyNamespace` to `true`. Alternatively, you can configure a list of specific namespaces to watch using `watchNamespaces`.

## 6. Verify your installation

This is done by listing the deployments and pods in your namespace. If installation is successful, you should see a `strimzi-cluster-operator` deployment and pod present in the cluster.

```
kubectl get deployments --namespace [***NAMESPACE***]
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
#...				
strimzi-cluster-operator	1/1	1	1	13m

```
kubectl get pods --namespace [***NAMESPACE***]
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
#...				
strimzi-cluster-operator	1/1	1	1	13m

## 7. Access supplemental resources available on the Cloudera Archive.

Supplemental resources available on the Cloudera Archive include various example files, diagnostic tools, and more. You can use these resources to quickly deploy Kafka clusters and to gain a better understanding of Strimzi and CSM Operator.

### What to do next

- Deploy a Kafka cluster, see [Deploying Kafka](#).
- Set up Prometheus for monitoring, see [Configuring Kafka for Prometheus monitoring](#) and [Monitoring with Prometheus](#).

# Upgrade Overview

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc eu ex ac odio lobortis scelerisque. Suspendisse et pretium lorem. Curabitur tristique, magna in tempus blandit, augue tellus convallis sem, eget viverra magna nisi eget felis. Duis ornare lorem egestas lobortis posuere. Nulla pharetra finibus consequat. Ut elit nulla, sagittis eget nibh sit amet, euismod viverra elit. Curabitur venenatis, erat ut sodales tincidunt, sapien leo vulputate felis, et pharetra nunc ipsum et nisl. Donec maximus velit in ipsum pretium accumsan. Praesent ut pulvinar eros.

Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. In hac habitasse platea dictumst. Sed massa leo, gravida laoreet turpis et, finibus molestie libero. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Praesent convallis, orci quis gravida volutpat, ligula tortor vulputate nisl, ac eleifend ante justo vitae sapien. Donec molestie dolor ac felis tincidunt dapibus. In aliquam sem arcu. Sed dui nunc, tincidunt eget nisl id, blandit sollicitudin metus. Cras auctor urna eu sapien fermentum, id sagittis ipsum cursus. Cras eu sem sit amet nibh luctus faucibus. Mauris maximus neque vitae justo convallis molestie. Nam in mi non dui fermentum pellentesque ut non tortor.

## Upgrading CSM Operator

Complete these steps to upgrade CSM Operator. Upgrading CSM Operator involves upgrading both Strimzi and Kafka in your cluster.

## About this task

Upgrading Strimzi consists of upgrading the Strimzi CRDs to the new version and upgrading the cluster operator using Helm commands. If you are upgrading from a maintenance version, you also need to temporarily set the image of the Kafka cluster to the maintenance version in your Kafka resources. This is necessary because without the explicit image, the Strimzi upgrade will automatically update the image used by the Kafka cluster, which might not contain all bug fixes provided in the maintenance version.

Upgrading the cluster operator may affect the Kafka resources in the cluster. All Kafka clusters that specify the version of the cluster but not the image will be restarted during the cluster operator upgrade. This is due to the fact that the default image of all versions changes with the Strimzi upgrade, triggering a restart. This restart is safe, that is, all healthy topics with a proper replication factor and minimum ISR are kept online during the restart.

Upgrading Kafka involves updating your Kafka resources for each cluster to the latest supported version after a Strimzi upgrade. Upgrading Kafka is:

- Strongly recommended by Cloudera after every Strimzi upgrade.
- Mandatory if you are upgrading from a maintenance version.

This procedure upgrades your Kafka cluster in a rolling upgrade. The upgrade is safe, all healthy topics with a proper replication factor and minimum ISR are kept online during the upgrade.

## Before you begin

- Ensure that your Kubernetes environment meets requirements listed in [System requirements](#).
- Ensure that you have access to your Cloudera credentials (username and password). Credentials are required to access the Cloudera Archive and Cloudera Docker registry where upgrade artifacts are hosted.
- If you are upgrading from a maintenance version, check that the bug fixes provided in the maintenance version are available in the newer Kafka supported by CSM Operator. If certain fixes are not available, be aware that upgrading will result in regressed functionality.
- The following steps instruct you to set `inter.broker.protocol.version` during the upgrade to keep Kafka in backward compatible state during the upgrade. This is only necessary if the protocol version differs between your current and new versions.

If there is no change in the protocol version, you also do not finalize the upgrade. This means that for these types of upgrades, a rollback is possible even after you completed the upgrade.

You can find the Kafka protocol version in [Component versions](#).

## Procedure

### 1. Update the Strimzi CRDs.

You do this by replacing the currently installed CRDs with the new version. The CRDs are published as a single YAML on the Cloudera Archive in `/p/csm-operator/1.1/install`.

```
curl -s --user [***USERNAME***] \
  https://archive.cloudera.com/p/csm-operator/1.1.0/install/strimzi-c
rds-0.41.0.1.1.0-b79.yaml \
| kubectl replace --filename -
```

Enter your Cloudera password when prompted.

### 2. If you are upgrading from a maintenance version, explicitly set the image of the Kafka cluster to the maintenance version in your Kafka resources.

These steps ensure that the Kafka clusters keep the exact maintenance image during the Strimzi upgrade. This is a temporary setup, and you need to change it after the Strimzi upgrade. Repeat these steps for each Kafka cluster.

a) Extract the currently used Zookeeper image.

```
kubectl get pod --namespace [***KAFKA CLUSTER NAMESPACE***] \
```

```
--selector strimzi.io/component-type=zookeeper,strimzi.io/clust
er=[***KAFKA CLUSTER NAME***] \
--output=jsonpath="{@[ 'items' ][0].spec.containers[0].image}{'\n'}"
```

b) Extract the currently used Kafka image.

```
kubectl get pod --namespace [***KAFKA CLUSTER NAMESPACE***] \
--selector strimzi.io/component-type=kafka,strimzi.io/clust
er=[***KAFKA CLUSTER NAME***] \
--output=jsonpath="{@[ 'items' ][0].spec.containers[0].image}{'\n'}"
```

c) Update the spec.zookeeper.image and the spec.kafka.image properties in the Kafka resource.

```
kubectl edit kafka [***KAFKA CLUSTER NAME***] \
--namespace [***KAFKA CLUSTER NAMESPACE***]
```

3. Log in to the Cloudera Docker registry with helm.

```
helm registry login container.repository.cloudera.com
```

Enter your Cloudera credentials when prompted.

4. Upgrade Strimzi using Helm.

This step upgrades the Strimzi Cluster Operator. Under the hood, the Strimzi Cluster Operator deployment is updated by changing the image used by the Strimzi Cluster Operator.

```
helm upgrade strimzi-cluster-operator \
--namespace [***STRIMZI CLUSTER OPERATOR NAMESPACE***] \
--atomic \
oci://container.repository.cloudera.com/cloudera-helm/csm-operator/str
imzi-kafka-operator \
--version 1.1.0-b79
```

- The string strimzi-cluster-operator is the Helm release name of the chart installation. This is an arbitrary, user-defined name. Replace this string if you used a different name when you installed Strimzi.
- The --atomic option makes the command wait for the upgrade to complete or roll back if the upgrade fails.



**Note:** Usually, you do not need to set new Helm chart properties during upgrade, the properties that you configured during installation are preserved. However, If you set even a single property using the --set options in your helm upgrade command, properties set previously might be ignored. If you decide to set properties during the upgrade, ensure that you do one of the following.

- Set all your properties (including the ones that were set during installation) during upgrade. This ensures that all required properties are set explicitly.
- Set the properties you want to update and use the --reset-then-reuse-values option. Using this option preserves previously set properties. This option is only available in the more recent versions of Helm.

5. Double check that the Strimzi Cluster operator is in a healthy state.

```
kubectl get deployments strimzi-cluster-operator \
--namespace [***STRIMZI CLUSTER OPERATOR NAMESPACE***]
```

6. If the upgraded pods crash, check their logs.

7. If needed, roll back this procedure with helm rollback.

```
helm rollback strimzi-cluster-operator \
--namespace [***STRIMZI CLUSTER OPERATOR NAMESPACE***]
```

8. Upgrade Kafka.

This is done by updating your Kafka resources for each cluster.

```
kubectl edit kafka [***KAFKA CLUSTER NAME***] \
```

```
--namespace [***KAFKA NAMESPACE***]
```

- a) Set the `.spec.kafka.version` to the latest version.
- b) For maintenance versions: Remove the `spec.zookeeper.image` and `spec.kafka.image` properties.
- c) Add the old `inter.broker.protocol.version` in `spec.kafka.config`, for example:

```
#...
kind: Kafka
spec:
  kafka:
    version: 3.7.0.1.1
    config:
      inter.broker.protocol.version: "3.7"
```

Specifying `inter.broker.protocol.version` keeps Kafka in a backward-compatible state following the upgrade.

- d) Save the changes made to the Kafka resource.

**9.** Wait for the cluster pods to become ready.

```
kubectl get pod --namespace [***KAFKA NAMESPACE***] \
--selector strimzi.io/cluster=[***KAFKA CLUSTER NAME***] \
--watch
```

**10.** If the upgraded pods crash, check their logs.

**11.** If needed, roll back this procedure by editing the Kafka resource.

- a) Set the `spec.kafka.version` to the previous version.
- b) If you are upgrading from a maintenance version, set the `spec.kafka.image` property to the previous image.
- c) Remove the `inter.broker.protocol.version` property from `spec.kafka.config`.
- d) Save the changes made to the Kafka resource.

**12.** Verify that the Kafka upgrade is successful and everything is working fine.

**13.** Finalize the Kafka upgrade by removing the old `inter.broker.protocol.version`.

After this step, rollback is no longer possible.