Cloudera Streams Messaging - Kubernetes Operator 1.3.0

Installation

Date published: 2024-06-11
Date modified: 2025-02-28

https://docs.cloudera.com/

Legal Notice

Contents

Installation overview

Get started with installing Cloudera Streams Messaging - Kubernetes Operator. Learn about available installation methods, the installation artifacts, and where these artifacts are hosted.

You can install Cloudera Streams Messaging - Kubernetes Operator using Helm. Alternatively, if you are on OpenShift, you can choose to install from OperatorHub as well.

Both installation methods install the Strimzi Custom Resource Definitions (CRDs) included in Cloudera Streams Messaging - Kubernetes Operator and deploy 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.

Installation using Helm involves installing Strimzi using the Strimzi Cluster Operator Helm chart. For Helm-based installation see Installing Strimzi in an internet environment on page 5 or Installing Strimzi in an air-gapped environment on page 7.

Installation from OperatorHub in OpenShift involves the creation of two secrets in your installation namespace. Afterward, you install Cloudera Streams Messaging - Kubernetes Operator either using the OpenShift web console or CLI. Installation from OperatorHub is done using Operator Lifecycle Manager (OLM). For OperatorHub-based installation, see Installing from OperatorHub in OpenShift on page 11.

Installation artifacts and artifact locations

Cloudera Streams Messaging - Kubernetes 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

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

Table 1: Cloudera Streams Messaging - Kubernetes Operator artifacts on the Cloudera Docker registry

Artifact	Location	Description
Strimzi Docker image	container.repository.cloudera.com/cloudera/kafka-operator:0.45.0.1.3.0-b52	Docker image used for deploying Strimzi and its components.
Kafka Docker image	container.repository.cloudera.com/cloudera/kafka:0.45.0.1.3.0-b52-kafka-3.9.0.1.3	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:1.3.0-b52	Helm chart used to install the Strimzi Cluster Operator with helm install.

Cloudera Archive – 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 Cloudera Streams Messaging - Kubernetes Operator directories located in the archive with an overview of what artifacts they contain and how you can use them.

Table 2: Cloudera Streams Messaging - Kubernetes Operator directories on the Cloudera Archive

Archive Directory	Description
https://archive.cloudera.com/p/csm-operator/1.3/charts/	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.
https://archive.cloudera.com/p/csm-operator/1.3/examples/	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
https://archive.cloudera.com/p/csm-operator/1.3/install/	The install directory contains a single YAML file that collects all Strimzi Cluster Operator CRDs. The purpose of this file is twofold. One, the CRDs are rich in comments. Reviewing them can help you better understand how Kafka is deployed and managed with Strimzi in Cloudera Streams Messaging - Kubernetes Operator. It is a supplemental resource to the documentation. Two, this file is used to upgrade CRDs during upgrades. 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.
https://archive.cloudera.com/p/csm-operator/1.3/maven-repository/	The maven artifacts can be used to develop your own applications or tools for use with Cloudera Streams Messaging - Kubernetes Operator.
https://archive.cloudera.com/p/csm-operator/1.3/tools/	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 Cloudera Streams Messaging - Kubernetes 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/str
   imzi-kafka-operator \
     --version 1.3.0-b52
   ```

   - 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 Cloudera Streams Messaging - Kubernetes 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.

Related Information

Cloudera Archive

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 Cloudera Streams Messaging - Kubernetes 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: Cloudera Streams Messaging - Kubernetes Operator artifacts on the Cloudera Docker registry

Artifact	Location	Description
Strimzi Docker image	container.repository.cloudera.com/cloudera/kafka-operator:0.45.0.1.3.0-b52	Docker image used for deploying Strimzi and its components.
Kafka Docker image	container.repository.cloudera.com/cloudera/kafka:0.45.0.1.3.0-b52-kafka-3.9.0.1.3	Docker image used for deploying Kafka and related components.

Artifact	Location	Description
Strimzi Cluster Operator Helm chart	oci://container.repository.cloudera.com/cloudera-helm/csm-operator/strimzi-kafka-operator:1.3.0-b52	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 1.3.0-b52
```

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-1.3.0-b52.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 1.3.0-b52 \
  --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 Cloudera Streams Messaging - Kubernetes 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.

Related Information

Cloudera Archive

Installing from OperatorHub in OpenShift

Learn how to install Cloudera Streams Messaging - Kubernetes Operator from OperatorHub in OpenShift.

About this task

Important: When installing from OperatorHub, Cloudera Streams Messaging - Kubernetes Operator is installed using Operator Lifecycle Manager (OLM). Customizing your installation and setting the properties of the Strimzi Cluster Operator is limited both during and following installation.

Installation from OperatorHub in Openshift involves creating two `Secrets` in your installation namespace. One containing your Cloudera license, and one containing your Cloudera credentials (username and password). The license is required for Cloudera Streams Messaging - Kubernetes Operator to function properly. The credentials provide access to the Cloudera Docker registry (container.repository.cloudera.com) where installation artifacts are pulled from.

After the `Secrets` are available in your cluster, you can continue with the standard process of installing operators from OperatorHub.

Before you begin

- Ensure that you have access to your Cloudera credentials (username and password).
- Ensure that you have access to a valid Cloudera license.
- These instructions use oc to create `Secrets`. However, you can also create both `Secrets` using the OpenShift web console.

Procedure

1. Create a `Secret` containing your license.

   The name of the `Secret` is fixed. It must be called csm-op-license.

   ```
   oc create secret generic csm-op-license --from-file=license.txt=[***PATH
    TO LICENSE FILE***]
   ```

2. Create a `Secret` containing your Cloudera credentials.

   The name of the `Secret` is fixed. It must be called cloudera-container-repository-credentials.

   ```
   oc create secret docker-registry cloudera-container-repository-credentials
    \
     --docker-username=[***USERNAME***] \
     --docker-password=[***PASSWORD***] \
   ```

```
  --docker-server=container.repository.cloudera.com
```

Note: The `Secret` containing your Cloudera credentials must also be available in 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. Install Cloudera Streams Messaging - Kubernetes Operator from OperatorHub using the web console or the CLI.

   For detailed steps, see Adding Operators to a cluster in the OpenShift documentation.

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.