CSA Operator Installation

Date published:
Date modified:



https://docs.cloudera.com/

Legal Notice

© Cloudera Inc. 2026. 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 overview....4

Installing Cloudera Streaming Analytics Operator for Kubernetes in an environment with internet access....5

Install Cloudera Streaming Analytics Operator for Kubernetes in Taikun CloudWorks [Technical Preview]....8

Importing the Cloudera Streaming Analytics Operator for Kubernetes Helm chart into Taikun CloudWorks....8

Installing Cloudera Streaming Analytics Operator for Kubernetes in Taikun CloudWorks....9

Installing Cloudera Streaming Analytics Operator for Kubernetes in an air-gapped environment....11

Configuring FIPS mode....15

Installation overview

Get started with installing Cloudera Streaming Analytics Operator for Kubernetes. Learn about the concept of installing, the installation artifacts, and where these artifacts are hosted.

Cloudera Streaming Analytics Operator for Kubernetes is installed using a Helm chart, which installs the Apache Kubernetes Flink Operator (Flink Operator), Cloudera SQL Stream Builder, and other components.

Installation artifacts and artifact locations

Note:

Cloudera Streaming Analytics Operator for Kubernetes operator offers alternative images that include the Hadoop, Hive, Iceberg, and Kudu connectors, as well as all required dependencies, in the Flink image.

To use the images with the connectors included, replace flink:1.20.5-csaop1.6.1-b6 with the name of the alternative image in the installation commands.

The default database image for Cloudera SQL Stream Builder uses PostgreSQL 18.4.

Cloudera Streaming Analytics Operator for Kubernetes ships with various installation artifacts, hosted at two locations: the Cloudera Docker registry and the Cloudera Archive.

Both the Cloudera Docker registry and the Cloudera Archive require your Cloudera credentials (username and password) to 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 Cloudera Docker registry hosts the Helm chart as well as all Docker images used for the installation.

Table 1: Cloudera Streaming Analytics Operator for Kubernetes artifacts on the Cloudera Docker registry

Artifact	Location (Base image)	Location (Image with connectors)	Description
Flink Kubernetes Operator Docker image	container.repository.cloudera.com/cloudera/flink-kubernetes-operator:1.13-csaop1.6.1-b6		Docker image used for deploying the various operator components shipped with the Cloudera Streaming Analytics Operator for Kubernetes.
Flink Docker image	container.repository.cloudera.com/cloudera/flink:1.20.5-csaop1.6.1-b6		Docker image used for deploying Apache Flink and its related components.
SQL Runner Docker image	container.repository.cloudera.com/cloudera/flink-extended:1.20.5-csaop1.6.1-b6	container.repository.cloudera.com/cloudera/flink-extended-hadoop:1.20.5-csaop1.6.1-b6	Docker image used for deploying Flink application when SQL query is executed in SSB.
SQL Stream Engine Docker image	container.repository.cloudera.com/cloudera/ssb-sse:1.20.5-csaop1.6.1-b6	container.repository.cloudera.com/cloudera/ssb-sse-hadoop:1.20.5-csaop1.6.1-b6	Docker image used for deploying Cloudera SQL Stream Builder and its UI.

Note: The images listed in the column Image with connectors contain the following connectors: Hadoop, Hive, Iceberg, and Kudu.

Note: The images are built for linux/arm64 and linux/amd64 architectures.

Cloudera Archive – archive.cloudera.com/p/csa-operator/1.6.1/

The Cloudera Archive hosts various installation artifacts including the Helm chart, diagnostic tools, and the Maven artifacts.

All artifacts hosted in the Cloudera Archive are supplemental resources, accessing them is not required to complete the installation. The following table collects the Cloudera Streaming Analytics Operator for Kubernetes directories located in the Cloudera Archive, with an overview of what artifacts they contain and how you can use them:

Table 2: CSA Operator artifacts on the Cloudera Archive

Archive Directory	Description
archive.cloudera.com/p/csa-operator/1.6.1/**charts**/	The charts directory contains the Helm chart. This is the same chart that is available in the Cloudera Docker registry. Cloudera recommends that whenever possible you install with the chart hosted in the registry. The chart in the archive is provided in case you cannot access the registry or want to download the chart using a browser.
archive.cloudera.com/p/csa-operator/1.6.1/**maven-repository**/	The Maven artifacts can be used to develop your own applications or tools for use with Cloudera Streaming Analytics Operator for Kubernetes.
archive.cloudera.com/p/csa-operator/1.6.1/**tools**/	The tools directory contains command line tools that you use to collect diagnostic information and to troubleshoot cluster issues.

Installing Cloudera Streaming Analytics Operator for Kubernetes in an environment with internet access

Complete these steps to install Cloudera Streaming Analytics Operator for Kubernetes if your Kubernetes cluster has internet access. Installing the Cloudera Streaming Analytics Operator for Kubernetes enables you to deploy and manage Flink and Cloudera SQL Stream Builder in Kubernetes.

About this task

Cloudera Streaming Analytics Operator for Kubernetes is installed in your Kubernetes cluster with the provided Helm chart through the helm install command. When you install the chart, Helm installs the Custom Resource Descriptors (CRD) included in Cloudera Streaming Analytics Operator for Kubernetes, and deploys the Apache Kubernetes Flink Operator (Flink Operator), Cloudera SQL Stream Builder engine, and a PostgreSQL (18.4 by default) database for Cloudera SQL Stream Builder.

Installing Cloudera Streaming Analytics Operator for Kubernetes does not create or deploy a Flink cluster. The Flink cluster is created after the installation by deploying the Flink Deployment resource in the Kubernetes cluster with kubectl or oc, or when you execute a SQL job in Streaming SQL Console.

Note: Cloudera recommends that you install Cloudera Streaming Analytics Operator for Kubernetes once per Kubernetes cluster.

By default the Flink Operator (deployed with installation) watches and manages all the Flink clusters that are deployed in the single same namespace as the Flink Operator. However, you can also configure it to watch and manage multiple namespaces. This allows you to manage multiple Flink clusters deployed in different namespaces, using a single Cloudera Streaming Analytics Operator for Kubernetes installation.

Important: Cloudera SQL Stream Builder can only be deployed in one namespace. In case you require multiple instances, you must install Cloudera SQL Stream Builder in every namespace. For more information about deploying Flink and Cloudera SQL Stream Builder in multiple namespaces, see the Namespace management documentation.

Before you begin

- Ensure that your Kubernetes environment meets requirements listed in System requirements.

  Note:

  If you're installing Cloudera Streaming Analytics Operator for Kubernetes on a Longhorn storage provider (Rancher/RKE2) the following line has to be added to the Helm values.yaml file:

  ```
  ssb:
    database:
      pod:
        securityContext:
          fsGroup: 999
  ```

- Your Kubernetes cluster requires internet connectivity to complete these steps, as it must be able to reach the Cloudera Docker registry.
- Ensure that you have access to a valid Cloudera license.
- Ensure that you have access to your Cloudera credentials (username and password, that are provided together with the Cloudera license) required to access the Cloudera Docker registry (and, if needed, the Cloudera Archive), where installation artifacts are hosted.
- Review the Helm chart reference before installation.

  The Helm chart accepts various configuration properties that you can set during installation. You can customize your installation using these properties.
- If you want to use the Webhook of Flink Operator, ensure that you have cert-manager installed on your Kubernetes cluster, which you can install using the following command:

  ```
  kubectl create -f https://github.com/jetstack/cert-manager/releases/down
  load/v1.8.2/cert-manager.yaml
  kubectl wait -n cert-manager --for=condition=Available deployment --all
  ```

  - The webhook functionality is enabled by default. You can disable it using the following command, and skip the cert-manager installation:

    ```
    --set flink-kubernetes-operator.webhook.create=false
    ```

Procedure

1. Create a namespace in your Kubernetes cluster where you will install and use the Cloudera Streaming Analytics Operator for Kubernetes.

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

   This is the namespace where you install Flink and Cloudera SQL Stream Builder. Use this namespace you create in all installation steps that follow.
2. Create a Kubernetes secret to contain your Cloudera credentials.

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

Ensure that the placeholders are replaced with your specific information:

a) Provide a desired name for *[***SECRET NAME***]*.
b) Replace *[***USERNAME***]* and *[***PASSWORD***]* with your Cloudera credentials.
c) Provide the same name for *[***NAMESPACE***]* that you created in the previous step.

3. Log in to Cloudera Docker registry with helm.

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

Enter your Cloudera credentials when prompted.

4. Install Cloudera Streaming Analytics Operator for Kubernetes with helm install.

```
helm install csa-operator --namespace [***NAMESPACE***] \
    --set 'flink-kubernetes-operator.imagePullSecrets[0].name=[***SECRET
 NAME***]' \
    --set 'ssb.sse.image.imagePullSecrets[0].name=[***SECRET NAME***]' \
    --set 'ssb.sqlRunner.image.imagePullSecrets[0].name=[***SECRET
 NAME***]' \
    --set-file flink-kubernetes-operator.clouderaLicense.fileCon
tent=[***PATH TO LICENSE FILE***] \
oci://container.repository.cloudera.com/cloudera-helm/csa-operator/csa-op
erator --version 1.6.1-b6
```

Important: When you install Cloudera Streaming Analytics Operator for Kubernetes, Cloudera SQL Stream Builder will also be installed by default. In case you want to skip installing it, add --set ssb.enable d=false to the helm install command.

Ensure that the placeholders are replaced with your specific information:

a) Provide the same name for *[***NAMESPACE***]* that you created in Step 1.
b) Provide the same name for *[***SECRET NAME***]* that you created in the previous step. imagePullSecrets specifies what secret is used to pull images from the Cloudera registry. Setting this property is mandatory, otherwise Helm will be unable to pull the necessary images from the Cloudera Docker registry.
c) Replace *[***PATH TO LICENSE FILE***]* with the full (absolute) path to your Cloudera license file. clou deraLicense.fileContent is used to register your license. When this property is set, a secret is generated that contains the license you specify. Setting this property is mandatory. The Cloudera Streaming Analytics Operator for Kubernetes will not function without a valid license.
d) You can use --set to set various other properties of the Helm chart. This enables you to customize your installation. For example, by default the Flink Operator has access to watch all namespaces. However, you can configure a list of specific namespaces to watch using watchNamespaces.. For example, in case you created multiple namespaces, you can configure the Flink Operator to only watch specific ones with --set flink-kubernetes-operator.watchNamespaces={*[***NAMESPACE1***], [***NAMESPACE2***]*}. For more information about deploying Flink and SSB in multiple namespaces, see the Namespace management documentation.

5. Check that the Flink Operator, and the Cloudera SQL Stream Builder engine with its database are running.

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

```
NAME                                READY   STATUS      RESTARTS    AGE
flink-kubernetes-operator           1/2     Running      0          7s
ssb-postgresql                      1/1     Running      0          7s
ssb-sse                             1/1     Running      0          7s
```

What to do next

After successfully installing the Cloudera Streaming Analytics Operator for Kubernetes, you can start using Flink and Cloudera SQL Stream Builder on Kubernetes. The Getting Started with Flink and Getting Started with Cloudera SQL Stream Builder guides can help you with the basic operations.

Install Cloudera Streaming Analytics Operator for Kubernetes in Taikun CloudWorks [Technical Preview]

Installing the Cloudera Streaming Analytics Operator for Kubernetes enables you to deploy and manage Flink and Cloudera SQL Stream Builder in Taikun CloudWorks.

Before you begin

Note: This feature is in Technical Preview and is not ready for production deployments. Cloudera recommends trying this feature in test or development environments and encourages you to provide feedback on your experiences.

Confirm these prerequisites before you start installing Cloudera Streaming Analytics Operator for Kubernetes in a Taikun-managed Kubernetes environment.

- You have access to a project and Kubernetes cluster in Taikun CloudWorks.
- Your Kubernetes environment meets the following sizing requirements:
  - **master:**4 CPU, 16GB RAM
  - **bastion**4 CPU, 16GB RAM
  - **worker (2x):**4 CPU, 16GB RAM
- The worker nodes in your cluster satisfy the minimum sizing guidelines outlined in *Sizing and performance considerations*.
- Access to the cluster with kubectl is configured. For details, see *Accessing Cluster with Kubeconfig*.
- A catalog is available that includes the csa-operator application. For guidance, review *Importing the Cloudera Streaming Analytics Operator for Kubernetes Helm chart into Taikun CloudWorks*.
- Your Kubernetes cluster has outbound internet access to reach the Cloudera Docker registry.
- You have access to your Cloudera Cloudera credentials (username and password).

  Credentials are required to access the Cloudera Archive and Cloudera Docker registry where installation artifacts are hosted.
- You have access to a valid Cloudera Streaming Analytics Operator for Kubernetes 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.
- Install the cert-manager in your project, if it's not already present, so that Cloudera Streaming Analytics Operator for Kubernetes webhooks can operate correctly.

Related Information

Installing applications in Taikun CloudWorks

Cloudera Streaming Analytics Operator for Kubernetes documentation

Helm chart reference

Accessing Cluster with Kubeconfig

Importing the Cloudera Streaming Analytics Operator for Kubernetes Helm chart into Taikun CloudWorks

Import the Cloudera Streaming Analytics Operator for Kubernetes Helm chart repository and add it to a catalog so it is available for project installations.

About this task

Use these steps to make the Cloudera Streaming Analytics Operator for Kubernetes Helm chart available in your Taikun CloudWorks environment.

Procedure

1. Import the Cloudera Streaming Analytics Operator for Kubernetes repository.
   a) In Taikun CloudWorks, go to Repositories and select the Private tab.
   b) Choose Import Repository.
   c) Provide a unique name in Name.
   d) Enter the following OCI repository URL in URL:

      ```
      oci://container.repository.cloudera.com/cloudera-helm/csa-operator/csa-operator
      ```

      .
   e) Enter your Cloudera credentials.
   f) Click Import.
2. Add Cloudera Streaming Analytics Operator for Kubernetes to a catalog.

   These instructions create a new catalog. You can also add your application to an existing catalog.
   a) Go to Catalogs and click + Add Catalog.
   b) Enter the catalog Name and Description.
   c) Click Save.
   d) Go to *[***YOUR CATALOG***]* and click + Add Applications.
   e) Select *[*** YOUR REPIOSITORY***]* from the Repository drop-down list and click Apply.
   f) Find the **Cloudera Streaming Analytics Operator for Kubernetes** in the list of available applications and click +.
   g) Click + Add to the catalog.
3. Add catalog app parameters.
   a) Click Add Parameters.
   b) Find and add the following parameters:

      ```
      clouderaLicense.secretRef
      ```

   c) Set the following default values for the parameters you added:
      - clouderaLicense.secretRef = csa-op-license
   d) Click Save.

Installing Cloudera Streaming Analytics Operator for Kubernetes in Taikun CloudWorks

Deploy the Cloudera Streaming Analytics Operator for Kubernetes application to your Taikun project by using the Helm chart you imported.

About this task

These steps use the Taikun UI to install Cloudera Streaming Analytics Operator for Kubernetes after you published the Helm chart in a catalog and installed cert-manager.

Procedure

1. Create a namespace for the deployment.

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

   Use this namespace for all Cloudera Streaming Analytics Operator for Kubernetes resources.

2. Create a Kubernetes secret that stores your Cloudera license.

```
kubectl create secret generic csa-op-license \
  --namespace [***NAMESPACE***] \
  --from-file=license=[***PATH TO LICENSE FILE***]
```

3. Create a Docker registry secret with your Cloudera credentials.

```
kubectl create secret docker-registry [***REGISTRY CREDENTIALS SECRET***]
 \
  --namespace [***NAMESPACE***] \
  --docker-server container.repository.cloudera.com \
  --docker-username [***USERNAME***] \
  --docker-password "$(echo -n 'Enter your Cloudera password: ' >&2; read
 -s password; echo >&2; echo $password)"
```

   - Take note of the name you specify as *[***REGISTRY CREDENTIALS SECRET***]*. You will need to specify the name in a later step.
   - Replace *[***USERNAME***]* with your Cloudera username.
   - Enter your Cloudera password when prompted.

4. In Taikun CloudWorks, go to Projects*[***YOUR PROJECT***]*Applications.
5. Click + Install.
6. Find the csa-operator.

   Select the application from the catalog that includes the repository you added earlier.

7. Click Bind if you get a prompt to bind the catalog to your project.
8. Configure the following common settings in Application Instance.
   a) Enter a name for Application Instance Name.
   b) In Namespace, select the namespace you created in step 1
   c) Enable the Extra Values tab by clicking the Extra Values toggle.
   d) Click Continue.
9. Configure the following parameters in Installation Params:
   a) Set clouderaLicense.secretRef to the name of the Secret you created in Step 2.
   b) Click Continue.
10. Provide the following values in Extra Values:

```
flink-kubernetes-operator:
  watchNamespaces:
    - [***NAMESPACE***]
  imagePullSecrets:
    - name: [***REGISTRY CREDENTIALS SECRET***]
ssb:
  sqlRunner:
    image:
      imagePullSecrets:
        - name: [***REGISTRY CREDENTIALS SECRET***]
  sse:
    image:
      imagePullSecrets:
        - name: [***REGISTRY CREDENTIALS SECRET***]
```

```
    ingress:
      spec:
        ingressClassName: taikun
        rules:
          - host: ssb.[***YOUR CLUSTER IP***].sslip.io
            http:
              paths:
                - backend:
                    service:
                      name: ssb-sse
                      port:
                        name: sse
                  path: /
                  pathType: ImplementationSpecific
  database:
    image:
      tag: 16.9
    pod:
      securityContext:
        fsGroup: 999
```

Replace *[***REGISTRY CREDENTIALS SECRET***]* with the name of the secret you created in Step 3. Replace *[***YOUR CLUSTER IP***]* with the IP address of *[***YOUR PROJECT***]*.

11. Validate your extra values by clicking Check extra values.
12. Choose Run Installation to deploy the application.
13. Go to ProjectsLiveOps to verify your installation.
14. Go to https://ssb.*[***YOUR CLUSTER IP***]*.sslip.io (see *Taikun Ingress* for more details) in a browser and log in to Cloudera SQL Stream Builder.

Related Information

Taikun Ingress

Installing Cloudera Streaming Analytics Operator for Kubernetes in an air-gapped environment

Complete these steps to install Cloudera Streaming Analytics Operator for Kubernetes if your Kubernetes cluster does not have internet access, or if you want to install from a self-hosted registry. Installing the Cloudera Streaming Analytics Operator for Kubernetes enables you to deploy and manage Flink and Cloudera SQL Stream Builder in Kubernetes.

About this task

Cloudera Streaming Analytics Operator for Kubernetes is installed in your Kubernetes cluster with the provided Helm chart through the helm install command. When you install the chart, Helm installs the Custom Resource Descriptors (CRD) included in Cloudera Streaming Analytics Operator for Kubernetes, and deploys the Apache Kubernetes Flink Operator (Flink Operator), Cloudera SQL Stream Builder engine, and a PostgreSQL (18.4 by default) database for Cloudera SQL Stream Builder.

Installing Cloudera Streaming Analytics Operator for Kubernetes does not create or deploy a Flink cluster. The Flink cluster is created after the installation by deploying the Flink Deployment resource in the Kubernetes cluster with kubectl or oc, or when you execute a SQL job in Streaming SQL Console.



Note: Cloudera recommends that you install Cloudera Streaming Analytics Operator for Kubernetes once per Kubernetes cluster.

By default the Flink Operator (deployed with installation) watches and manages all the Flink clusters that are deployed in the single same namespace as the Flink Operator. However, you can also configure it to watch and manage multiple namespaces. This allows you to manage multiple Flink clusters deployed in different namespaces, using a single Cloudera Streaming Analytics Operator for Kubernetes installation.

Important: Cloudera SQL Stream Builder can only be deployed in one namespace. In case you require multiple instances, you must install Cloudera SQL Stream Builder in every namespace. For more information about deploying Flink and Cloudera SQL Stream Builder in multiple namespaces, see the Namespace management documentation.

Before you begin

- Ensure that your Kubernetes environment meets requirements listed in System requirements.

  Note:

  If you're installing Cloudera Streaming Analytics Operator for Kubernetes on a Longhorn storage provider (Rancher/RKE2) the following line has to be added to the Helm values.yaml file:

  ```
  ssb:
    database:
      pod:
        securityContext:
          fsGroup: 999
  ```

- A self-hosted Docker registry is required. Your registry must be accessible by your Kubernetes cluster.
- While the Kubernetes cluster does not need internet access in an air-gapped environment, the preparation steps to create the local (offline) repository, from which you can install Cloudera Streaming Analytics Operator for Kubernetes, require that you can download and move the artifacts hosted on the Cloudera Docker registry and Cloudera Archive.
- Access to docker or equivalent utility that you can use to pull and push images is required. The Cloudera-recommended way is using docker. Replace commands where necessary, if you use a different utility.
- Ensure that you have access to your Cloudera credentials (username and password). Credentials are required to access the Cloudera Docker registry (and, if needed, the Cloudera Archive) 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 want to use the Webhook of Flink Operator, ensure that you have cert-manager installed on your Kubernetes cluster, which you can install using the following command:

  ```
  kubectl create -f https://github.com/jetstack/cert-manager/releases/down
  load/v1.8.2/cert-manager.yaml
  kubectl wait -n cert-manager --for=condition=Available deployment --all
  ```

  - The webhook functionality is enabled by default. You can disable it using the following command, and skip the cert-manager installation:

    ```
    --set flink-kubernetes-operator.webhook.create=false
    ```

Procedure

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

Table 3: Cloudera Streaming Analytics Operator for Kubernetes artifacts on the Cloudera Docker registry

Artifact	Location (Base image)	Location (Image with connectors)	Description
Flink Kubernetes Operator Docker image	container.repository.cloudera.com/cloudera/flink-kubernetes-operator:1.13-csaop1.6.1-b6		Docker image used for deploying the various operator components shipped with the Cloudera Streaming Analytics Operator for Kubernetes.
Flink Docker image	container.repository.cloudera.com/cloudera/flink:1.20.5-csaop1.6.1-b6		Docker image used for deploying Apache Flink and its related components.
SQL Runner Docker image	container.repository.cloudera.com/cloudera/flink-extended:1.20.5-csaop1.6.1-b6	container.repository.cloudera.com/cloudera/flink-extended-hadoop:1.20.5-csaop1.6.1-b6	Docker image used for deploying Flink application when SQL query is executed in SSB.
SQL Stream Engine Docker image	container.repository.cloudera.com/cloudera/ssb-sse:1.20.5-csaop1.6.1-b6	container.repository.cloudera.com/cloudera/ssb-sse-hadoop:1.20.5-csaop1.6.1-b6	Docker image used for deploying Cloudera SQL Stream Builder and its UI.

Note: The images listed in the column Image with connectors contain the following connectors: Hadoop, Hive, Iceberg, and Kudu.

Note: The images are built for linux/arm64 and linux/amd64 architectures.

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 you set up your registry. The following substeps demonstrate a 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 \
  --platform [***PLATFORM/ARCHITECTURE***] \
  container.repository.cloudera.com/cloudera/[***IMAGE
 NAME***]:[***VERSION***]
```

c) Pull the Cloudera Streaming Analytics Operator for Kubernetes Helm chart.

```
helm pull \
  oci://container.repository.cloudera.com/cloudera-helm/csa-operator/csa
-operator \
  --version 1.6.1-b6
```

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

```
docker tag \
[***ORIGINAL IMAGE TAG***] \
[***REGISTRY HOSTNAME***]:[***PORT***]/cloudera/[***IMAGE NAME***]:
[***VERSION***]
```

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

```
docker push \
[***REGISTRY HOSTNAME***]:[***PORT***]/cloudera/[***IMAGE
 NAME***]:[***VERSION***]
```

```
helm push \
csa-operator-1.6.1-b6.tgz \
oci://[***REGISTRY HOSTNAME***]:[***PORT***]/cloudera-helm/csa-operator/
```

2. Create a namespace in your Kubernetes cluster where you will install and use the Cloudera Streaming Analytics Operator for Kubernetes.

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

This is the namespace where you install Flink and Cloudera SQL Stream Builder. Use this namespace you create in all installation steps that follow.

3. Create a Kubernetes secret to 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***]
```

Ensure that the placeholders are replaced with your specific information:

a) Provide a desired name for *[***SECRET NAME***]*.

b) Replace *[***REGISTRY HOSTNAME***]:[***PORT***]* with your self-hosted registry hostname and port.

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

d) Provide the same name for *[***NAMESPACE***]* that you created in the previous step.

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

```
helm registry login [***REGISTRY HOSTNAME***]:[***PORT***]
```

Enter your credentials when prompted.

5. Install Cloudera Streaming Analytics Operator for Kubernetes with helm install.

```
helm install csa-operator \
    --namespace [***NAMESPACE***] \
    --set 'flink-kubernetes-operator.image.repository=[***REGISTRY
 HOSTNAME***]:[***PORT***]/cloudera/[***IMAGE NAME***]'\
    --set 'ssb.sqlRunner.image.repository=[***REGISTRY
 HOSTNAME***]:[***PORT***]/cloudera/[***IMAGE NAME***]'\
    --set 'ssb.sse.image.repository=[***REGISTRY
 HOSTNAME***]:[***PORT***]/cloudera/[***IMAGE NAME***]'\
    --set 'flink-kubernetes-operator.imagePullSecrets[0].name=[***SECRET
 NAME***]' \
    --set 'ssb.sse.image.imagePullSecrets[0].name=[***SECRET NAME***]' \
    --set 'ssb.sqlRunner.image.imagePullSecrets[0].name=[***SECRET
 NAME***]' \
    --set-file flink-kubernetes-operator.clouderaLicense.fileCon
tent=[***PATH TO LICENSE FILE***] \
oci://[***REGISTRY HOSTNAME***]:[***PORT***]/cloudera-helm/csa-operator/
csa-operator --version 1.6.1-b6
```



Important: When you install Cloudera Streaming Analytics Operator for Kubernetes, Cloudera SQL Stream Builder will also be installed by default. In case you want to skip installing it, add --set ssb.enable d=false to the helm install command.

Ensure that the placeholders are replaced with your specific information:

a) Provide the same name for *[***NAMESPACE***]* that you created in Step 1.
b) Replace *[***REGISTRY HOSTNAME***]*:*[***PORT***]* with your self-hosted registry hostname and port.
c) Provide the same name for *[***SECRET NAME***]* that you created in the previous step. 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.
d) Replace *[***PATH TO LICENSE FILE***]* with the full (absolute) path to your Cloudera license file. clou deraLicense.fileContent is used to register your license. When this property is set, a secret is generated that contains the license you specify. Setting this property is mandatory. The Cloudera Streaming Analytics Operator for Kubernetes will not function without a valid license.
e) You can use --set to set various other properties of the Helm chart. This enables you to customize your installation. (For more information on the available properties, see Helm chart reference.) For example, by default the Flink Operator has access to watch all namespaces. However, you can configure a list of specific namespaces to watch using watchNamespaces.. For example, in case you created multiple namespaces, you can configure the Flink Operator to only watch specific ones with --set flink-kubernetes-operato r.watchNamespaces={*[***NAMESPACE1***]*, *[***NAMESPACE2***]*}. For more information about deploying Flink and Cloudera SQL Stream Builder in multiple namespaces, see the Namespace management documentation.

6. Check that the Flink Operator, and the Cloudera SQL Stream Builder engine with its database are running.

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

```
NAME                                READY   STATUS      RESTARTS    AGE
flink-kubernetes-operator           1/2     Running      0          7s
ssb-postgresql                      1/1     Running      0          7s
ssb-sse                             1/1     Running      0          7s
```

What to do next

After successfully installing the Cloudera Streaming Analytics Operator for Kubernetes, you can start using Flink and Cloudera SQL Stream Builder on Kubernetes. The Getting Started with Flink and Getting Started with Cloudera SQL Stream Builder guides can help you with the basic operations.

Configuring FIPS mode

Configure Cloudera Streaming Analytics Operator for Kubernetes to run on FIPS-enabled OpenShift clusters.

About this task

Cloudera Streaming Analytics Operator for Kubernetes is not FIPS-certified by default. On FIPS-enabled OpenShift clusters installed on FIPS-enabled Red Hat nodes, you must supply FIPS-compatible cryptographic providers and rebuild or retag container images before the Flink Kubernetes Operator webhook and Cloudera SQL Stream Builder pods start successfully.

Crypto provider JARs, truststores, and image registries vary by environment; adjust the examples below for your deployment.

Before you begin

- A FIPS-enabled OpenShift cluster with outbound access to your private container registry.
- Customer-supplied FIPS crypto provider JARs (for example CryptoComply and Bouncy Castle JSSE) and a modified java.security file that registers them.
- Cloudera Streaming Analytics Operator for Kubernetes Helm chart and license file for your target version.

Procedure

1. Install Cloudera Streaming Analytics Operator for Kubernetes with image repository overrides pointing at your registry namespace.

   At minimum, override images for the Flink Kubernetes Operator, Flink extended SQL runner, and Cloudera SQL Stream Builder SSE components in values.yaml.
2. Build custom images for the Flink Kubernetes Operator webhook.
   a) Copy your FIPS provider JARs into a build context directory.
   b) Extract the stock java.security file from a running operator pod and add your providers at the top of the security.provider and fips.provider lists. Set fips.keystore.type to BCFKS.
   c) Build and push the image with docker build --platform linux/amd64 when building on a different architecture.
3. Repeat the image customization for Flink extended and Cloudera SQL Stream Builder SSE images.

   Create BCFKS truststores for TLS endpoints your jobs connect to and mount them through Helm podVolumes / podVolumeMounts as needed.
4. Update values.yaml with the custom image tags and roll out the release.

   Verify the Flink Kubernetes Operator webhook pod reaches Ready state and that Cloudera SQL Stream Builder SSE starts without JSSE or keystore errors.

What to do next



Note: Cloudera Streaming Analytics Operator for Kubernetes OLM metadata continues to report fips-compliant: false. FIPS compliance depends on the providers and truststores you supply.