Cloudera Streaming Analytics Operator 1.5.0

Upgrade

Date published: 2024-06-15
Date modified: 2026-02-18

https://docs.cloudera.com/

Legal Notice

Contents

Upgrading Cloudera Streaming Analytics Operator for Kubernetes

Complete these steps to upgrade Cloudera Streaming Analytics Operator for Kubernetes.

About this task

Upgrading Cloudera Streaming Analytics Operator for Kubernetesconsists of three steps:

1. upgrading the Flink CRDs to the new version,
2. upgrading the cluster operator using Helm commands, and
3. upgrading the Flink jobs with the proper image and Flink version.

Upgrading the cluster operator may affect the Flink resources in the cluster. All Flink clusters that specify the version of the cluster (but not the image) will be restarted during the cluster operator upgrade, due to the fact that the default image of all versions changes with the Flink upgrade, triggering a restart.

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.

Procedure

1. Upgrade the Flink CRDs

   Upgrade the CRDs for FlinkDeployment and FlinkSessionJob resources:

   ```
   # download and extract the helm chart
   helm pull oci://container.repository.cloudera.com/cloudera-helm/csa-operat
   or/csa-operator --version 1.5.0-b275
   tar xf csa-operator-1.5.0-b275.tgz
   kubectl replace -f csa-operator/charts/flink-kubernetes-operator/crds/f
   linkdeployments.flink.apache.org-v1.yml
   kubectl replace -f csa-operator/charts/flink-kubernetes-operator/crds/fl
   inksessionjobs.flink.apache.org-v1.yml
   ```

   Note: Using the replace command ensures that running deployments are unaffected.

2. Upgrade the cluster operator using helm commands

Important:

Before upgrading, compare the version difference between the currently generated and the running yaml files that will be used for backup and restore:

```
helm template [*** CUSTOM SETTINGS ***] csa-operator oci://containe
r.repository.cloudera.com/cloudera-helm/csa-operator/csa-operator --
version 1.5.0-b275 | kubectl diff -f -
```

Upgrade the Helm deployment:

```
helm upgrade [*** CUSTOM SETTINGS ***] csa-operator oci://container.reposi
tory.cloudera.com/cloudera-helm/csa-operator/csa-operator --version 1.5.0-
b275
```

Important: The exact upgrade command depends on your current environment and settings. Please refer to Installation for details.

Note:

You can also upgrade by uninstalling the running Helm deployment and install a new version:

```
helm uninstall csa-operator
helm install [*** CUSTOM SETTINGS ***] csa-operator oci://containe
r.repository.cloudera.com/cloudera-helm/csa-operator/csa-operator --
version 1.5.0-b275
```

3. Upgrade the Flink jobs with the proper image and Flink version
   a) Updated custom Flink applications to use the latest Flink and connector versions in the pom.xml file of the project:
      1. Update the flink.version property to 1.20.1-csaop1.5.0-b275
      2. Update all connectors to match the version. For example:

```
<flink.version<1.20.1-csaop1.5.0-b275<flink.version<
<flink.cloudera.connector.version<1.0-csaop1.1.0<flink.cloudera.conn
ector.version<
<flink.kafka.connector.version<3.2-csaop1.1.0<flink.kafka.connector.v
ersion<
```

Note: Refer to catalog.txt for a full list of connectors and version numbers.

   b) Update any custom docker images built for Flink jobs to be based on the latest Flink Image:

```
# Dockerfile
FROM container.repository.cloudera.com/cloudera/flink:1.20.1-csaop1.5.0-
b275
```

   c) Rebuild any custom Flink application jars and/or docker images and push them to their respective repositories.
   d) Suspend all running Flink jobs:

```
kubectl --namespace [*** NAMESPACE ***] patch FlinkDeployment [*** DEPL
OYMENT NAME ***] --type=merge --patch='{"spec":{"job":{"state":"suspende
d"}}}'
```

```
kubectl --namespace [*** NAMESPACE ***] patch FlinkSessionJob [*** DEPLO
YMENT NAME ***] --type=merge --patch='{"spec":{"job":{"state":"suspended
"}}}'
```

e) Update all FlinkDeployment objects to use the proper image and flinkVersion:

```
apiVersion: flink.apache.org/v1beta1
kind: FlinkDeployment
metadata:
  name: [*** NAME ***]
  namespace: [*** NAMESPACE ***]
spec:
  flinkVersion: v1_19
  image: container.repository.cloudera.com/cloudera/flink:1.20.1-cs
aop1.5.0-b275
  [...]
```

f) If the name of the jar files have changed because of the update, update the jarURI properties of the FlinkDeployment and FlinkSessionJob objects accordingly:

```
apiVersion: flink.apache.org/v1beta1
kind: FlinkDeployment # or FlinkSessionJob
metadata:
  name: [*** NAME ***]
  namespace: [*** NAMESPACE ***]
spec:
  [...]
  job:
     jarURI: local:///opt/flink/example.jar # ensure this is correct
  [...]
```

g) Resume the suspended Flink jobs:

```
kubectl --namespace [*** NAMESPACE ***] patch FlinkDeployment [***
 DEPLOYMENT NAME ***] --type=merge --patch='{"spec":{"job":{"state":"
running"}}}'

kubectl --namespace [*** NAMESPACE ***] patch FlinkSessionJob [***
 DEPLOYMENT NAME ***] --type=merge --patch='{"spec":{"job":{"state":"
running"}}}'
```

Note: The Flink jobs will resume from where they were stopped, provided they were configured to use savepoints.