

## Getting Started with Flink

Date published: 2024-06-15

Date modified: 2024-11-15

The Cloudera logo is displayed in a bold, orange, sans-serif font. The word "CLOUDERA" is written in all caps, with a stylized 'E' that has a horizontal bar extending to the right.

# Legal Notice

© Cloudera Inc. 2025. 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

<b>Deploying a Flink application.....</b>	<b>4</b>
---	----------

## Deploying a Flink application

This Getting Started guide walks you through how to deploy a Flink application on a Kubernetes cluster using the Cloudera Streaming Analytics - Kubernetes Operator.

### What is Apache Flink?

Flink is a distributed processing engine and a scalable data analytics framework. You can use Flink to process data streams at a large scale and to deliver real-time analytical insights about your processed data with your streaming application.

Flink is designed to run in all common cluster environments, perform computations at in-memory speed and at any scale. Furthermore, Flink provides communication, fault tolerance, and data distribution for distributed computations over data streams. A large variety of enterprises choose Flink as a stream processing platform due to its ability to handle scale, stateful stream processing, and event time.

### About this task

This Getting Started guide walks you through how to deploy a Flink application on a Kubernetes cluster using the Cloudera Streaming Analytics - Kubernetes Operator. The Getting Started guide uses the Flink Kubernetes Tutorial from the *Flink tutorials public repository*.



**Note:** The following steps require internet connectivity. In an air-gapped environment additional steps are required to build the repository. For more information, see the *Installing Cloudera Streaming Analytics - Kubernetes Operator in an air-gapped environment*.

### Procedure

1. Clone the Flink Tutorials repository, and build the tutorial Docker image that contains the built Flink JAR file.

```
git clone https://github.com/cloudera/flink-tutorials.git -b CSA-OPERATOR-1.1.2
cd flink-tutorials/flink-kubernetes-tutorial
mvn clean package
docker build -t flink-kubernetes-tutorial .
```

2. Tag the tutorial image to attach the required information about the Docker registry that you will use in the next step.

```
docker image tag flink-kubernetes-tutorial [***REGISTRY HOST***]:[***PORT***]/[***PROJECT***]/flink-kubernetes-tutorial:latest
```

3. Push the newly tagged tutorial image to your Docker registry. This way the Kubernetes nodes can download the image from the (local or public) Docker registry.

```
docker push [***REGISTRY HOST***]:[***PORT***]/[***PROJECT***]/flink-kubernetes-tutorial:latest
```

You will see the Image successfully pushed message when the image is correctly pushed.

4. Create the `FlinkDeployment` configuration file by saving the following example as `flink-deployment.yaml`. Make sure to set `spec.image` to the image that you have pushed in the previous step.

```
apiVersion: flink.apache.org/v1beta1
kind: FlinkDeployment
metadata:
  name: flink-kubernetes-tutorial
spec:
```

```

image: [***REGISTRY HOST***]:[***PORT***]/[***PROJECT***]/flink-kuber
netes-tutorial:latest
flinkVersion: v1_19
flinkConfiguration:
  taskmanager.numberOfTaskSlots: "4"
serviceAccount: flink
mode: native
jobManager:
  resource:
    memory: "2048m"
    cpu: 1
taskManager:
  resource:
    memory: "2048m"
    cpu: 1
job:
  args: ["--rowsPerSec", "10"]
  jarURI: local:///opt/flink/usrlib/flink-kubernetes-tutorial.jar
  parallelism: 4
  state: running
  upgradeMode: stateless

```

5. Run the Flink job by applying the FlinkDeployment configuration file to the cluster using the following command:

```
kubectl -n flink apply -f flink-deployment.yaml
```

The Flink Operator automatically recognizes the new resource and starts the execution of the Flink job, which is reflected by JOB STATUS switching to RUNNING:

```

kubectl -n flink get FlinkDeployment
NAME                                JOB STATUS  LIFECYCLE STATE
flink-kubernetes-tutorial          RUNNING    STABLE

```

6. Configure access to the Flink Dashboard at <http://localhost:8081> (or IP address/domain name of your machine, if it differs from localhost) using port-forwarding with the following command:

```
kubectl -n flink port-forward service/flink-kubernetes-tutorial-rest 8081:8081
```

All requests to the <http://localhost:8081> URL will be forwarded to the Flink service while the kubectl command is running. To stop the port-forwarding, exit the kubectl command by pressing CTRL+C.

### Related Information

[Flink tutorials public repository](#)

[Installing CSA Operator in an air-gapped environment](#)