Cloudera Streaming Analytics 1.3.0

Quick Start

Date published: 2019-12-17
Date modified: 2021-03-25

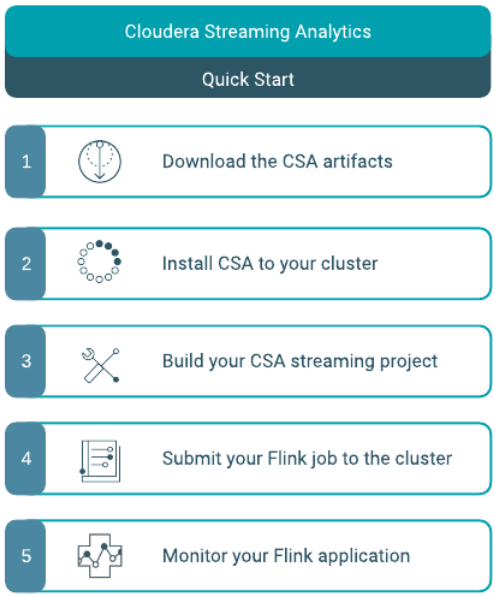
https://docs.cloudera.com/

Legal Notice

Contents

Quick Start for Flink

You can use the Quick Start to easily get familiar with downloading, installing Flink on CDP Private Cloud Base, and also building, submitting and monitoring your Flink application. You can learn more information by clicking on the high level steps covered in the illustration.



rect 38, 136, 443, 196 https://docs.cloudera.com/csa/1.3.0/download/topics/csa-download-location.html
rect 39, 221, 443, 282 https://docs.cloudera.com/csa/1.3.0/installation/topics/csa-installing-parcel.html
rect 41, 302, 442, 362 https://docs.cloudera.com/csa/1.3.0/quickstart/topics/csa-flink-archetype.html
rect 41, 384, 444, 443 https://docs.cloudera.com/csa/1.3.0/job-lifecycle/topics/csa-run-job.html
rect 40, 466, 443, 523 https://docs.cloudera.com/csa/1.3.0/monitoring/topics/csa-hs-webui.html

Flink Project Template

The Quickstart Archetype serves as a template for a Flink streaming application. You can use the Archetype to add source, sink and computation to the template. Like this you can practice the development of a simple Flink application, or use the Archetype as the starting point for a more complex application including state,watermark and checkpoint.

About this task

The Flink quickstart archetype can be used to quickly build a basic Flink streaming project.

Note: You need to install the archetype locally on your host as Cloudera does not release maven archetypes to the Maven Central Repository.

Procedure

1. Perform the following commands to create the archetype locally:

```
git clone https://github.com/cloudera/flink-tutorials
```

```
cd flink-tutorials
cd flink-quickstart-archetype
mvn clean install
cd ..
```

The following entry should be seen in your local catalog:

```
cat ~/.m2/repository/archetype-catalog.xml
 ...
 <archetypes>
    <archetype>
      <groupId>com.cloudera.flink</groupId>
      <artifactId>flink-quickstart-archetype</artifactId>
      <version>1.10.0-csa1.2.0.0</version>
      <description>flink-quickstart-archetype</description>
    </archetype>
  </archetypes>
```

2. Generate the project skeleton with the following commands:

```
mvn archetype:generate \
      -DarchetypeGroupId=com.cloudera.flink \
      -DarchetypeArtifactId=flink-quickstart-archetype \
      -DarchetypeVersion=1.10.0-csa1.2.0.0
```

3. Provide basic information about the streaming application.

   You can choose to customize configurations yourself or use automatically generated information.

   - To use automatically generated information, press Enter.
   - If you choose to customize configuration, set the following properties:

     ```
     Define value for property 'groupId': com.cloudera.flink
                                     Define value for property 'artifactId':
     sample-project
                                     Define value for property 'version' 1.0-
     SNAPSHOT: :
                                     Define value for property 'package' c
     om.cloudera.flink: :
                                     Confirm properties configuration:
                                     groupId: com.cloudera.flink
                                     artifactId: sample-project
                                     version: 1.0-SNAPSHOT
                                     package: com.cloudera.flink
                                     Y: :
     ```

   The generated project will look like this:

   ```
   sample-project
   ### pom.xml
   ### src
       ### main
           ### java
           #   ### com
           #       ### cloudera
           #           ### flink
           #               ### StreamingJob.java
           ### resources
               ### log4j.properties
   ```

4. Open StreamingJob.java file.

   The archetype application will look like this:

```
public class StreamingJob {
 public static void main(String[] args) throws Exception {
  final StreamExecutionEnvironment env = StreamExecutionEnvironment.getEx
ecutionEnvironment();
  DataStream<Integer> ds = env.fromElements(1,2,3,4);
  ds.printToErr();
  env.execute("Flink Streaming Java API Skeleton");
 }
}
```

5. Customize the archetype application by adding source, stream transformation and sink to the Datastream class.
6. Execute the application with env.execute command.

Results

You have built your Flink streaming project.

What to do next

You can further develop your application, or you can run the Flink application archetype.