Cloudera Streaming Analytics 1.4.0

Apache Flink References

Date published: 2019-12-17
Date modified: 2021-05-25

https://docs.cloudera.com/

Legal Notice

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.

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

Flink Terminology

The list of Flink terminology details the Flink specific terms that are used in the Cloudera Streaming Analytics documentation.

Event

An event is a statement about a change of the state of the domain modelled by the application. Events can be input and/or output of a stream or batch processing application. Events are special types of records.

Function

Functions are implemented by the user and encapsulate the application logic of a Flink program. Most Functions are wrapped by a corresponding Operator.

Flink Application

A Flink application is a Java Application that submits one or multiple Flink Jobs from the main() method. Submitting jobs is usually done by calling execute() on an execution environment.

Flink Job

A Flink Job is the runtime representation of a logical graph (also often called dataflow graph) that is created and submitted by calling execute() in a Flink Application.

Flink JobManager

The JobManager is the orchestrator of a Flink Cluster. It contains three distinct components: Flink Resource Manager, Flink Dispatcher and one Flink JobMaster per running Flink Job.

Logical Graph

A logical graph is a directed graph where the nodes are Operators and the edges define input/output-relationships of the operators and correspond to data streams or data sets. A logical graph is created by submitting jobs from a Flink Application. Logical graphs are also often referred to as dataflow graphs.

Operator

Node of a Logical Graph. An Operator performs a certain operation, which is usually executed by a Function. Sources and Sinks are special Operators for data ingestion and data egress.

Flink Session Cluster

A long-running Flink Cluster which accepts multiple Flink Jobs for execution. The lifetime of this Flink Cluster is not bound to the lifetime of any Flink Job.

State Backend

For stream processing programs, the State Backend of a Flink Job determines how its state is stored on each TaskManager (Java Heap of TaskManager or (embedded) RocksDB) as well as where it is written upon a checkpoint (Java Heap of JobManager or Filesystem).

Task

Node of a Physical Graph. A task is the basic unit of work, which is executed by Flink’s runtime. Tasks encapsulate exactly one parallel instance of an Operator or Operator Chain.

Flink TaskManager

TaskManagers are the worker processes of a Flink Cluster. Tasks are scheduled to TaskManagers for execution. They communicate with each other to exchange data between subsequent Tasks.

Transformation

A Transformation is applied on one or more data streams or data sets and results in one or more output data streams or data sets. A transformation might change a data stream or data set on a per-record basis, but might also only change its partitioning or perform an aggregation. While Operators and Functions are the “physical” parts of Flink’s API, Transformations are only an API concept. Specifically, most transformations are implemented by certain Operators.

For the complete list of Flink terminology, see the Apache documentation.

Cloudera Flink Tutorials

The Cloudera Flink Tutorials walks you through the basic steps to create a Stateless Monitoring, a Stateful Inventory and a Secure application using Flink.

For newcomers, Cloudera recommends starting with the Simple Tutorial.

The Simple Tutorial details the following steps:

- Basic structure of a Flink application
- Logging with Kafka
- Job submission
- Alerting functionality

For a more advanced application, you can use the Stateful Tutorial to get familiar with using state and windowing.

The Stateful Tutorial details the following steps:

- Using state within the application
- Windowing function
- Generating data from Kafka
- Production configuration

The Secure Tutorial serves as a first step to learn everything about the Flink security features within Cloudera.

The Secure Tutorial details the following steps:

- Securing Kafka
- Security parameters in a Flink job
- Kafka Metrics Reporter
- Integration with Schema Registry