Cloudera Streaming Analytics 1.11.1

SQL Stream Overview

Date published: 2019-12-17
Date modified: 2024-04-16

https://docs.cloudera.com/

Legal Notice

Contents

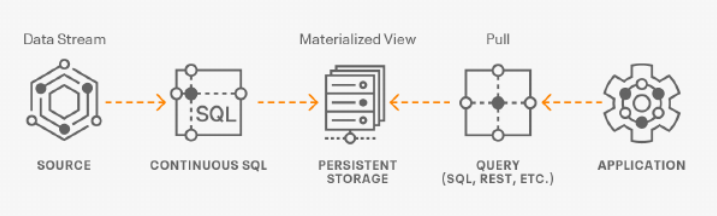
What is SQL Stream Builder?

Cloudera Streaming Analytics offers an easy to use and interactive SQL Stream Builder as a service to create queries on streams of data through SQL.

The SQL Stream Builder (SSB) is a comprehensive interactive user interface for creating stateful stream processing jobs using SQL. By using SQL, you can simply and easily declare expressions that filter, aggregate, route, and otherwise mutate streams of data. SSB is a job management interface that you can use to compose and run SQL on streams, as well as to create durable data APIs for the results.

What is Continuous SQL?

SSB runs Structured Query Language (SQL) statements continuously, this is called Continuous SQL or Streaming SQL. Continuous SQL can run against both bounded and unbounded streams of data. The results are sent to a sink of some type, and can be connected to other applications through a Materialized View interface. Compared to traditional SQL, in Continuous SQL the data has a start, but no end. This means that queries continuously process results. When you define your job in SQL, the SQL statement is interpreted and validated against a schema. After the statement is executed, the results that match the criteria are continuously returned.



Integration with Flink

SSB runs in an interactive fashion where you can quickly see the results of your query and iterate on your SQL syntax. The executed SQL queries run as jobs on the Flink cluster, operating on boundless streams of data until canceled. This allows you to author, launch, and monitor stream processing jobs within SSB as every SQL query is a Flink job. You can use Flink and submit Flink jobs without using Java, as SSB automatically builds and runs the Flink job in the background.

As a result of Flink integration, you are able to use the basic functionalities offered by Flink. You can choose exactly once processing, process your data stream using event time, save your jobs with savepoints, and use Flink SQL to create tables and use connectors based on your requirements. As a result of the various connectors, you are able to enrich your streaming data with data from slowly changing connectors.

Key features of SSB

SQL Stream Builder (SSB) within Cloudera supports out-of-box integration with Flink. Using Flink SQL, you can create tables directly from the Streaming SQL Console window or built-in templates. For integration with Business Intelligence tools you can create Materialized Views.

Flink SQL

SQL Stream Builder allows you to use Data Definition Language (DDL), Data Manipulation Language (DML) and Query Language directly from the Streaming SQL Console.

For more information about the supported Flink SQL statements, see the Flink SQL statements section.

Change Data Capture

SQL Stream Builder supports PostgreSQL, Oracle, MySQL, Db2 and SQL Server as Debezium connectors using Flink SQL. With Change Data Capture (CDC), you can capture changes in your databases and update your applications with the newly added data.

For more information about Debezium, see the official documentation.

Session Cluster

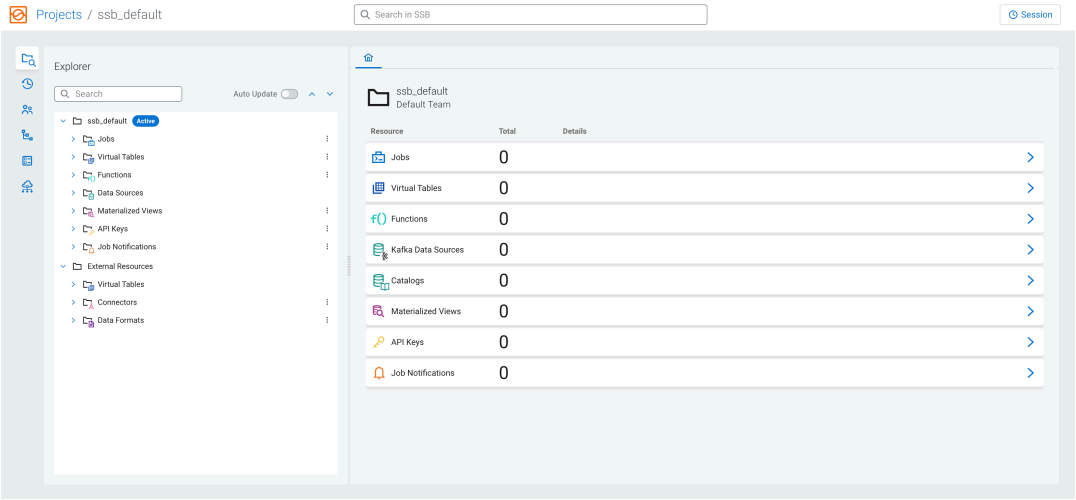
When you start SSB, a session cluster is deployed to share and maintain resources. The submitted SQL jobs are executed as Flink jobs in the same session cluster that share a Job Manager. The properties for the session cluster can be viewed using the Streaming SQL Console, and can be modified with SET statements in the SQL window.

Built-in Templates

The Build-in Templates in SSB allows you to quickly and simply create tables for your SQL queries. You only need to provide the connection and job specific information to the template to use it in SSB.

Streaming SQL Console

SSB comes with an interactive user interface that allows you to easily create, and manage your SQL jobs in one place. It allows you to create and iterate on SQL statements with robust tooling and capabilities. Query parsing is logged to the console, and results are sampled back to the interface to help with iterating on the SQL statement as required.



Materialized Views

SSB has the capability to materialize results from a Streaming SQL query to a persistent view of the data that can be read through REST and over the PG wire protocol. Applications can use this mechanism to query streams of data in a way of high performance without deploying additional database systems. Materialized Views are built into the SQL Stream Builder service, and require no configuration or maintenance. The Materialized Views act like a special kind of sink, and can even be used in place of a sink. They require no indexing, storage allocation, or specific management.

Input Transform

In case you are not aware of the incoming data structure or raw data is being collected from for example sensors, you can use the Input Transform to clean up and organize the incoming data before querying. Input transforms also allow access to Kafka header metadata directly in the query itself. Input transforms are written in Javascript and compiled to Java bytecode deployed with the Flink jar.

User-defined Functions

You can create customized and complex SQL queries by using User-defined Functions to enrich your data, apply computations or a business logic on it. User defined functions are written in Javascript or Java language.

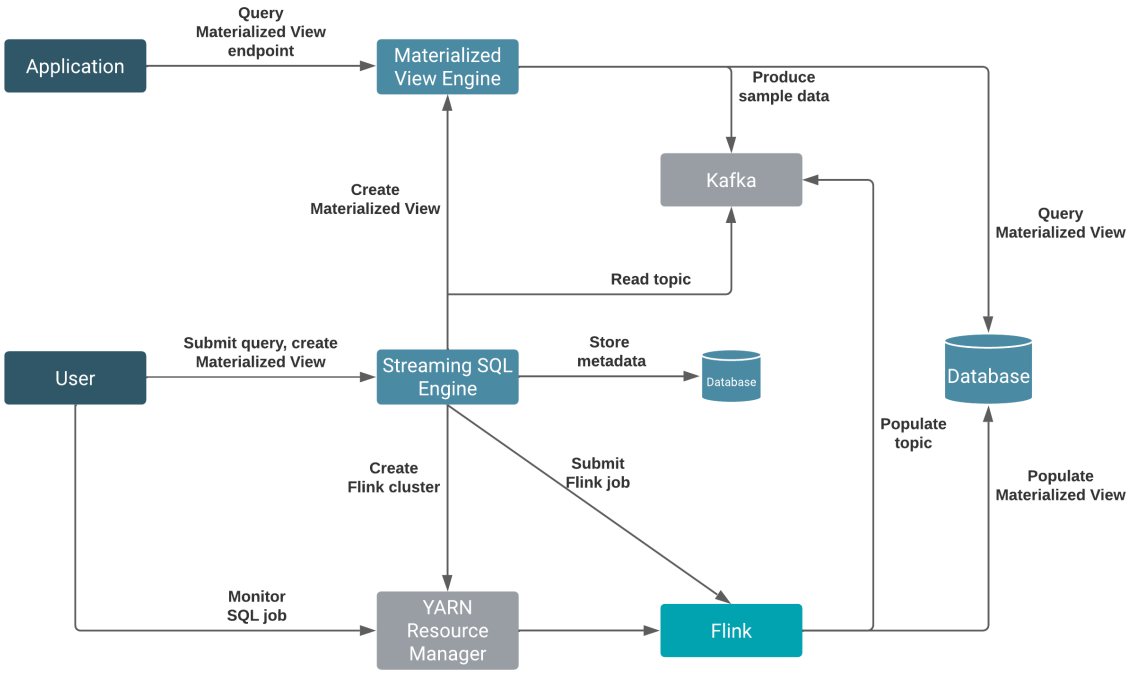
Related Information
Integration with Flink

SQL Stream Builder architecture

The SQL Stream Builder (SSB) service is integrated on Cloudera Data Platform (CDP) and connected to Flink and its services. The SSB architecture includes SQL Stream Engine and Materialized View Engine. These main components within SSB are responsible for executing jobs, populating topics, creating metadata and querying data that happens in the background.

SSB consists of the following main components:

- SQL Stream Engine
- Materialized View Engine



The primary point of user interaction for SQL Stream Builder is the Streaming SQL Console User Interface running on the Streaming SQL Engine. When a query is submitted on the Streaming SQL Console, the Streaming SQL Engine automatically creates a Flink job in the background on the cluster. SSB also requires a Kafka service on the same cluster. This mandatory Kafka service is used to automatically populate topics for the websocket output. The websocket output is needed for sampling data to the Console, and when no table is added to output the results of the SQL query.

When a Materialized View query is submitted, Flink generates the data to the Materialized View database from which the Materialized View Engine queries the required data.

Database management in SSB

SSB uses databases in the following cases:

- To store metadata of SQL jobs
- To store data for creating Materialized Views
- As a connector for Flink SQL

The Streaming SQL Console and the Materialized Views need databases where the metadata of SQL jobs are stored and from which the Materialized View Engine queries data to create the views. SSB supports MySQL/MariaDB, PostgreSQL and Oracle as databases for the Streaming SQL Engine. You can only use PostgreSQL for the Materialized View Engine. When using the JDBC connector in Flink SQL, you can use MySQL and PostgreSQL from the supported databases.

When configuring the databases for SSB, you only set the databases that are used for metadata management. When you want to use the JDBC connector, you need to add the database information to the CREATE TABLE statement using the JDBC connection parameter.