Cloudera Streaming Analytics 1.18.0

Cloudera SQL Stream Builder Overview

Date published: 2019-12-17
Date modified: 2026-09-16



https://docs.cloudera.com/

Legal Notice

© Cloudera Inc. 2026. 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

What is Cloudera SQL Stream Builder?.....4

Key features of Cloudera SQL Stream Builder.....4

Cloudera SQL Stream Builder architecture.....6

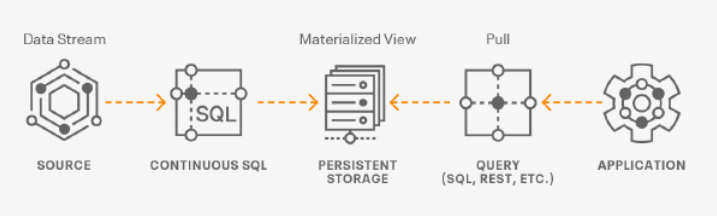
What is Cloudera SQL Stream Builder?

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

Cloudera SQL Stream Builder 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. Cloudera SQL Stream Builder 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?

Cloudera SQL Stream Builder 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

Cloudera SQL Stream Builder 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 Cloudera SQL Stream Builder as every SQL query is a Flink job. You can use Flink and submit Flink jobs without using Java, as Cloudera SQL Stream Builder 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 Cloudera SQL Stream Builder

Cloudera SQL Stream Builder 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

Cloudera 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

Cloudera 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

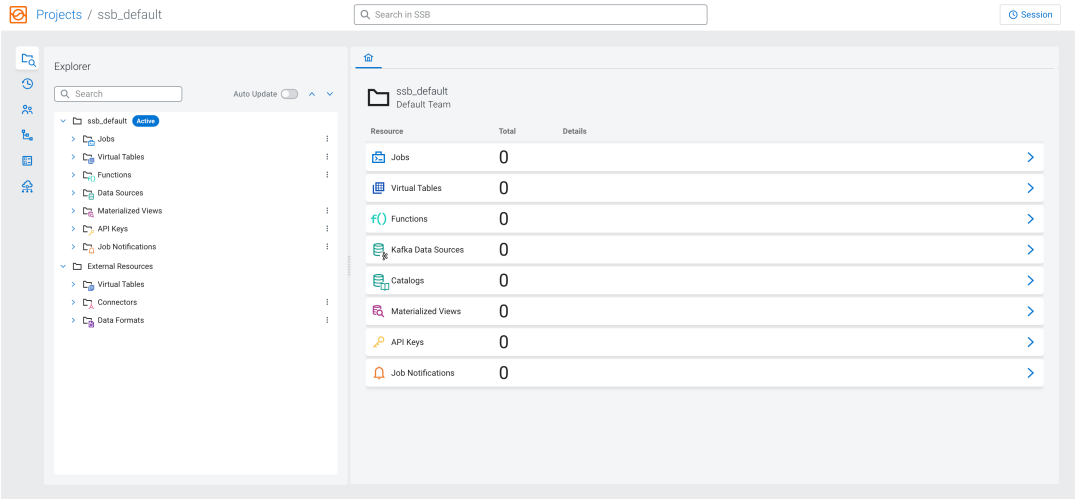
When you start Cloudera SQL Stream Builder, a session cluster is deployed to share and maintain resources. The submitted SQL jobs are executed as Flink jobs in the same session cluster that share a Job Manager. The properties for the session cluster can be viewed using the Streaming SQL Console, and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

Built-in Templates

The Built-in Templates in Cloudera SQL Stream Builder allow 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 Cloudera SQL Stream Builder.

Streaming SQL Console

Cloudera SQL Stream Builder 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

Cloudera SQL Stream Builder 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.

User-defined Functions

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

Related Information

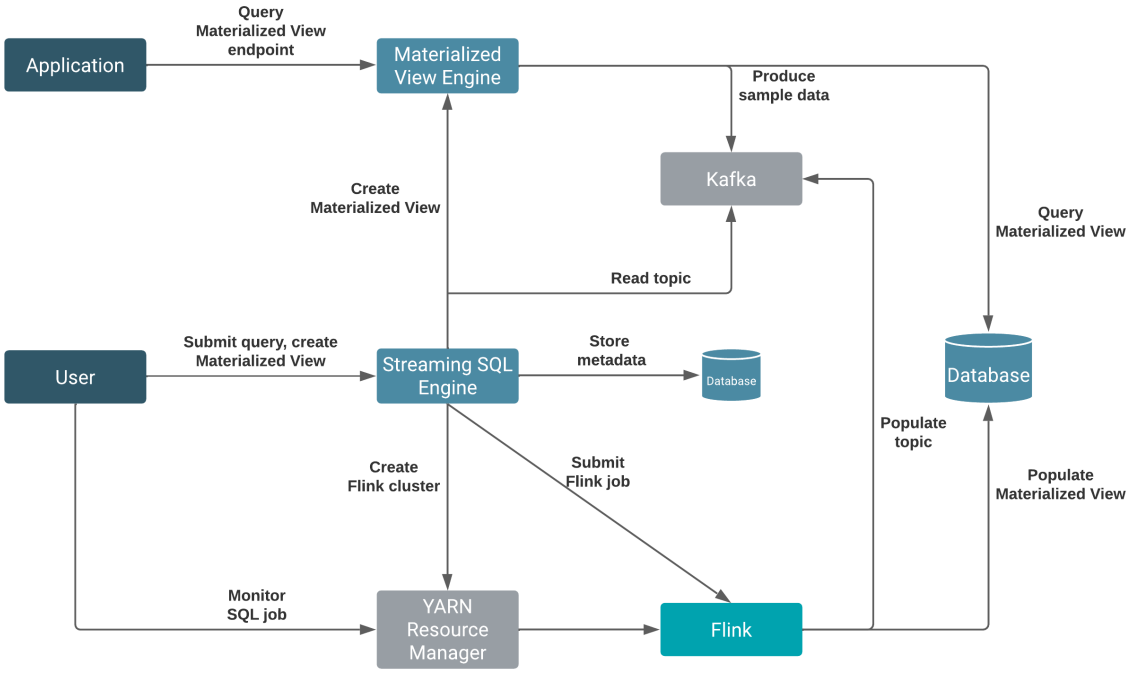
Integration with Flink

Cloudera SQL Stream Builder architecture

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

Cloudera SQL Stream Builder consists of the following main components:

- SQL Stream Engine
- Materialized View Engine



The primary point of user interaction for Cloudera 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. Cloudera SQL Stream Builder also requires a Kafka service on the same cluster. This mandatory Kafka service is used to automatically populate topics for the Server-Side Events (SSE) output. The SSE 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 Cloudera SQL Stream Builder

Cloudera SQL Stream Builder 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. Cloudera SQL Stream Builder 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 Cloudera SQL Stream Builder, 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.