

SQL Stream Job Lifecycle

Date published: 2019-12-17

Date modified: 2021-05-25



Legal Notice

© Cloudera Inc. 2024. 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

Running SQL Stream jobs.....	4
Configuring advanced job management.....	6
Stopping, restarting and editing SQL jobs.....	6
Sampling data for a running job.....	7

Running SQL Stream jobs

Every time you run an SQL statement in the SQL Stream console, it becomes a job and runs on the deployment as a Flink job. You can manage the running jobs using the Jobs tab on the UI.

About this task

There are two logical phases to run a job:

1. **Parse:** The SQL is parsed and checked for validity and then compared against the table schema(s) for correct typing and key/columns.
2. **Execution:** If the parse phase is successful, a job is dynamically created, and runs on an open slot on your cluster. The job is a valid Flink job.

Before you begin

- Make sure that you have registered a Data Provider.
- Make sure that you have created a Table that can be used as a source in the SQL query.

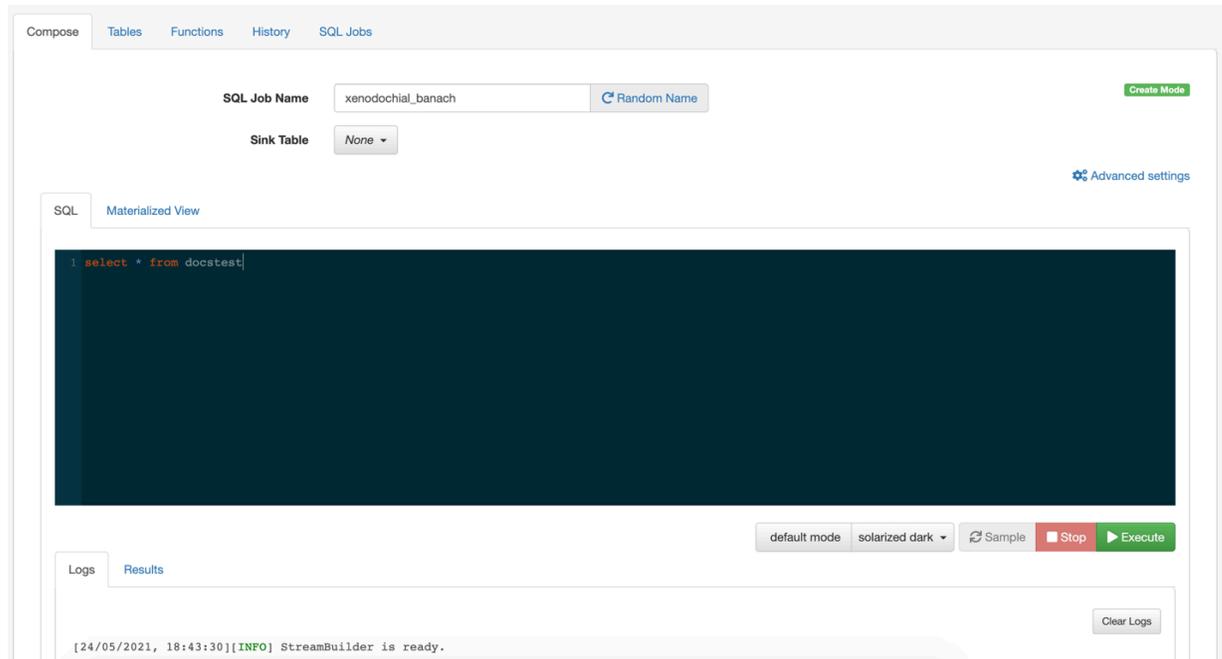
Procedure

1. Go to your cluster in Cloudera Manager.
2. Click SQL Stream Builder from the list of services.
3. Click SQLStreamBuilder Console.
The Streaming SQL Console opens in a new window.
4. Provide a name for the SQL job.
 - a) Optionally, you can click Random Name to generate a name for the SQL job.
5. Select a Sink Table.
 - a) Optionally, you can leave the sink as None.



Note: The Sink Table is an optional argument to a job that specifies the destination for the continuous results. If you select None, the results are not sent to a sink, but sampled on the screen or to a materialized view.

6. Add a SQL query to the SQL window.

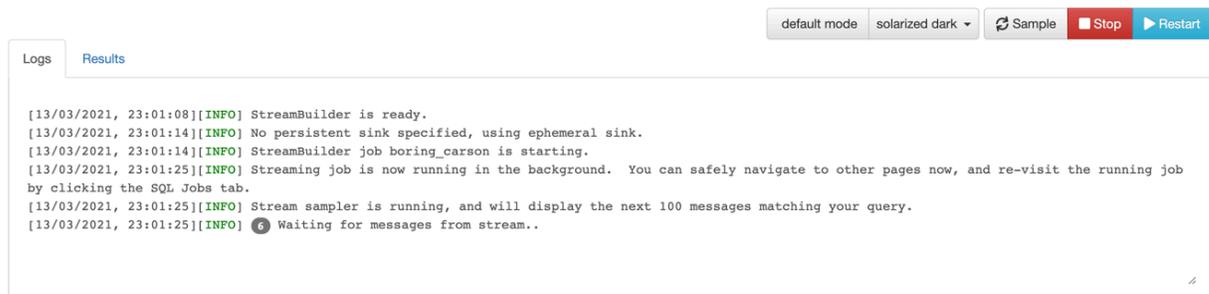



Note: You cannot start a job without adding a SQL statement in the SQL editor window. In case, there are no SQL statements provided and you click Execute, the following error message is displayed: You must provide a SQL query.

When starting a job, the number of slots consumed on the specified cluster is equal to the parallelism setting. The default is one slot. To change the parallelism setting, click Advanced settings.

7. Click Execute.

The Logs window updates the status of SSB.



8. Click Results to check the sampled data.

These results are only samples, not the entire result of the new stream being created from the output of the query. The entire result set is sent to the Sink Table and/or a Materialized View.



Note: While SSB is querying unbounded data streams, you can click Stop to stop the job execution.

Results

A job is generated that runs the SQL continuously on the stream of data from the source Table, and pushes the results to a sink Table, to the Console under the Results tab or to a Materialized View.

Related Information

[Registering Data Providers in SSB](#)

[Using Tables in SQL Stream jobs](#)

[Monitoring SQL Stream jobs](#)

Configuring advanced job management

If you need to further customize your SQL Stream job, you can add more advanced features to configure the job restarting method and time, threads for parallelism, sample behavior, exactly once processing and restoring from savepoint.

Before running a SQL query, you can configure advanced features by clicking on the Show Advanced Settings button on the Compose tab.

The screenshot shows the 'Console' interface for configuring an SQL job. The main heading is 'Console' with a subtitle 'Run SQL against unbounded streams of data and create persistent SQL streaming jobs'. Below this, there are tabs for 'Compose', 'Tables', 'Functions', 'History', and 'SQL Jobs'. The 'Compose' tab is active, showing a form for configuring an SQL job. The form includes the following fields:

- SQL Job Name:** A text input field containing 'confident_mirzakhani' and a 'Random Name' button.
- Sink Table:** A dropdown menu set to 'None'.
- Restart Strategy:** A dropdown menu set to 'never'.
- Restart Retry Time(sec):** A text input field containing '30'.
- Job Parallelism (threads):** A text input field containing '1'.
- Sample Behavior:** A dropdown menu set to 'Sample one message every second'.
- Restore From Savepoint:** A dropdown menu set to 'false'.

At the bottom right of the form, there is a 'Hide Advanced settings' button. At the bottom left, there are tabs for 'SQL' and 'Materialized View'.

Restart strategy and restart retry time

The job is restarted after Restart Retry Time seconds, if set to Always. It is not restarted if you select Stop from the SQL Jobs tab. If set to Never, the job does not restart unless you select Restart from the Compose tab.

Job parallelism

The number of threads to start to process the job. Each thread consumes a slot on the cluster.

Sample interval

How often to sample data (in milliseconds) from the output stream. 1000ms is common and recommended.

Exactly Once Processing

Enabling the Flink exactly once processing feature for the data that is generated to the sink.

Restore From Savepoint

Enabling restoring from savepoint for the SQL job, using the state in Flink.

Stopping, restarting and editing SQL jobs

As a SQL Stream job processes streaming data, you need to stop the job to finish the process. You can restart a SQL Stream job after stopping it. In case you need to update or change the configurations that you have set for a SQL Stream job, you can restart it. You can navigate through the job life cycle using the Streaming SQL Console.

Stopping a SQL Stream job

1. Select Console from the left hand menu.

2. Go to the SQL Jobs tab.
3. Click red stop corresponding the job you would like to stop.

Restarting a SQL Stream job

1. Select Console from the left hand menu.
2. Go to the SQL Jobs tab.
3. Select Cancelled or Failed from the State drop-down menu to see the stopped jobs.
4. Select the job you want to restart.
5. Go to the Details tab at the bottom.
6. Click Edit Selected Job .

The SQL window in Edit Mode appears.

7. Click Restart .

Editing a SQL Stream job



Note: You need to stop the SQL Stream job before editing it.

1. Select Console from the left hand menu.
2. Go to the SQL Jobs tab.
3. Select the job you want to edit.
4. Go to the Details tab at the bottom.
5. Click Edit Selected Job .

This brings up the SQL window in Edit Mode.

6. Edit the Target Deployment, Sink Table, and the SQL itself as needed.
7. Select Restart to restart the job.

The job will be stopped and restarted.

Sampling data for a running job

You can sample data from a running job. This is useful if you want to inspect the data to make sure the job is running correctly and producing the results you expect.

About this task

Sampling the results to your browser allows you to inspect the queried data and iterate on your query. You can sample 100 rows in the Results tab by clicking on the Sample button in the Console. In case you do not add any sink to the SQL job, the results automatically appear in the Results tab.

Procedure

1. Select Console on the main menu.
2. Go to the SQL Jobs tab.
3. Select the job you want to edit.
4. Go to the Details tab at the bottom.
5. Click Edit Selected Job.
The SQL window in Edit Mode appears.
6. Click Sample.

Results

Sample results are displayed in the results window. If there is no data meeting the SQL query, sampling stops after a few attempts.