

SQL Client

Date published: 2020-05-11

Date modified:

CLOUDERA

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

SQL Client configuration.....	4
Configuring SQL Client for session mode.....	4
SQL Client security.....	5

SQL Client configuration

The SQL client provides a simple and efficient command line tool to interactively develop and submit Flink SQL queries to your clusters without using Java code. The SQL Client CLI enables you to use the command line for retrieving and visualizing real-time results from the running distributed applications.



Important: The SQL Client is provided as a technical preview at this time. The tool is still under development and not recommended for a production environment.

You can control the Flink SQL Client behaviour by the following layers of configuration:

- Regular Flink configuration (flink-conf.yaml)
- SQL Client default settings (sql-client-defaults.yaml)
- Custom environment settings

The regular Flink configuration is shared with all other Flink jobs, and you can configure it through Cloudera Manager.

The SQL Client default settings file is located at `/etc/flink/conf/sql-client-defaults.yaml` by default. This file stores the default SQL-specific settings, such as catalog configurations and default table execution parameters. You can add content to this file through the safety valve exposed in Cloudera Manager:

Flink Client Advanced Configuration Snippet
(Safety Valve) for `flink-conf/sql-client-defaults.yaml`

You have the option to override some of these settings by creating a custom environment YAML file, and pass it when starting the SQL Client with the `-e sql-env.yaml` parameter. This is a convenient way of handling user-specific SQL Client configurations.

Configuring SQL Client for session mode

Every Flink SQL query is an independent Flink job. As with other Flink applications, you must decide on how you want to run them. The queries can run as standalone (per-job) YARN applications, as the default mode for all Flink jobs, or you can run them on a Flink session cluster.

While the per-job mode ensures better resource isolation and production characteristics, this mode requires more time to startup that can be inconvenient when exploring the SQL API. If you decide to stay in the per-job mode, you only need to start the SQL client without any additional parameters.

If you decide to run a Flink session cluster, perform the following steps.

1. Start a Flink session

You can start a Flink YARN session from the command line:

```
flink-yarn-session -tm 2048 -s 2 -d
```

This command starts a new session and specifies the Taskmanager size for future executors. You do not need to set the size of the cluster, because Flink does this automatically. The cluster starts without any TaskManager containers, and will grow and shrink in size as the number of deployed queries change.

2. Configure the SQL Client for session mode

To configure the SQL Client for the session mode, you need to create an environment YAML file (sql-env.yaml), and add the following configuration:

```
configuration:
  execution.target: yarn-session
```

This overrides the Flink configuration of the execution target to yarn-session mode.



Note:

This configuration works only if the Flink session cluster was started by the current user. For explicit control over the target session cluster, you must specify the `yarn.application.id` additional configuration parameter.

The SQL Client start command looks like this:

```
flink-sql-client embedded -e sql-env.yaml
```

SQL Client security

Before launching the SQL Client, you need to use the `kinit` command to have Kerberos authentication for the client. You also need to add the keytab files for starting the YARN session to use the SQL Client in a secured environment.

flink-sql-client

In Kerberos protected environments, you must use `kinit <principal>` before launching the `flink-sql-client`. The CLI does not support keytab based Kerberos authentication yet.

flink-yarn-session

YARN sessions that `flink-sql-client` is connected to should be started with Kerberos keytab and principal properties:

```
flink-yarn-session -tm 2096 -s 2 -d -nm "<application_name>" \
-D security.kerberos.login.principal=<username> \
-D security.kerberos.login.keytab=<keytab_name>
```



Note: In case you did not receive the keytab file from your administrator, you can use the following command to generate one:

```
> ktutil
ktutil: add_entry -password -p test -k 1 -e des3-cbc-sha1
Password for test@:
ktutil: wkt test.keytab
ktutil: quit
```