Cloudera Runtime 1.5.5

Starting Apache Hive

Date published: 2020-11-30
Date modified: 2025-06-06

https://docs.cloudera.com/

Legal Notice

Contents

Running a query in Cloudera Data Warehouse..... 4
Converting Hive CLI scripts to Beeline..... 4

Running a query in Cloudera Data Warehouse

You simply open Hue and submit your query. You do not need to manually start beeline or any other shell.

About this task

As a DWUser, you open Hue from a Virtual Warehouse that you set up, and run the query. The SQL engine reads from and writes to the same metastore, regardless of the type Virtual Warehouse.

Before you begin

- Required role: DW User
- You obtained permission to run SQL queries from the EnvironmentAdmin, who added you to a Hadoop SQL policy.

Procedure

1. Log into the Cloudera Data Warehouse service as DWUser.
2. Go to the **Virtual Warehouses** tab, locate the Virtual Warehouse using which you want to run queries, and click HUE.

   The Hue query editor opens in a new browser tab.
3. To run a query:
   a) Click a database to view the tables it contains.

      When you click a database, it sets it as the target of your query in the main query editor panel.
   b)

      Type a query in the editor panel and click  to run the query.

Converting Hive CLI scripts to Beeline

If you have legacy scripts that run Hive queries from edge nodes using the Hive CLI, you must solve potential incompatibilities with variable substitution in these scripts. Cloudera supports Beeline instead of Hive CLI. You can use Beeline to run legacy scripts with a few caveats.

About this task

In this task, you resolve incompatibilities in legacy Hive CLI scripts and Beeline:

- Configuration variables
  - Problem: You cannot refer to configuration parameters in scripts using the hiveconf namespace unless allowed.
  - Solution: You include the parameter in the HiveServer allowlist (whitelist).
- Namespace problems
  - Problem: Beeline does not support the system and env namespaces for variables.
  - Solution: You remove these namespace references from scripts using a conversion technique described in this task.

Procedure

1. Create a conversion script named env_to_hivevar.sh that removes env references in your SQL scripts.

```
#!/usr/bin/env bash
```

```
CMD_LINE=""

#Blank conversion of all env scoped values
for I in `env`; do
  CMD_LINE="$CMD_LINE --hivevar env:${I} "
done
echo ${CMD_LINE}
```

2. On the command line of a node in your cluster, define and export a variable named HIVEVAR, for example, and set it to run the conversion script.

```
export HIVEVAR=`./env_to_hivevar.sh`
```

3. Define and export variables to hold a few variables for testing the conversion.

```
export LOC_TIME_ZONE="US/EASTERN"
export MY_TEST_VAR="TODAY"
```

4. On the command line of a cluster node, test the conversion: Execute a command that references HIVEVAR to parse a SQL statement, remove the incompatible env namespace, and execute the remaining SQL.

```
hive ${HIVEVAR} -e 'select "${env:LOC_TIME_ZONE}";'
```

```
+-------------+
|     _c0     |
+-------------+
| US/EASTERN  |
+-------------+
```

5. Create a text file named init_var.sql to simulate a legacy script that sets two configuration parameters, one in the problematic env namespace.

```
set mylocal.test.var=hello;
set mylocal.test.env.var=${env:MY_TEST_VAR};
```

6. Include these configuration parameters in the allowlist: In Cloudera Manager, go to Clusters HIVE_ON_TEZ-1 Configuration , and search for hive-site.
7. In HiveServer2 Advanced Configuration Snippet (Safety Valve) for hive-site.xml, add the property key: hive.security.authorization.sqlstd.confwhitelist.append.
8. Provide the property value, or values, to allowlist, for example: mylocal\..*|junk.

   This action appends mylocal.test.var and mylocal.test.env.var parameters to the allowlist.
9. Save configuration changes, and restart any components as required.
10. Run a command that references HIVEVAR to parse a SQL script, removes the incompatible env namespace, and runs the remaining SQL, including the whitelisted configuration parameters identified by hiveconf:.

```
hive -i init_var.sql ${HIVEVAR} -e 'select "${hiveconf:mylocal.test.var}
","${hiveconf:mylocal.test.env.var}";'
```

```
+--------+--------+
|  _c0   |  _c1   |
+--------+--------+
| hello  | TODAY  |
+--------+--------+
```

Related Information

Custom Configuration (about Cloudera Manager Safety Valve)