Data Access 3

Starting Apache Hive

Date of Publish: 2019-12-17

https://docs.hortonworks.com

Contents

Start a Hive shell as the hive user

You can start the Hive shell, which uses Beeline in the background, to enter statements on the command line of a node in a cluster. After starting the Hive shell as the hive user, you can submit queries to Hive.

About this task

The architectural changes in Hive 3 support only Beeline for querying Hive from the command line. On an HDP cluster, you start a Hive shell. From the Hive shell, you can SQL statements. After connecting to Hive, you see a prompt that resembles the following example:

```
0: jdbc:hive2://myhiveserver.com>
```

The prompt consists of the following components:

- jdbc: The Java Database Connectivity protocol designator
- hive2: The HiveServer protocol designator in HDP 3 for using Hive 3
- myhiveserver.com: The fully-qualified domain name (FQDN) of the node that hosts HiveServer

Before you begin

- You added the Hive service on the cluster, using Ambari for example, and the following components are running:
  - HiveServer
  - Hive Metastore
  - A database for the metastore, such as the default MySQL Server
  - Hive clients
- In Ambari, Hive > Settings > Run as end user instead of Hive user is set to False.

Procedure

1. Start Hive using the FQDN of the HiveServer in your cluster to replace myhiveserver.com and the Database Username and Database Password password for the default hive user.
   beeline -u jdbc:hive2://myhiveserver.com:10000 -n hive -p

   In Ambari > Services > Hive > Configs > Database, you can configure the database user name and password. Property names in hive-site.xml for these settings are javax.jdo.option.ConnectionPassword and javax.jdo.option.ConnectionUserName.
2. Enter a password at the prompt.
3. Enter a query.

   ```
   SHOW DATABASES;
   ```

   Hive creates two databases when you add the Hive service to a cluster: information_schema and sys. All Metastore tables are mapped into your tablespace and are available in sys. The information_schema data reveals the state of the system, similar to sys database data, but in a user-friendly way. You can query information_schema using SQL standard queries, which are portable from one DBMS to another.

   Output is:

   ```
   +----------------------+
   |    database_name     |
   +----------------------+
   | default              |
   | information_schema   |
   | sys                  |
   +----------------------+
   ```

4. Create a table in the default database.

```
CREATE TABLE students (name VARCHAR(64), age INT, gpa DECIMAL(3,2));
```

5. Insert data into the table.

```
INSERT INTO TABLE students VALUES ('fred flintstone', 35, 1.28), ('barney
 rubble', 32, 2.32);
```

6. Exit the Beeline and Hive shells.

```
!quit
```

Start Hive as an end user

You can start Apache Hive as an end user authorized by Apache Ranger. As administrator, you must first set up the user in the operating system and in Ranger.

Before you begin

- Create a user. For example, add a Linux user using the useradd operating system command.
- Add the user to the Ranger list of users.
- In Ranger, add the user name to Hive policies that grant full access to Hive.
- In Ranger, add the user name to the HDFS all-path policy.
- In Ambari, in Hive > Settings under Security, set Run as end user instead of Hive user to True.

  This is equivalent to setting hive.server2.enable.doAs=true in hive-site.xml.
- In Ambari, if you get AccessControlException, add the following key and value to Hive > Configs > Advanced > Custom ranger-hive-security: xasecure.hive.describetable.showcolumns.authorization.optionand show-all

Procedure

1. Start Hive.
   beeline -u jdbc:hive2://myhiveserver.com:10000 -n user1 -p
2. Enter a password at the prompt.
3. Enter a query.

```
SHOW DATABASES;
```

4. Create a table in the default database.

```
CREATE TABLE students (name VARCHAR(64), age INT, gpa DECIMAL(3,2));
```

5. Insert data into the table.

```
INSERT INTO TABLE students VALUES ('fred flintstone', 35, 1.28), ('barney
 rubble', 32, 2.32);
```

Related Information

Configure a Resource-based Policy: Hive

Run a Hive command

You can run most Hive commands from the command line of a node in your cluster. The hive keyword, which launches Beeline in the background, precedes the Hive command.

About this task

Hive 3 and later support only Beeline for running Hive commands from the command line. In the task below, you start Beeline in the background and enter the -e flag followed by a Hive command that lists system variables.

Before you begin

The following components are running:

- HiveServer
- Hive clients
- You have a network connection to Beeline.

Procedure

1. On the command line of a a node in your cluster, enter the hive command to send configuration properties to standard output.

   ```
   > hive -e set
   ```

   Supported commands appear. All obsolete Hive CLI commands are supported on the Beeline command line except set key=value commands that configure Hive metastore.
2. At the prompt, enter your user name and password for accessing Hive.
   The output includes the system variable settings:

   ```
   +-------------------------------------------------------------+
   |                             set                             |
   +-------------------------------------------------------------+
   | _hive.hdfs.session.path=/tmp/hive/hive/91ecb...00a          |
   | _hive.local.session.path=/tmp/hive/91ecb...00a              |
   | _hive.tmp_table_space=/tmp/hive/hive/91ecb...00a/_tmp_space.db |
   | ambari.hive.db.schema.name=hive                             |
   | atlas.hook.hive.maxThreads=1                                |
    ...
   ```

Convert Hive CLI scripts to Beeline

If you have legacy scripts that execute Hive queries from edge nodes using the Hive CLI, you must solve potential incompatibilities with variable substitution in these scripts. HDP 3.0 and later 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 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 execute 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. Whitelist these configuration parameters: In Ambari, go to Hive > Configs > Advanced > Custom hiveserver2-site.
7. Add the property key: hive.security.authorization.sqlstd.confwhitelist.append.
8. Provide the property value, or values, to whitelist, for example: mylocal\..*|junk.

   This action appends mylocal.test.var and mylocal.test.env.var parameters to the whitelist.
9. Save configuration changes, and restart any components as required.

10. Execute a command that references HIVEVAR to parse a SQL script, removes the incompatible env namespace, and executes 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

Apache Wiki: Language Manual Variable Substitution