Cloudera Runtime 7.3.3

Starting Apache Hive

Date published: 2019-08-21
Date modified: 2026-11-13



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

Starting Hive on an insecure cluster....4
Starting Hive using a password....4
Running a Hive command in Cloudera Data Hub....8
Running a query in Cloudera Data Warehouse....8
Converting Hive CLI scripts to Beeline....9
Configuring graceful shutdown property for HiveServer....11

Starting Hive on an insecure cluster

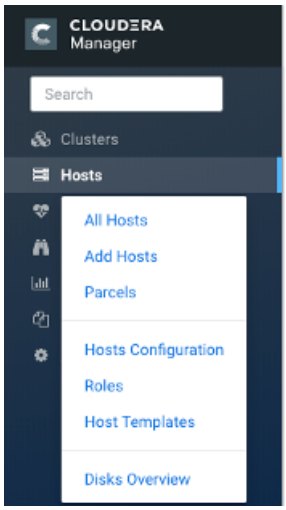
If you want to use Apache Hive for a quick test, you can do so using default Hive default authorization mode, assuming you are on an insecure cluster (no Kerberos or Ranger policies). In default authorization mode, only user hive can access Hive. Steps to start the Hive shell, not to be mistaken with the Hive CLI that Cloudera does not support, include how to log into a cluster.

About this task

From the cluster command line, you type hive on the command line of your cluster to start the Hive shell. In the background, Beeline launches the Hive shell.

Procedure

1. In Cloudera Manager, click Hosts All Hosts .



2. Make a note of the IP address or host name of a node in your cluster, for example myhost-vpc.cloudera.com.
3. Use ssh to log into the cluster.
   For example:

```
ssh myhost-vpc.cloudera.com
```

4. Type hive to start Hive from the command line.
5. Enter Hive queries.

```
SHOW DATABASES;
```

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

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

Starting Hive using a password

You start the Hive shell using a Beeline command to query Hive as an end user authorized by Apache Ranger. As administrator, you set up the end user in the operating system and in Ranger.

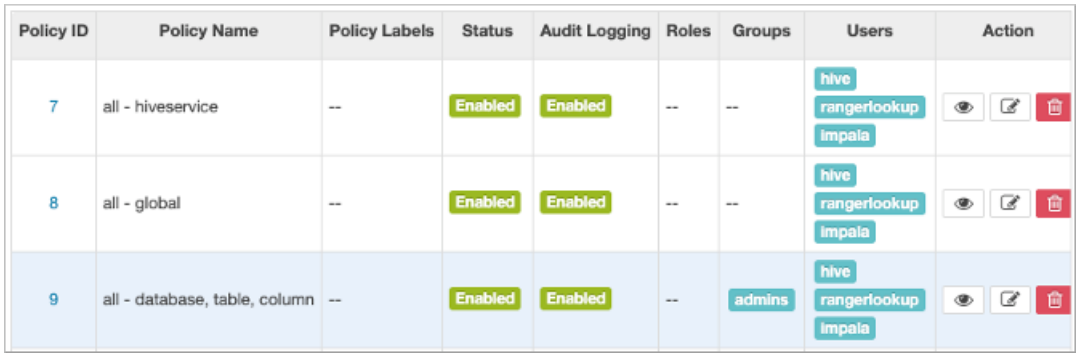
About this task

Before starting Hive for the first time, check that you are covered by Ranger policies required for basic operations as shown in the following steps. All users need to use the default database, perform operations such as listing database names, and query the information schema. The preloaded default database tables columns and information_schema database Ranger policies cover group public (all users). If these policies are disabled, you cannot use the default database, perform basic operations such as listing database names, or query the information schema. For example, if the default database tables columns policy is disabled, the following error appears if you try to use the default database:

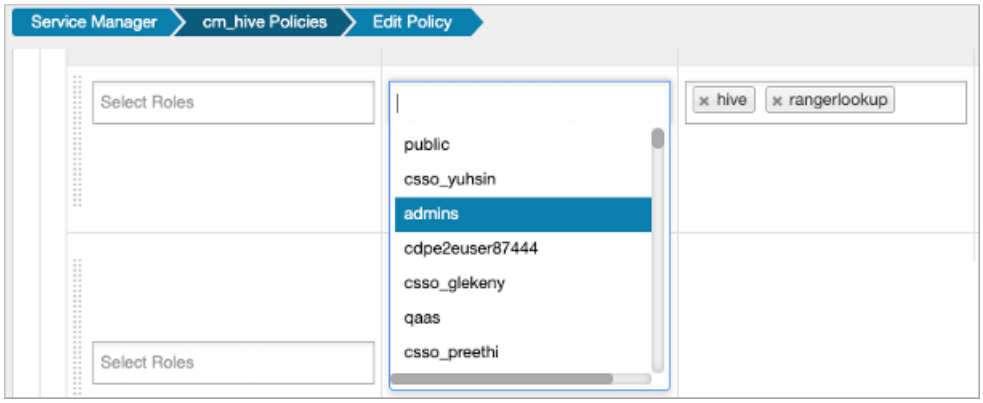
```
hive> USE default;
Error: Error while compiling statement: FAILED: HiveAccessControlException
Permission denied: user [hive] does not have [USE] privilege on [default]
```

Procedure

1. Access the Ranger Console: click the Ranger Admin web UI link, enter your user name and password, then click Sign In.
2. On the far right, click Ranger Hadoop SQL , and in Allow Conditions, edit all - database, table, column.



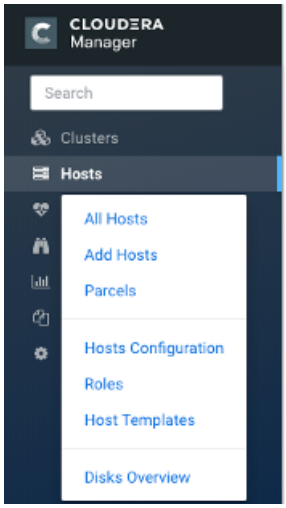
3. Add your user or group name to Hive policies to grant full access to Hive.
   For example, add the admins group name to the list of groups that can access Hive.



4. Check that the preloaded default database tables columns and information_schema database policies are enabled for group public.

Policy ID	Policy Name	Policy Labels	Status	Audit Logging	Roles	Groups	Users
7	all - global	--	Enabled	Enabled	--	--	hive beacon dpprofiler hue + More..
8	all - database, table, column	--	Enabled	Enabled	--	--	hive beacon dpprofiler hue + More..
9	all - database, table	--	Enabled	Enabled	--	--	hive beacon dpprofiler hue + More..
10	all - database	--	Enabled	Enabled	--	public	hive beacon dpprofiler hue + More..
11	all - hiveservice	--	Enabled	Enabled	--	--	hive beacon dpprofiler hue + More..
12	all - database, udf	--	Enabled	Enabled	--	--	hive beacon dpprofiler hue + More..
13	all - url	--	Enabled	Enabled	--	--	hive beacon dpprofiler hue + More..
14	default database tables columns	--	Enabled	Enabled	--	public	--
15	Information_schema database ...	--	Enabled	Enabled	--	public	--

5. In Environments *YOUR ENVIRONMENT YOUR CLUSTER* , click the CM-URL.
6. In Cloudera Manager, click Hosts All Hosts .



7. Make a note of the IP address or host name of a node in your cluster, for example myhost-vpc.cloudera.com.

Tip: In a Kerberized cluster, ensure the core-site.xml file is correctly configured on all nodes from which you plan to connect to beeline. Include the following property in the core-site.xml:

```
<property>
  <name>hadoop.security.authentication</name>
  <value>kerberos</value>
</property>
```

8. Use ssh to log into the cluster.
   For example:

```
ssh myhost-vpc.cloudera.com
```

You can get help about starting the Hive shell. On the command line, type

```
hive -h
```

Output is:

```
Connect using simple authentication to HiveServer2 on localhost:10000
beeline -u jdbc:hive2://localhost:10000 username password

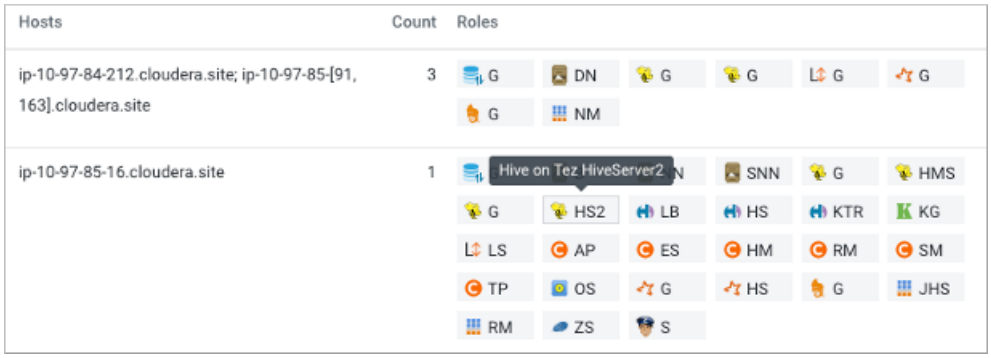
Connect using simple authentication to HiveServer2 on hs.local:10000 us
ing -n for username and -p for password
beeline -n username -p password -u jdbc:hive2://hs2.local:10012

Connect using Kerberos authentication with hive/localhost@mydomain.com as
 HiveServer2 principal
beeline -u "jdbc:hive2://hs2.local:10013/default;principal=hive/localho
st@mydomain.com"

Connect using SSL connection to HiveServer2 on localhost at 10000
beeline "jdbc:hive2://localhost:10000/default;ssl=true;sslTrustStore=/usr/
local/truststore;trustStorePassword=mytruststorepassword"

Connect using LDAP authentication
beeline -u jdbc:hive2://hs2.local:10013/default <ldap-username> <ldap-pas
sword>
```

9. Using the fully qualified domain name or IP address a node in your cluster from Cloudera Manager Hosts Role(s) and looking through the list of roles to find HS2 (Hive on Tez HiveServer2). For example:



This node has the HiveServer role, so you can use the name or IP address in Beeline.

10. Start the Hive shell.

- Use your user name if your cluster security is set up.
- Use the user name hive and no password.

Substitute the name or IP address of your HiveServer host for 10.65.13.98.

Simple authentiation:

```
beeline -u jdbc:hive2://10.65.13.98:10000 -n <your user name> -p
```

Kerberos:

```
beeline -u "jdbc:hive2://10.65.13.98:10000/default;principal=hive/_HOST@
CLOUDERA.SITE"
```

11. Enter Hive queries.

```
SHOW DATABASES;
```

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

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

Related Information

Configure a Resource-based Policy: Hive

Running a Hive command in Cloudera Data Hub

You can run most Hive commands that push configuration variables to Hive SQL scripts from the command line of a node in your cluster. You use a keyword and options to launch these commands in Beeline.

About this task

Hive supports running Hive commands from the command line. The command you enter launches Beeline in the background. The -e flag followed by a Hive set command lists system variables.

Procedure

On the command line of a node in your Cloudera 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 by Beeline except set key=value commands that configure Hive Metastore.

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                  |
                        |
 ...
```

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

Convert Hive CLI legacy scripts to Beeline by resolving variable substitution incompatibilities with hiveconf allowlists and namespace conversion techniques.

About this task

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.

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)

Example of using the Cloudera Manager Safety Valve

Configuring graceful shutdown property for HiveServer

Learn how to configure the graceful shutdown timeout property for HiveServer (HS2), which ensures that HS2 waits for a specified time period before shutting down, thereby allowing queries that are already running to complete before HS2 stops.

About this task

By default, the graceful shutdown timeout property is set to 5 minutes. Queries that are still running beyond the timeout period are aborted and will have to be manually submitted by the user because HS2 cannot automatically resubmit these queries.

Important: If you want to disable HS2 graceful shutdown, set the graceful timeout property value to 0.

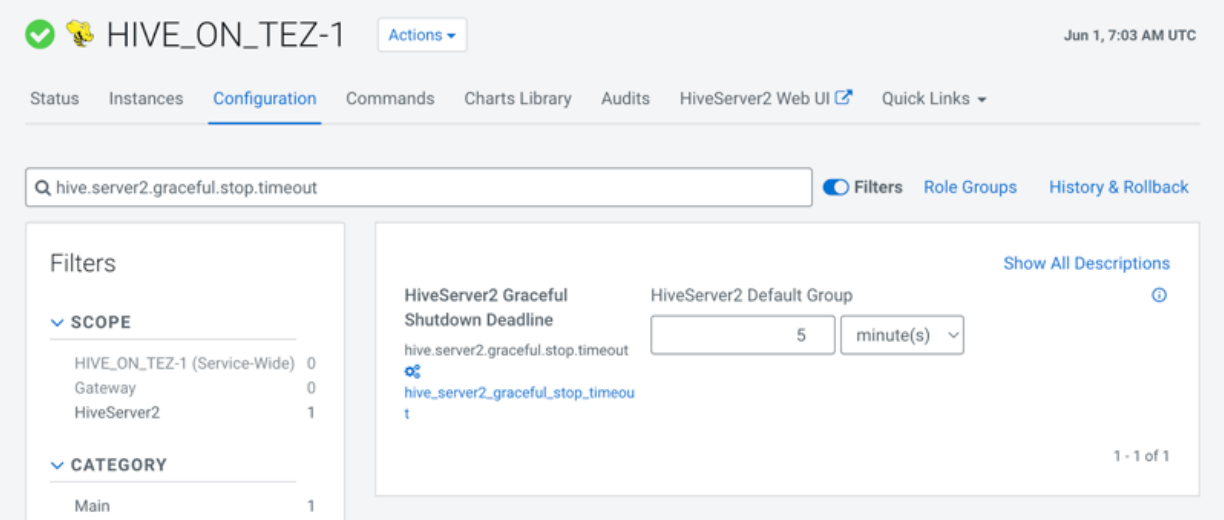
You must also know that submitting new queries or starting a new session during a HS2 graceful shutdown will fail. Also, if there are no active queries during the graceful shutdown, HS2 stops in 30 seconds instead of waiting 5 minutes for all the live sessions to close.

You can modify the graceful shutdown timeout property based on a customer's usage, for example, you can choose to increase the timeout property if the customer has long running queries.

The HS2 graceful shutdown is initiated during a Cloudera Runtime Rolling Upgrade, Rolling Restart, can be triggered by stopping HS2 in Cloudera Manager, or by running the hive --service hiveserver2 --graceful_stop command.

Procedure

1. In Cloudera Manager, click Clusters and select the Hive on Tez service.
2. From the Hive on Tez service, go to the **Configuration** tab and search for hive.server2.graceful.stop.timeout.



3. Modify the graceful timeout value. By default, the timeout value is set to 5 minutes.
4. Click Save Changes and restart the Hive on Tez service.

Results

The changes do not take effect during the first HS2 restart and the graceful timeout value that was specified prior to the change is considered for the graceful shutdown. The new changes take effect only from the second restart.

For example, if the timeout value was changed from 5 minutes to 10 minutes, the first HS2 restart considers the 5 minute timeout for the graceful shutdown. The second HS2 restart will consider the 10 minute timeout value.