

## Starting Apache Hive

Date published: 2019-08-21

Date modified: 2021-09-08



# 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

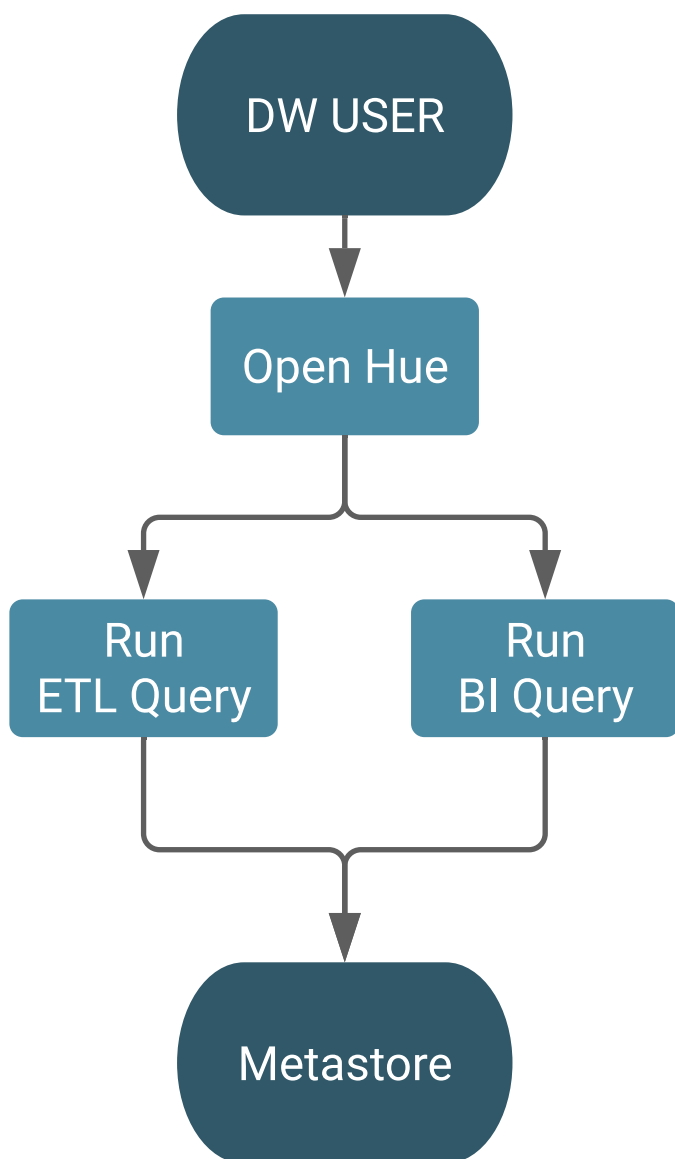
<b>Running a query in Cloudera Data Warehouse.....</b>	<b>4</b>
<b>Converting Hive CLI scripts to Beeline.....</b>	<b>6</b>

## 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 DW User, you open Hue from a Virtual Warehouse that you set up for ETL or BI jobs, and run the query. The SQL engine reads from and writes to the same metastore, regardless of the type Virtual Warehouse, as shown in the following diagram.




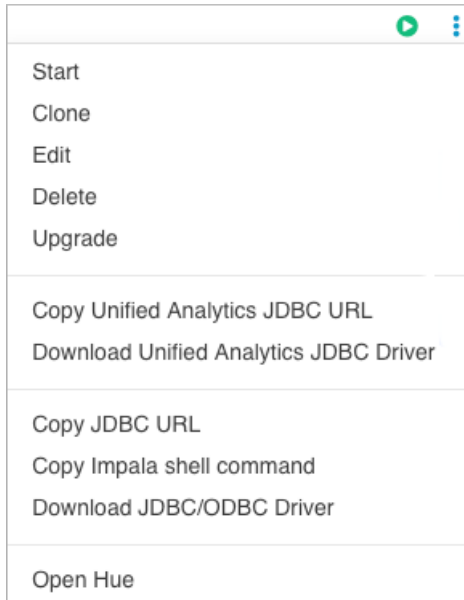
### Before you begin

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

## Procedure

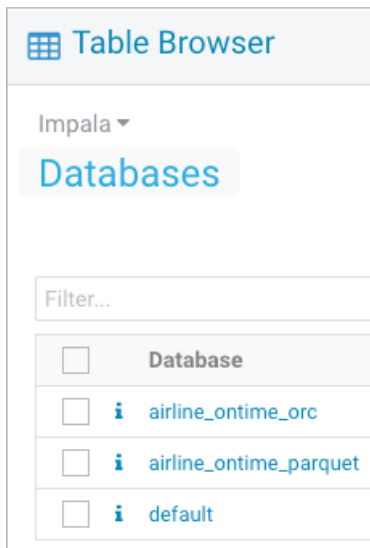
1.

On the **Overview** page under Virtual Warehouses, click options , and select Open Hue.




2. Select a database.

For example, select database `airline_ontime_parquet`.



3. In Hue, enter a query.

```
SELECT dest, origin
FROM flights
GROUP BY dest, origin;
```

4. 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. CDP 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 stat
  - 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 run 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 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

[Custom Configuration \(about Cloudera Manager Safety Valve\)](#)

[Example of using the Cloudera Manager Safety Valve](#)