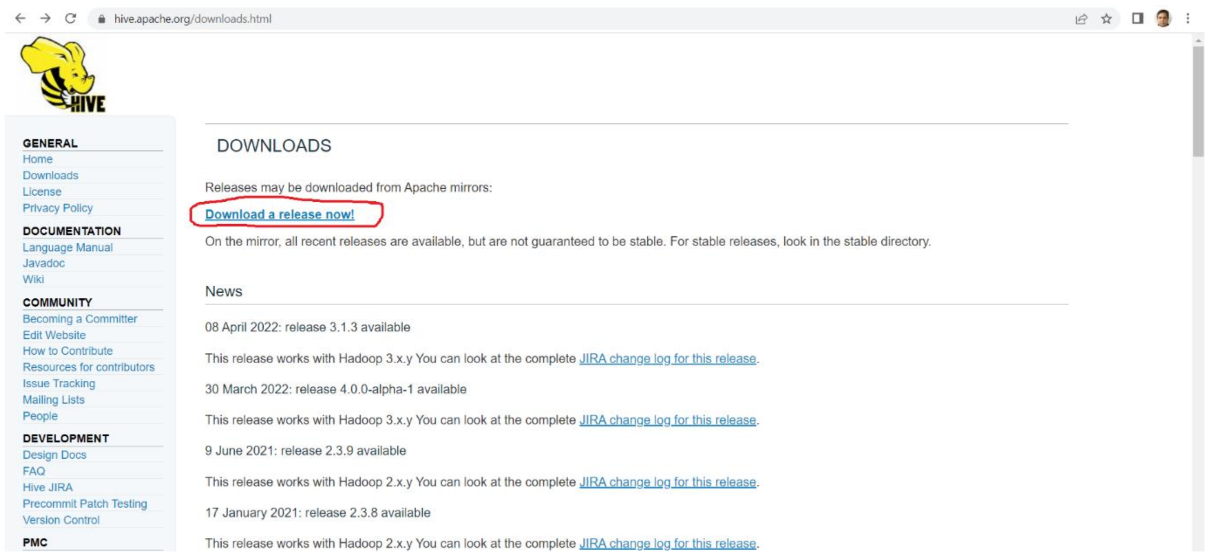
Tutorial 10
Apache Hive

Step 1: Download and Untar Hive

Visit the Apache Hive official download page and determine which Hive version is best suited for your Hadoop edition. Once you establish which version you need, select the Download a Release Now! option.


hive.apache.org/downloads.html

GENERAL
Home
Downloads
License
Privacy Policy
DOCUMENTATION
Language Manual
Javadoc
Wiki
COMMUNITY
Becoming a Committer
Edit Website
How to Contribute
Resources for contributors
Issue Tracking
Mailing Lists
People
DEVELOPMENT
Design Docs
FAQ
Hive JIRA
Precommit Patch Testing
Version Control
PMC

DOWNLOADS

Releases may be downloaded from Apache mirrors:

Download a release now!

On the mirror, all recent releases are available, but are not guaranteed to be stable. For stable releases, look in the stable directory.

News

08 April 2022: release 3.1.3 available

This release works with Hadoop 3.x.y You can look at the complete JIRA change log for this release.

30 March 2022: release 4.0.0-alpha-1 available

This release works with Hadoop 3.x.y You can look at the complete JIRA change log for this release.

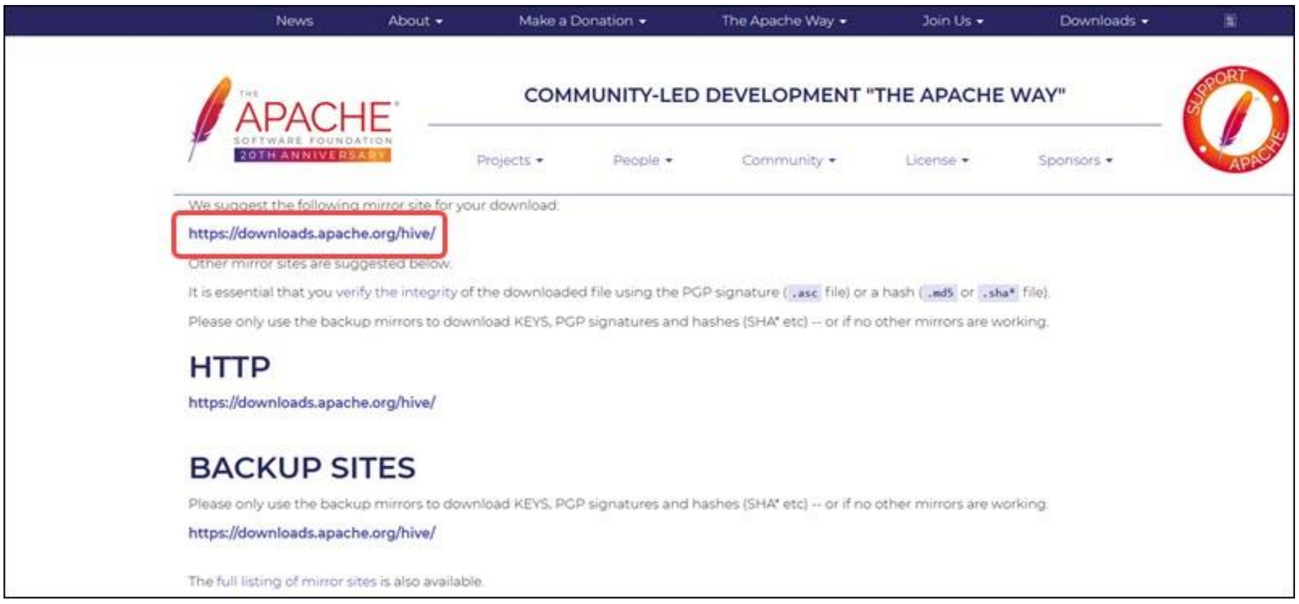
9 June 2021: release 2.3.9 available

This release works with Hadoop 2.x.y You can look at the complete JIRA change log for this release.

17 January 2021: release 2.3.8 available

This release works with Hadoop 2.x.y You can look at the complete JIRA change log for this release.

The mirror link on the subsequent page leads to the directories containing available Hive tar packages. This page also provides useful instructions on how to validate the integrity of files retrieved from mirror sites.


News About ▾ Make a Donation ▾ The Apache Way ▾ Join Us ▾ Downloads ▾

COMMUNITY-LED DEVELOPMENT "THE APACHE WAY"

Projects ▾ People ▾ Community ▾ License ▾ Sponsors ▾

We suggest the following mirror site for your download:

https://downloads.apache.org/hive/

Other mirror sites are suggested below.

It is essential that you verify the integrity of the downloaded file using the PGP signature (.asc file) or a hash (.md5 or .sha* file).

Please only use the backup mirrors to download KEYS, PGP signatures and hashes (SHA* etc) -- or if no other mirrors are working.

HTTP

https://downloads.apache.org/hive/

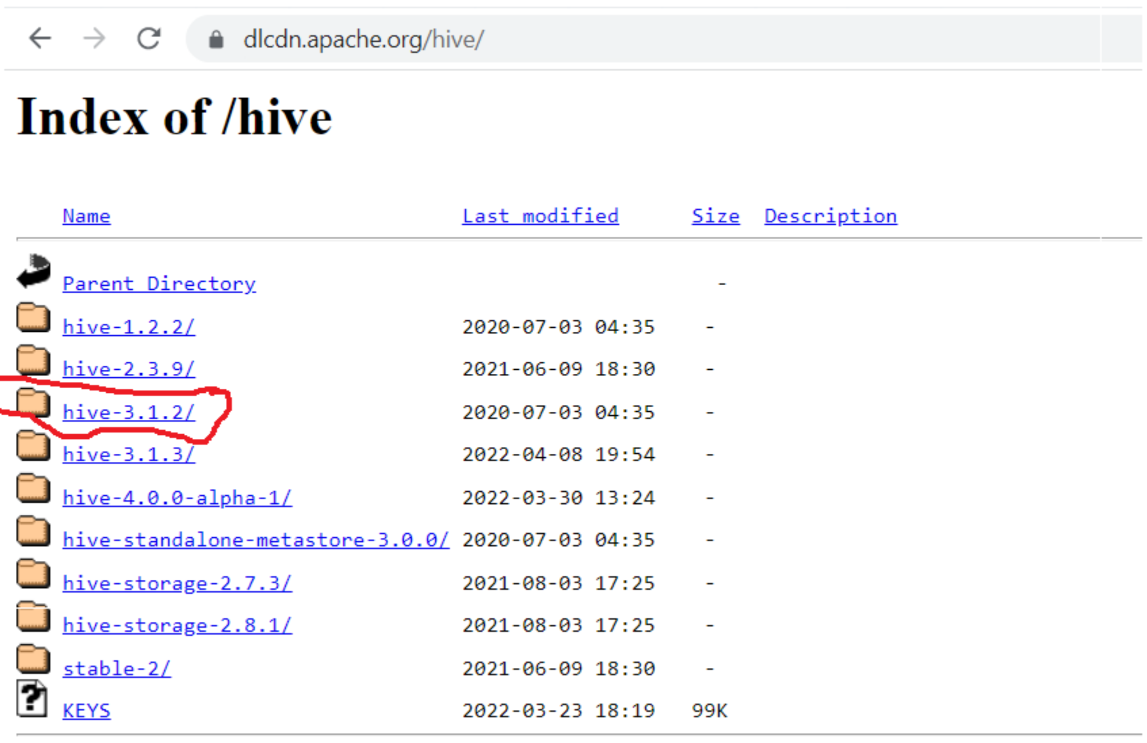
BACKUP SITES

Please only use the backup mirrors to download KEYS, PGP signatures and hashes (SHA* etc) -- or if no other mirrors are working.

https://downloads.apache.org/hive/

The full listing of mirror sites is also available.

The Ubuntu system presented in this guide already has Hadoop 3.2.4 installed. This Hadoop version is compatible with the Hive 3.1.2 release.

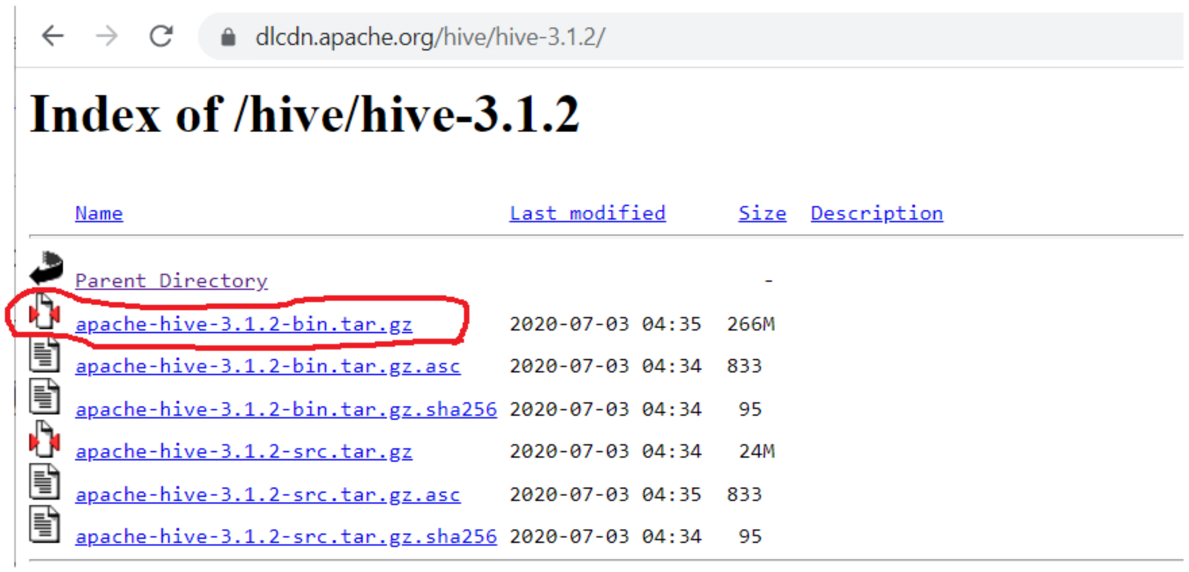


dlcdn.apache.org/hive/

Index of /hive

Name	Last modified	Size	Description
Parent Directory		-	
hive-1.2.2/	2020-07-03 04:35	-	
hive-2.3.9/	2021-06-09 18:30	-	
hive-3.1.2/	2020-07-03 04:35	-	
hive-3.1.3/	2022-04-08 19:54	-	
hive-4.0.0-alpha-1/	2022-03-30 13:24	-	
hive-standalone-metastore-3.0.0/	2020-07-03 04:35	-	
hive-storage-2.7.3/	2021-08-03 17:25	-	
hive-storage-2.8.1/	2021-08-03 17:25	-	
stable-2/	2021-06-09 18:30	-	
KEYS	2022-03-23 18:19	99K	

Select the apache-hive-3.1.2-bin.tar.gz file to begin the download process.



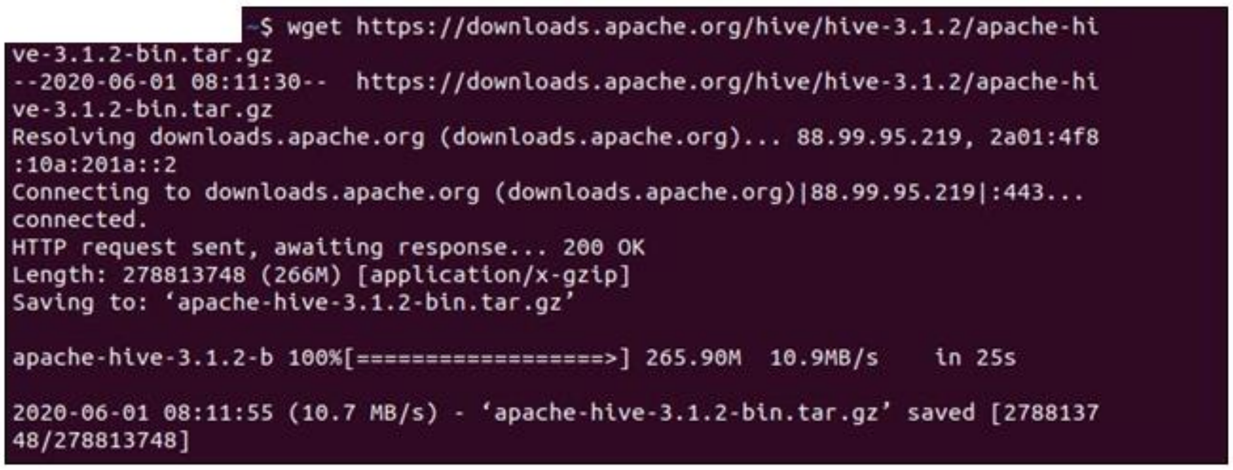
dlcdn.apache.org/hive/hive-3.1.2/

Index of /hive/hive-3.1.2

Name	Last modified	Size	Description
Parent Directory		-	
apache-hive-3.1.2-bin.tar.gz	2020-07-03 04:35	266M	
apache-hive-3.1.2-bin.tar.gz.asc	2020-07-03 04:34	833	
apache-hive-3.1.2-bin.tar.gz.sha256	2020-07-03 04:34	95	
apache-hive-3.1.2-src.tar.gz	2020-07-03 04:34	24M	
apache-hive-3.1.2-src.tar.gz.asc	2020-07-03 04:35	833	
apache-hive-3.1.2-src.tar.gz.sha256	2020-07-03 04:34	95	

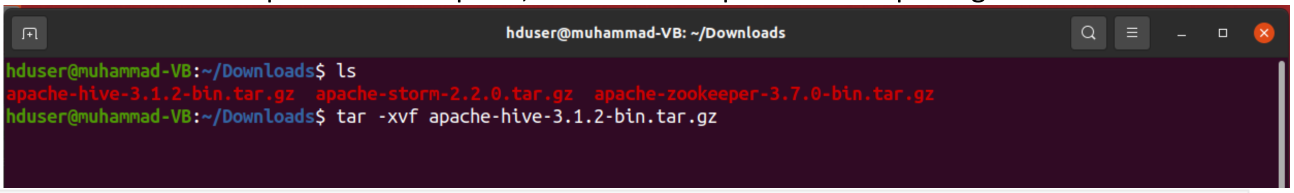
Alternatively, access your Ubuntu command line and download the compressed Hive files using and the wget command followed by the download path:

```
$wget https://downloads.apache.org/hive/hive-3.1.2/apache-hive-3.1.2-bin.tar.gz
```



```
~$ wget https://downloads.apache.org/hive/hive-3.1.2/apache-hi
ve-3.1.2-bin.tar.gz
--2020-06-01 08:11:30--  https://downloads.apache.org/hive/hive-3.1.2/apache-hi
ve-3.1.2-bin.tar.gz
Resolving downloads.apache.org (downloads.apache.org)... 88.99.95.219, 2a01:4f8
:10a:201a::2
Connecting to downloads.apache.org (downloads.apache.org)|88.99.95.219|:443...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 278813748 (266M) [application/x-gzip]
Saving to: 'apache-hive-3.1.2-bin.tar.gz'

apache-hive-3.1.2-b 100%[=================>] 265.90M  10.9MB/s    in 25s

2020-06-01 08:11:55 (10.7 MB/s) - 'apache-hive-3.1.2-bin.tar.gz' saved [2788137
48/278813748]
```

Once the download process is complete, untar the compressed Hive package:



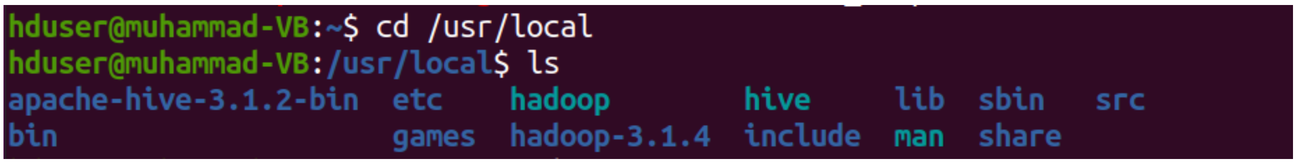
```
hduser@muhammad-VB: ~/Downloads
hduser@muhammad-VB:~/Downloads$ ls
apache-hive-3.1.2-bin.tar.gz  apache-storm-2.2.0.tar.gz  apache-zookeeper-3.7.0-bin.tar.gz
hduser@muhammad-VB:~/Downloads$ tar -xvf apache-hive-3.1.2-bin.tar.gz
```

```
$tar -xvf apache-hive-3.1.2-bin.tar.gz
```

Move unzip folder to /usr/local/ folder

```
$sudo mv ./apache-hive-3.1.2-bin /usr/local
```

The Hive binary files are now located in the apache-hive-3.1.2-bin directory.



```
hduser@muhammad-VB:~$ cd /usr/local
hduser@muhammad-VB:/usr/local$ ls
apache-hive-3.1.2-bin  etc    hadoop        hive     lib  sbin   src
bin                    games  hadoop-3.1.4  include  man  share
```

```
$cd /usr/local
$sudo ln -sf ./apache-hive-3.1.2-bin ./hive
$sudo chown -R hduser:hadoopgroup ./hive
```

Step 2: Configure Hive Environment Variables (bashrc)

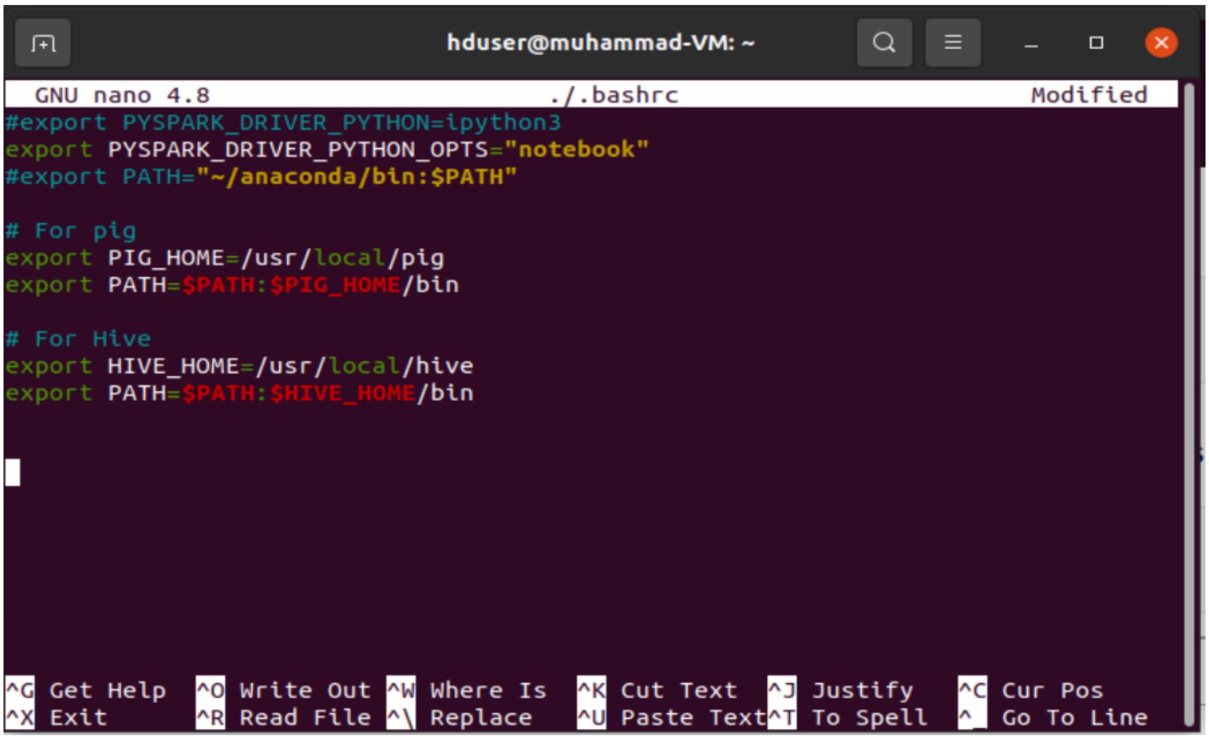
The $HIVE_HOME environment variable needs to direct the client shell to the apache-hive-3.1.2-bin directory. Edit the .bashrc shell configuration file using a text editor of your choice (we will be using nano):

```
$cd /home/hduser
$sudo nano ./.bashrc
```

Append the following Hive environment variables to the ./.bashrc file:

```
export HIVE_HOME=/usr/local/hive
export PATH=$PATH:$HIVE_HOME/bin
```

The Hadoop environment variables are located within the same file.



```
hduser@muhammad-VM: ~
  GNU nano 4.8                    ./.bashrc                        Modified
#export PYSPARK_DRIVER_PYTHON=ipython3
export PYSPARK_DRIVER_PYTHON_OPTS="notebook"
#export PATH="~/anaconda/bin:$PATH"

# For pig
export PIG_HOME=/usr/local/pig
export PATH=$PATH:$PIG_HOME/bin

# For Hive
export HIVE_HOME=/usr/local/hive
export PATH=$PATH:$HIVE_HOME/bin


^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Paste Text^T To Spell  ^_ Go To Line
```

Save and exit the .bashrc file once you add the Hive variables. Apply the changes to the current environment with the following command:

```
$source ./.bashrc
```

Step 3: Edit hive-config.sh file

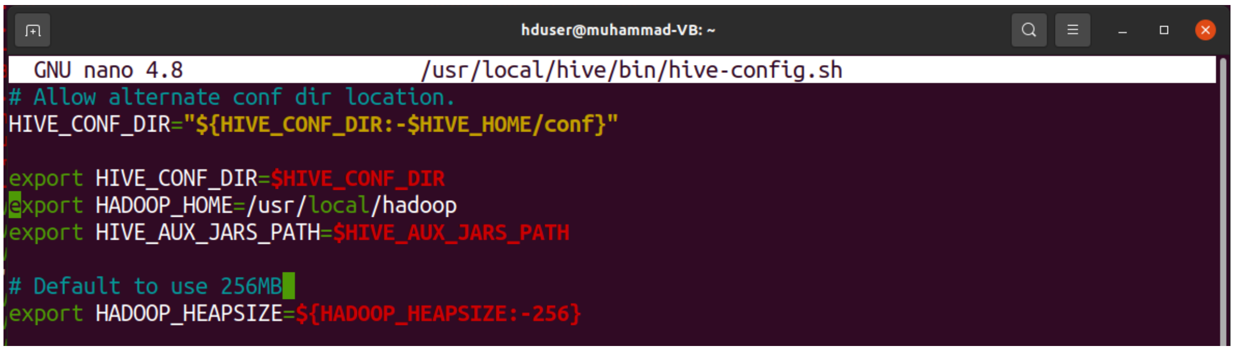
Apache Hive needs to be able to interact with the Hadoop Distributed File System. Access the hive-config.sh file using the previously created $HIVE_HOME variable:

```
$sudo nano /usr/local/hive/bin/hive-config.sh
```

Note: The hive-config.sh file is in the bin directory within your Hive installation directory.

Add the **`HADOOP_HOME`** variable and the full path to your Hadoop directory:

```
export HADOOP_HOME=/usr/local/hadoop
```



```
  GNU nano 4.8                  /usr/local/hive/bin/hive-config.sh
# Allow alternate conf dir location.
HIVE_CONF_DIR="${HIVE_CONF_DIR:-$HIVE_HOME/conf}"

export HIVE_CONF_DIR=$HIVE_CONF_DIR
export HADOOP_HOME=/usr/local/hadoop
export HIVE_AUX_JARS_PATH=$HIVE_AUX_JARS_PATH

# Default to use 256MB
export HADOOP_HEAPSIZE=${HADOOP_HEAPSIZE:-256}
```

Save the edits and exit the hive-config.sh file.

Step 4: Create Hive Directories in HDFS

Create two separate directories to store data in the HDFS layer:

- The temporary, tmp directory is going to store the intermediate results of Hive processes.
- The warehouse directory is going to store the Hive related tables.
- Create tmp Directory

First delete the directory (tmp) on hadoop and then create a tmp directory within the HDFS storage layer. This directory is going to store the intermediary data Hive sends to the HDFS:

```
$hadoop fs -rm -r /tmp
$hadoop fs -mkdir /tmp
```

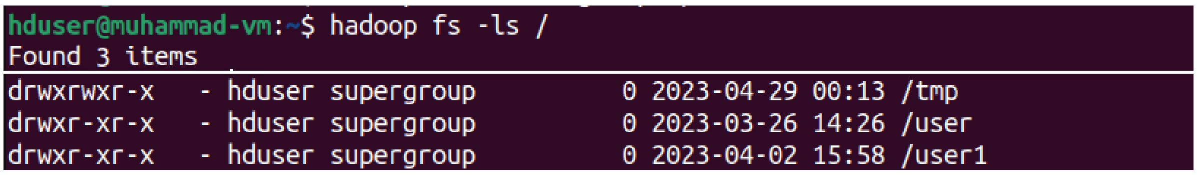
Add write and execute permissions to tmp group members:

```
$hadoop fs -chmod g+w /tmp
```

Check if the permissions were added correctly:

```
$hadoop fs -ls /
```

The output confirms that users now have write and execute permissions.



```
hduser@muhammad-vm:~$ hadoop fs -ls /
Found 3 items
drwxrwxr-x   - hduser supergroup          0 2023-04-29 00:13 /tmp
drwxr-xr-x   - hduser supergroup          0 2023-03-26 14:26 /user
drwxr-xr-x   - hduser supergroup          0 2023-04-02 15:58 /user1
```

Create warehouse Directory

Create the *warehouse* directory within the */user/hive/* parent directory:

```
$hadoop fs -mkdir -p /user/hive/warehouse
```

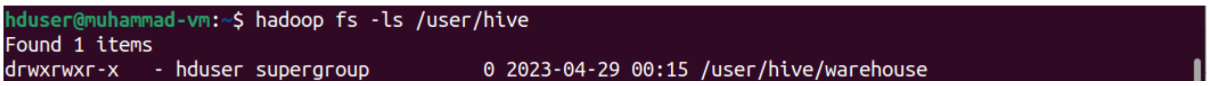
Add **write** and **execute** permissions to *warehouse* group members:

```
$hadoop fs -chmod g+w /user/hive/warehouse
```

Check if the permissions were added correctly:

```
$hadoop fs -ls /user/hive
```

The output confirms that users now have write and execute permissions.



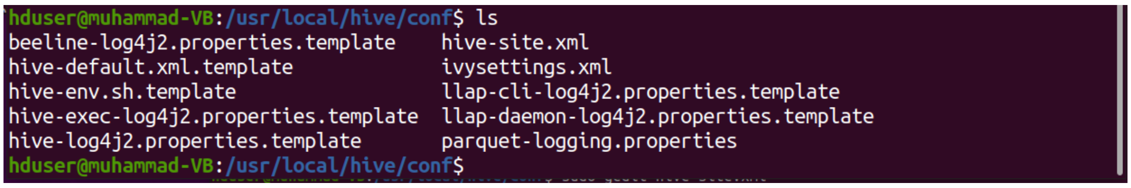
```
hduser@muhammad-vm:~$ hadoop fs -ls /user/hive
Found 1 items
drwxrwxr-x   - hduser supergroup          0 2023-04-29 00:15 /user/hive/warehouse
```

Step 5: Configure hive-site.xml File

Apache Hive distributions contain template configuration files by default. The template files are located within the Hive conf directory and outline default Hive settings.
Use the following command to locate the correct file:

```
$cd /usr/local/hive/conf
```

List the files contained in the folder using the ls command.



```
hduser@muhammad-VB:/usr/local/hive/conf$ ls
beeline-log4j2.properties.template    hive-site.xml
hive-default.xml.template             ivysettings.xml
hive-env.sh.template                  llap-cli-log4j2.properties.template
hive-exec-log4j2.properties.template  llap-daemon-log4j2.properties.template
hive-log4j2.properties.template       parquet-logging.properties
hduser@muhammad-VB:/usr/local/hive/conf$
```

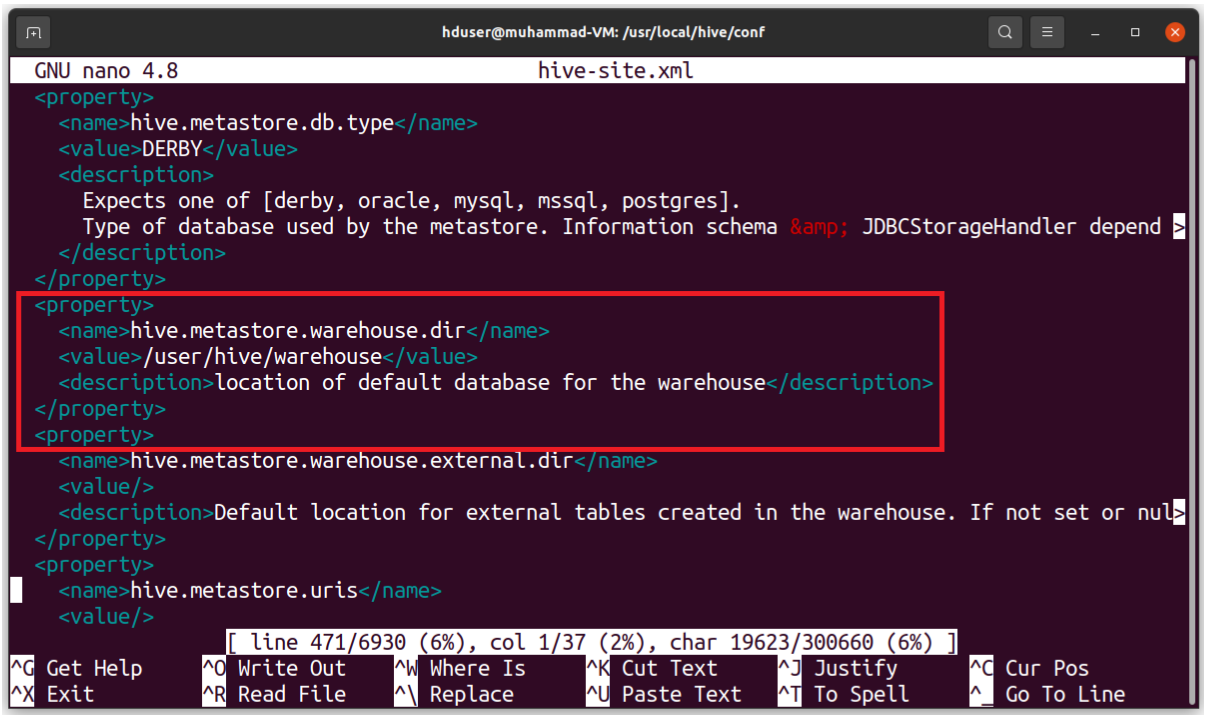
Use the hive-default.xml.template to create the hive-site.xml file:

```
$cp hive-default.xml.template hive-site.xml
```

Access the hive-site.xml file using the nano text editor:

```
$sudo nano hive-site.xml
```

Using Hive in a stand-alone mode rather than in a real-life Apache Hadoop cluster is a safe option for newcomers. You can configure the system to use your local storage rather than the HDFS layer by setting the hive.metastore.warehouse.dir parameter value to the location of your Hive warehouse directory at line 471 (**Line number display on nano editor can be switched on using Alt + c to find the line number**). No action required after opening this file and you just need to verify the presence of code on the below mentioned screenshot and close the file without anything to save.

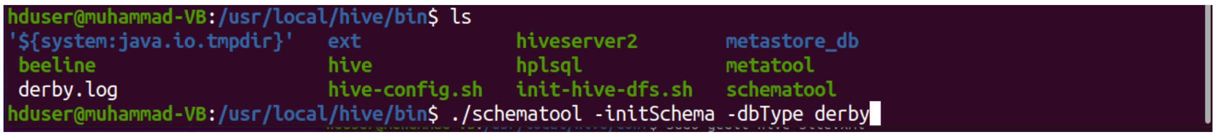


```
hduser@muhammad-VM: /usr/local/hive/conf
  GNU nano 4.8                          hive-site.xml
  <property>
    <name>hive.metastore.db.type</name>
    <value>DERBY</value>
    <description>
      Expects one of [derby, oracle, mysql, mssql, postgres].
      Type of database used by the metastore. Information schema &amp; JDBCStorageHandler depend >
    </description>
  </property>
  <property>
    <name>hive.metastore.warehouse.dir</name>
    <value>/user/hive/warehouse</value>
    <description>location of default database for the warehouse</description>
  </property>
  <property>
    <name>hive.metastore.warehouse.external.dir</name>
    <value/>
    <description>Default location for external tables created in the warehouse. If not set or nul>
  </property>
  <property>
    <name>hive.metastore.uris</name>
    <value/>
                  [ line 471/6930 (6%), col 1/37 (2%), char 19623/300660 (6%) ]
^G Get Help     ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos
^X Exit         ^R Read File    ^\ Replace      ^U Paste Text   ^T To Spell     ^_ Go To Line
```

Step 6: Initiate Derby Database

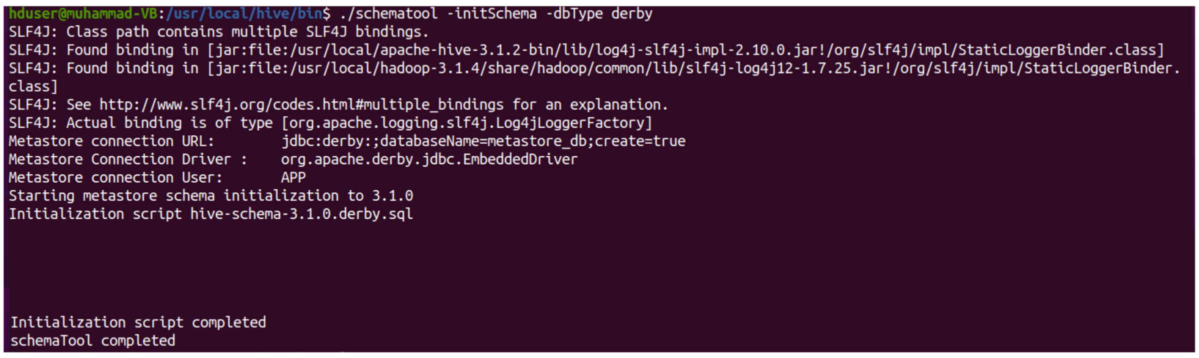
Apache Hive uses the Derby database to store metadata. Initiate the Derby database, from the Hive bin directory using the schematool command:

```
$cd /usr/local/hive/bin
$./schematool -initSchema -dbType derby
```



```
hduser@muhammad-VB:/usr/local/hive/bin$ ls
'${system:java.io.tmpdir}'   ext            hiveserver2      metastore_db
 beeline                     hive           hplsql           metatool
 derby.log                   hive-config.sh init-hive-dfs.sh schematool
hduser@muhammad-VB:/usr/local/hive/bin$ ./schematool -initSchema -dbType derby
```

The process can take a few moments to complete. If you face some error during this step, check the next step to fix this error.



```
hduser@muhammad-VB:/usr/local/hive/bin$ ./schematool -initSchema -dbType derby
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/apache-hive-3.1.2-bin/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-3.1.4/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.
class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Metastore connection URL:        jdbc:derby:;databaseName=metastore_db;create=true
Metastore Connection Driver :    org.apache.derby.jdbc.EmbeddedDriver
Metastore connection User:       APP
Starting metastore schema initialization to 3.1.0
Initialization script hive-schema-3.1.0.derby.sql



Initialization script completed
schemaTool completed
```

Derby is the default metadata store for Hive. If you plan to use a different database solution, such as MySQL or PostgreSQL, you can specify a database type in the hive-site.xml file.

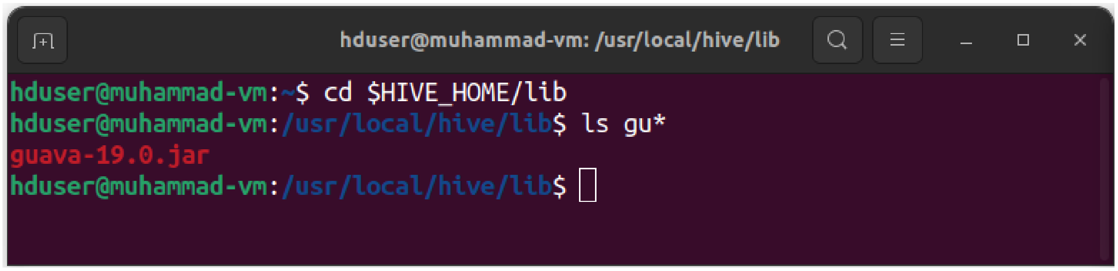
How to Fix guava Incompatibility Error in Hive

If the Derby database does not successfully initiate, you might receive an error with the following content:
“Exception in thread “main” java.lang.NoSuchMethodError: com.google.common.base.Preconditions.checkArgument(ZLjava/lang/String;Ljava/lang/Object;)V”

This error indicates that there is most likely an incompatibility issue between Hadoop and Hive guava versions.

Open a new terminal on Ubuntu. Locate the guava jar file in the Hive lib directory:

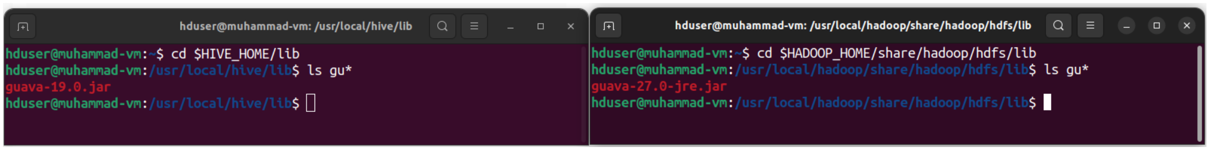
```
$cd $HIVE_HOME/lib
$ls gu*
```



```
hduser@muhammad-vm:~$ cd $HIVE_HOME/lib
hduser@muhammad-vm:/usr/local/hive/lib$ ls gu*
guava-19.0.jar
hduser@muhammad-vm:/usr/local/hive/lib$
```

Open another terminal. Locate the guava jar file in the Hadoop lib directory as well:

```
$cd $HADOOP_HOME/share/hadoop/hdfs/lib
$ls gu*
```



hduser@muhammad-vm: /usr/local/hive/lib
hduser@muhammad-vm: /usr/local/hadoop/share/hadoop/hdfs/lib

```
hduser@muhammad-vm:~$ cd $HIVE_HOME/lib
hduser@muhammad-vm:/usr/local/hive/lib$ ls gu*
guava-19.0.jar
hduser@muhammad-vm:/usr/local/hive/lib$
```

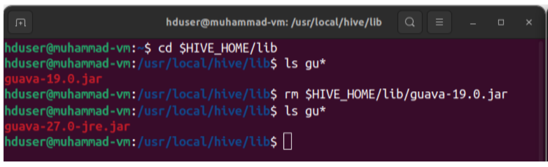
```
hduser@muhammad-vm:~$ cd $HADOOP_HOME/share/hadoop/hdfs/lib
hduser@muhammad-vm:/usr/local/hadoop/share/hadoop/hdfs/lib$ ls gu*
guava-27.0-jre.jar
hduser@muhammad-vm:/usr/local/hadoop/share/hadoop/hdfs/lib$
```

The two listed versions are not compatible and are causing the error. Remove the existing guava file from the Hive lib directory:

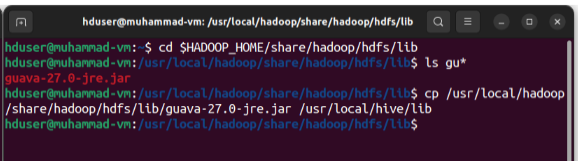
```
$rm $HIVE_HOME/lib/guava-19.0.jar
```

Copy the guava file from the Hadoop lib directory to the Hive lib directory:

```
$cp /usr/local/hadoop/share/hadoop/hdfs/lib/guava-27.0-jre.jar
/usr/local/hive/lib
```



```
hduser@muhammad-vm:~$ cd $HIVE_HOME/lib
hduser@muhammad-vm:/usr/local/hive/lib$ ls gu*
guava-19.0.jar
hduser@muhammad-vm:/usr/local/hive/lib$ rm $HIVE_HOME/lib/guava-19.0.jar
hduser@muhammad-vm:/usr/local/hive/lib$ ls gu*
guava-27.0-jre.jar
hduser@muhammad-vm:/usr/local/hive/lib$ 
```

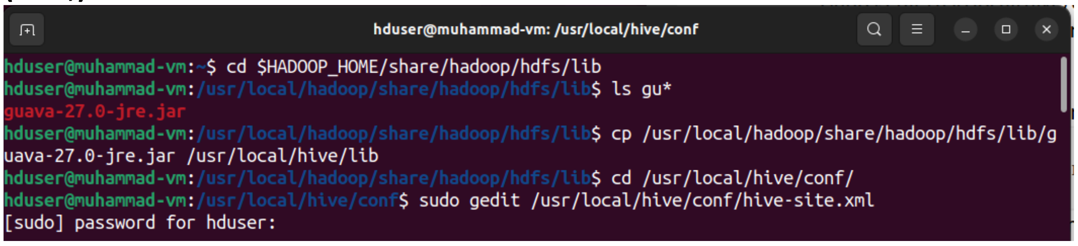


```
hduser@muhammad-vm:~$ cd $HADOOP_HOME/share/hadoop/hdfs/lib
hduser@muhammad-vm:/usr/local/hadoop/share/hadoop/hdfs/lib$ ls gu*
guava-27.0-jre.jar
hduser@muhammad-vm:/usr/local/hadoop/share/hadoop/hdfs/lib$ cp /usr/local/hadoop
/share/hadoop/hdfs/lib/guava-27.0-jre.jar /usr/local/hive/lib
hduser@muhammad-vm:/usr/local/hadoop/share/hadoop/hdfs/lib$
```

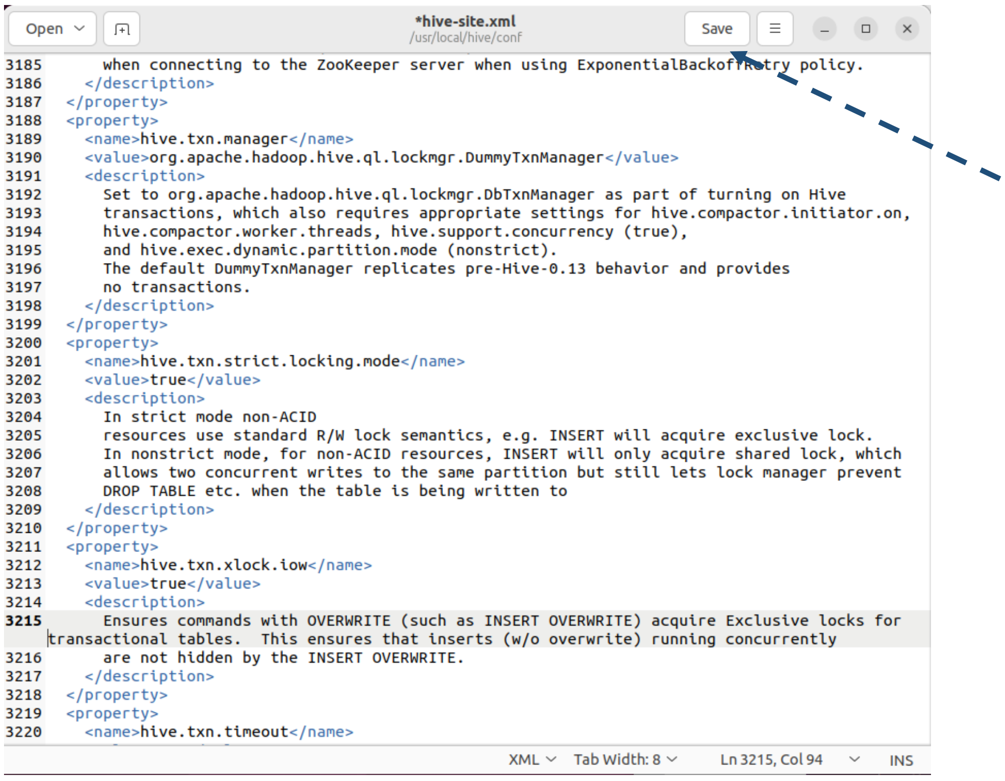
You have to remove one character from the file named as `"/usr/local/hive/conf/hivesite.xml"` by using the following commands.

```
$cd /usr/local/hive/conf/
$sudo gedit /usr/local/hive/conf/hive-site.xml
```

Move the cursor to the line no 3215 and column no 96 and remove the four characters **(&#8;)**.



```
hduser@muhammad-vm:~$ cd $HADOOP_HOME/share/hadoop/hdfs/lib
hduser@muhammad-vm:/usr/local/hadoop/share/hadoop/hdfs/lib$ ls gu*
guava-27.0-jre.jar
hduser@muhammad-vm:/usr/local/hadoop/share/hadoop/hdfs/lib$ cp /usr/local/hadoop/share/hadoop/hdfs/lib/g
uava-27.0-jre.jar /usr/local/hive/lib
hduser@muhammad-vm:/usr/local/hadoop/share/hadoop/hdfs/lib$ cd /usr/local/hive/conf/
hduser@muhammad-vm:/usr/local/hive/conf$ sudo gedit /usr/local/hive/conf/hive-site.xml
[sudo] password for hduser:
```



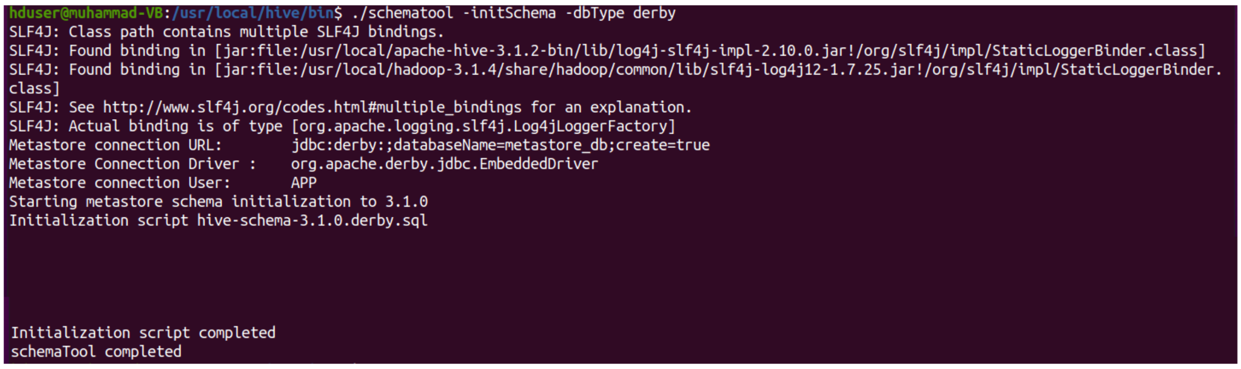
```
3185        when connecting to the ZooKeeper server when using ExponentialBackoffRetry policy.
3186      </description>
3187    </property>
3188    <property>
3189      <name>hive.txn.manager</name>
3190      <value>org.apache.hadoop.hive.ql.lockmgr.DummyTxnManager</value>
3191      <description>
3192        Set to org.apache.hadoop.hive.ql.lockmgr.DbTxnManager as part of turning on Hive
3193        transactions, which also requires appropriate settings for hive.compactor.initiator.on,
3194        hive.compactor.worker.threads, hive.support.concurrency (true),
3195        and hive.exec.dynamic.partition.mode (nonstrict).
3196        The default DummyTxnManager replicates pre-Hive-0.13 behavior and provides
3197        no transactions.
3198      </description>
3199    </property>
3200    <property>
3201      <name>hive.txn.strict.locking.mode</name>
3202      <value>true</value>
3203      <description>
3204        In strict mode non-ACID
3205        resources use standard R/W lock semantics, e.g. INSERT will acquire exclusive lock.
3206        In nonstrict mode, for non-ACID resources, INSERT will only acquire shared lock, which
3207        allows two concurrent writes to the same partition but still lets lock manager prevent
3208        DROP TABLE etc. when the table is being written to
3209      </description>
3210    </property>
3211    <property>
3212      <name>hive.txn.xlock.iow</name>
3213      <value>true</value>
3214      <description>
3215        Ensures commands with OVERWRITE (such as INSERT OVERWRITE) acquire Exclusive locks for
     transactional tables.  This ensures that inserts (w/o overwrite) running concurrently
3216        are not hidden by the INSERT OVERWRITE.
3217      </description>
3218    </property>
3219    <property>
3220      <name>hive.txn.timeout</name>
```

Save the file contents after removal of the above-mentioned characters and close the terminal.

Use the schematool command once again to initiate the Derby database:

```
$HIVE_HOME/bin/schematool -initSchema -dbType derby
```

The process can take a few moments to complete.



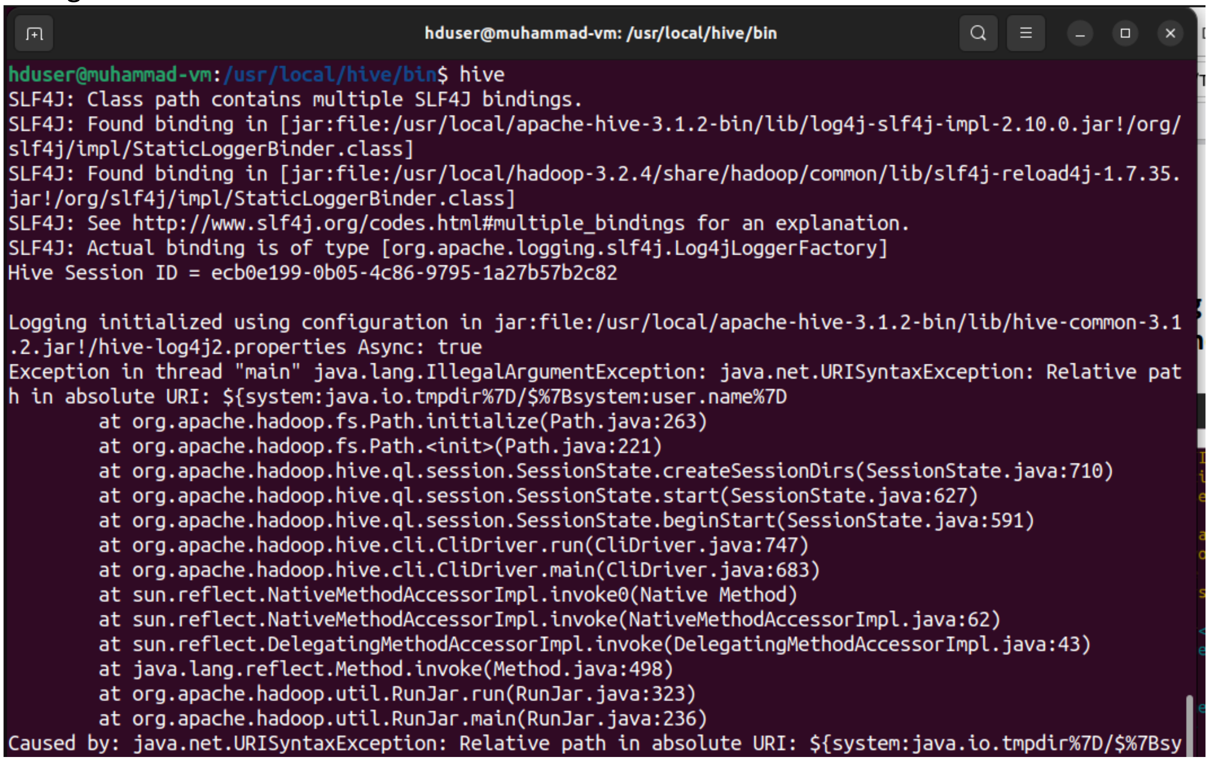
```
hduser@muhammad-VB:/usr/local/hive/bin$ ./schematool -initSchema -dbType derby
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/apache-hive-3.1.2-bin/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-3.1.4/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.
class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Metastore connection URL:        jdbc:derby:;databaseName=metastore_db;create=true
Metastore Connection Driver :    org.apache.derby.jdbc.EmbeddedDriver
Metastore connection User:       APP
Starting metastore schema initialization to 3.1.0
Initialization script hive-schema-3.1.0.derby.sql



 Initialization script completed
 schemaTool completed
```

Launch Hive Client Shell on Ubuntu

Start the Hive command-line interface using the following commands:

```
$cd /usr/local/hive/bin
$hive
```

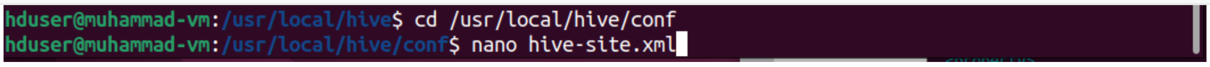
You are now able to issue SQL-like commands and directly interact with HDFS. If you are still facing some error as mentioned below



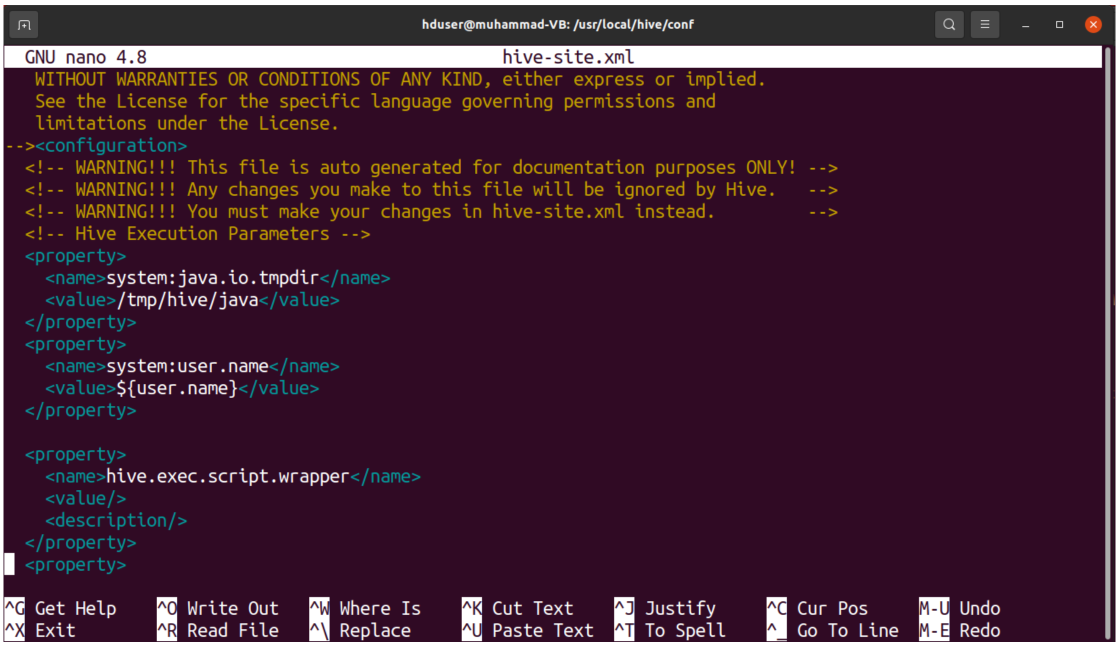
```
hduser@muhammad-vm: /usr/local/hive/bin
hduser@muhammad-vm:/usr/local/hive/bin$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/apache-hive-3.1.2-bin/lib/log4j-slf4j-impl-2.10.0.jar!/org/
slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-3.2.4/share/hadoop/common/lib/slf4j-reload4j-1.7.35.
jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = ecb0e199-0b05-4c86-9795-1a27b57b2c82

Logging initialized using configuration in jar:file:/usr/local/apache-hive-3.1.2-bin/lib/hive-common-3.1
.2.jar!/hive-log4j2.properties Async: true
Exception in thread "main" java.lang.IllegalArgumentException: java.net.URISyntaxException: Relative pat
h in absolute URI: ${system:java.io.tmpdir%7D/$%7Bsystem:user.name%7D
        at org.apache.hadoop.fs.Path.initialize(Path.java:263)
        at org.apache.hadoop.fs.Path.<init>(Path.java:221)
        at org.apache.hadoop.hive.ql.session.SessionState.createSessionDirs(SessionState.java:710)
        at org.apache.hadoop.hive.ql.session.SessionState.start(SessionState.java:627)
        at org.apache.hadoop.hive.ql.session.SessionState.beginStart(SessionState.java:591)
        at org.apache.hadoop.hive.cli.CliDriver.run(CliDriver.java:747)
        at org.apache.hadoop.hive.cli.CliDriver.main(CliDriver.java:683)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at org.apache.hadoop.util.RunJar.run(RunJar.java:323)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:236)
Caused by: java.net.URISyntaxException: Relative path in absolute URI: ${system:java.io.tmpdir%7D/$%7Bsy
```

Then add the following lines at the top of **hive-site.xml**, save and then run the hive command

```
$cd /usr/local/hive/conf
$nano hive-site.xml
```



```
hduser@muhammad-vm:/usr/local/hive$ cd /usr/local/hive/conf
hduser@muhammad-vm:/usr/local/hive/conf$ nano hive-site.xml
```



```
  GNU nano 4.8                               hive-site.xml
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
   See the License for the specific language governing permissions and
   limitations under the License.
--><configuration>
  <!-- WARNING!!! This file is auto generated for documentation purposes ONLY! -->
  <!-- WARNING!!! Any changes you make to this file will be ignored by Hive.   -->
  <!-- WARNING!!! You must make your changes in hive-site.xml instead.         -->
  <!-- Hive Execution Parameters -->
  <property>
    <name>system:java.io.tmpdir</name>
    <value>/tmp/hive/java</value>
  </property>
  <property>
    <name>system:user.name</name>
    <value>${user.name}</value>
  </property>

  <property>
    <name>hive.exec.script.wrapper</name>
    <value/>
    <description/>
  </property>
  <property>

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos    M-U Undo
^X Exit       ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line M-E Redo
```

```
 <property>
    <name>system:java.io.tmpdir</name>
    <value>/tmp/hive/java</value>
  </property>
  <property>
    <name>system:user.name</name>
    <value>${user.name}</value>
  </property>
```

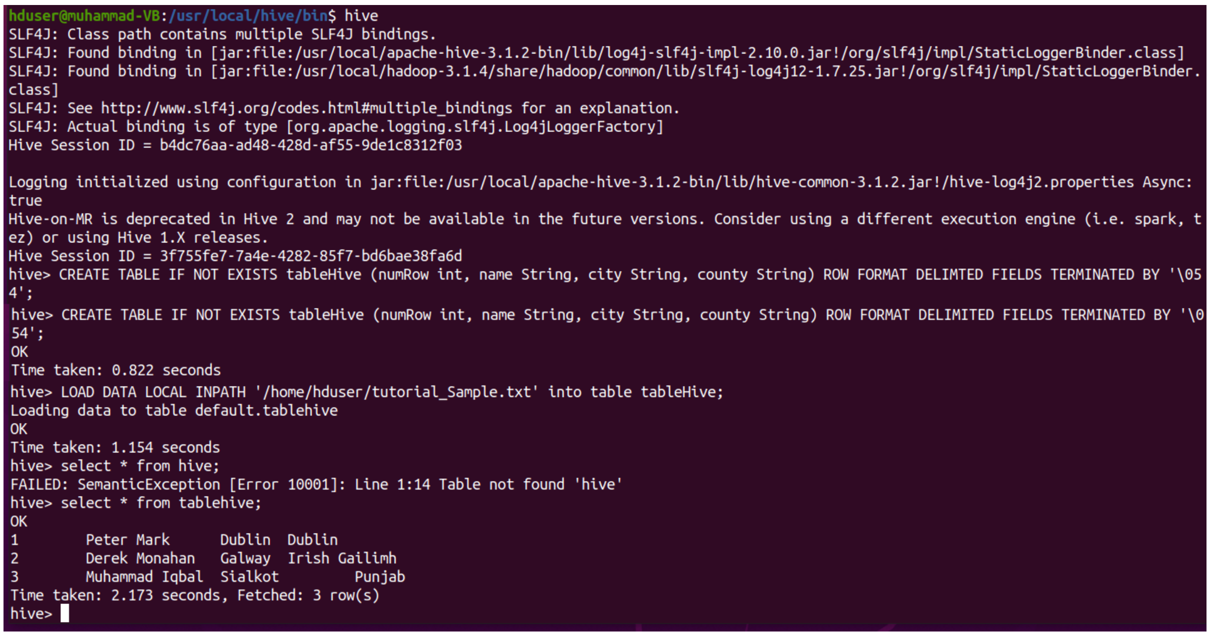
Save the hive-site.xml file using the nano editor (ctrl + x, y and Hit the Enter key)
Also create a folder java on the hadoop folder by using the following commands

```
$hadoop fs -mkdir -p /user/hive/java
$hadoop fs -chmod g+w /user/hive/java
```

Launch the hive command after this update and perform the following actions

- Start HDFS and Yarn services.
- Start Hive shell using the hive command

```
•  $cd /usr/local/hive/bin
•  $hive
```



```
hduser@muhammad-VB:/usr/local/hive/bin$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/apache-hive-3.1.2-bin/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-3.1.4/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.
class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = b4dc76aa-ad48-428d-af55-9de1c8312f03

Logging initialized using configuration in jar:file:/usr/local/apache-hive-3.1.2-bin/lib/hive-common-3.1.2.jar!/hive-log4j2.properties Async:
true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, t
ez) or using Hive 1.X releases.
Hive Session ID = 3f755fe7-7a4e-4282-85f7-bd6bae38fa6d
hive> CREATE TABLE IF NOT EXISTS tableHive (numRow int, name String, city String, county String) ROW FORMAT DELIMTED FIELDS TERMINATED BY '\05
4';
hive> CREATE TABLE IF NOT EXISTS tableHive (numRow int, name String, city String, county String) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\0
54';
OK
Time taken: 0.822 seconds
hive> LOAD DATA LOCAL INPATH '/home/hduser/tutorial_Sample.txt' into table tableHive;
Loading data to table default.tablehive
OK
Time taken: 1.154 seconds
hive> select * from hive;
FAILED: SemanticException [Error 10001]: Line 1:14 Table not found 'hive'
hive> select * from tablehive;
OK
1       Peter Mark      Dublin  Dublin
2       Derek Monahan   Galway  Irish Gailimh
3       Muhammad Iqbal  Sialkot         Punjab
Time taken: 2.173 seconds, Fetched: 3 row(s)
hive>
```

- Create a table using the command:

```
hive> CREATE TABLE IF NOT EXISTS tableHive (numRow int,name String,city
      String,county String,country String) ROW FORMAT DELIMITED FIELDS
      TERMINATED BY '\054';
```

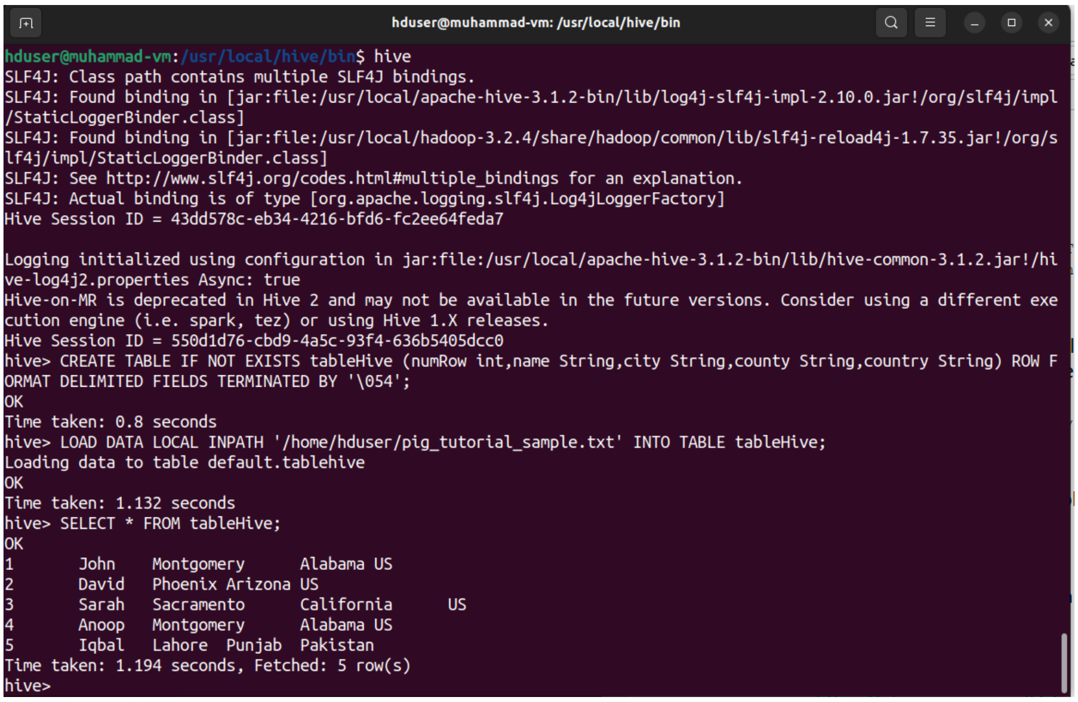
- Load the data from the local filesystem using the command:

```
hive> LOAD DATA LOCAL INPATH '/home/hduser/pig_tutorial_sample.txt' INTO
TABLE tableHive;
```

- Use the SELECT statement to display the data:

```
hive> SELECT * FROM tableHive;
```

- View the contents of the folder in HDFS: **/user/hive/warehouse/tablehive** by using this command as mentioned below

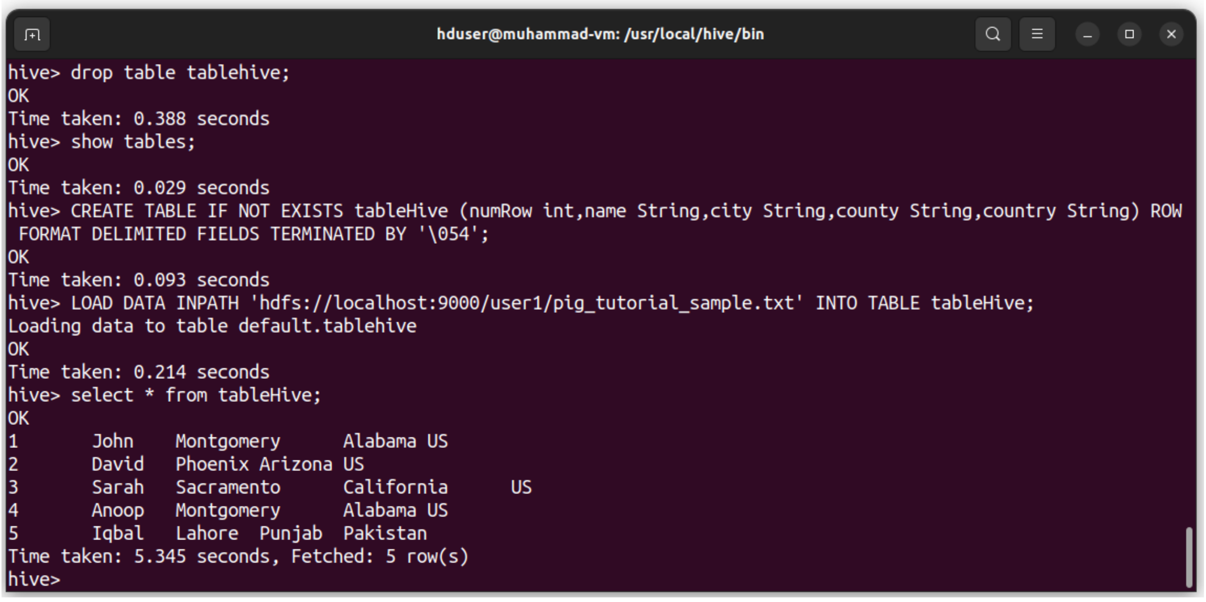


```
hduser@muhammad-vm: /usr/local/hive/bin
hduser@muhammad-vm:/usr/local/hive/bin$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/apache-hive-3.1.2-bin/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl
/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-3.2.4/share/hadoop/common/lib/slf4j-reload4j-1.7.35.jar!/org/s
lf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = 43dd578c-eb34-4216-bfd6-fc2ee64feda7

Logging initialized using configuration in jar:file:/usr/local/apache-hive-3.1.2-bin/lib/hive-common-3.1.2.jar!/hi
ve-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different exe
cution engine (i.e. spark, tez) or using Hive 1.X releases.
Hive Session ID = 550d1d76-cbd9-4a5c-93f4-636b5405dcc0
hive> CREATE TABLE IF NOT EXISTS tableHive (numRow int,name String,city String,county String,country String) ROW F
ORMAT DELIMITED FIELDS TERMINATED BY '\054';
OK
Time taken: 0.8 seconds
hive> LOAD DATA LOCAL INPATH '/home/hduser/pig_tutorial_sample.txt' INTO TABLE tableHive;
Loading data to table default.tablehive
OK
Time taken: 1.132 seconds
hive> SELECT * FROM tableHive;
OK
1       John    Montgomery      Alabama US
2       David   Phoenix Arizona US
3       Sarah   Sacramento      California      US
4       Anoop   Montgomery      Alabama US
5       Iqbal   Lahore  Punjab  Pakistan
Time taken: 1.194 seconds, Fetched: 5 row(s)
hive>
```

Read Data from Hadoop

To understand how to load the data from hadoop, remove the tableHive from Apache Hive and create the table again and load the data from hdfs using the commands as mentioned in the screenshot.

```
hive>drop table tableHive
hive show tables;
hive>CREATE TABLE IF NOT EXISTS tableHive (numRow int, name String,
city String, county String, country String) ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\054';
hive>LOAD DATA INPATH
'hdfs://localhost:9000/user1/pig_tutorial_sample.txt' INTO TABLE
tableHive;
hive>SELECT * FROM tableHive;
```



```
hduser@muhammad-vm: /usr/local/hive/bin
hive> drop table tablehive;
OK
Time taken: 0.388 seconds
hive> show tables;
OK
Time taken: 0.029 seconds
hive> CREATE TABLE IF NOT EXISTS tableHive (numRow int,name String,city String,county String,country String) ROW
 FORMAT DELIMITED FIELDS TERMINATED BY '\054';
OK
Time taken: 0.093 seconds
hive> LOAD DATA INPATH 'hdfs://localhost:9000/user1/pig_tutorial_sample.txt' INTO TABLE tableHive;
Loading data to table default.tablehive
OK
Time taken: 0.214 seconds
hive> select * from tableHive;
OK
1       John    Montgomery      Alabama US
2       David   Phoenix Arizona US
3       Sarah   Sacramento      California      US
4       Anoop   Montgomery      Alabama US
5       Iqbal   Lahore  Punjab  Pakistan
Time taken: 5.345 seconds, Fetched: 5 row(s)
hive>
```

Further exploration can be performed by reading a book chapter provided on Moodle and references provided at the end of the tutorial.

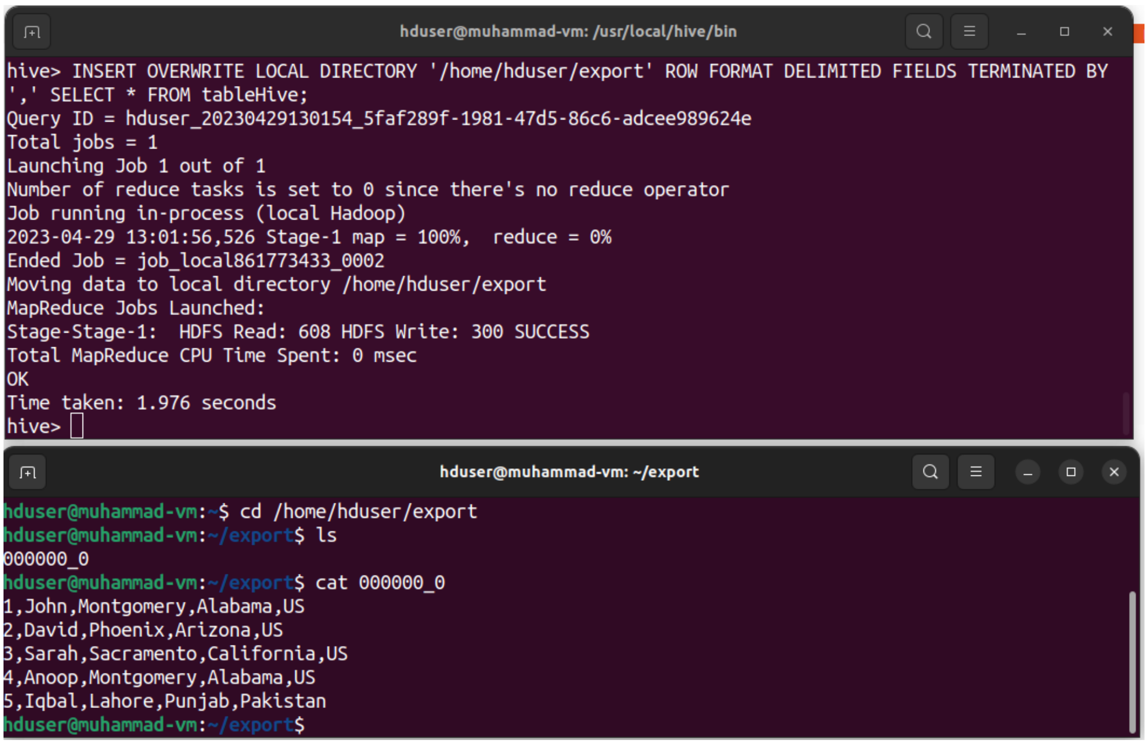
Export HIVE TABLE data to Ubuntu localdrive (/home/hduser/) and HADOOP Directories

Export to Ubuntu local Drive (/home/hduser/) directory

```
hive>INSERT OVERWRITE LOCAL DIRECTORY '/home/hduser/export' ROW
FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT * FROM tableHive;
```

Open an new terminal and check the output obtained in the export directory.

```
$cd /home/hduser/export
$ls
$cat 000000_0
```



```
hive> INSERT OVERWRITE LOCAL DIRECTORY '/home/hduser/export' ROW FORMAT DELIMITED FIELDS TERMINATED BY
',' SELECT * FROM tableHive;
Query ID = hduser_20230429130154_5faf289f-1981-47d5-86c6-adcee989624e
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Job running in-process (local Hadoop)
2023-04-29 13:01:56,526 Stage-1 map = 100%,  reduce = 0%
Ended Job = job_local861773433_0002
Moving data to local directory /home/hduser/export
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 608 HDFS Write: 300 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Time taken: 1.976 seconds
hive>
```

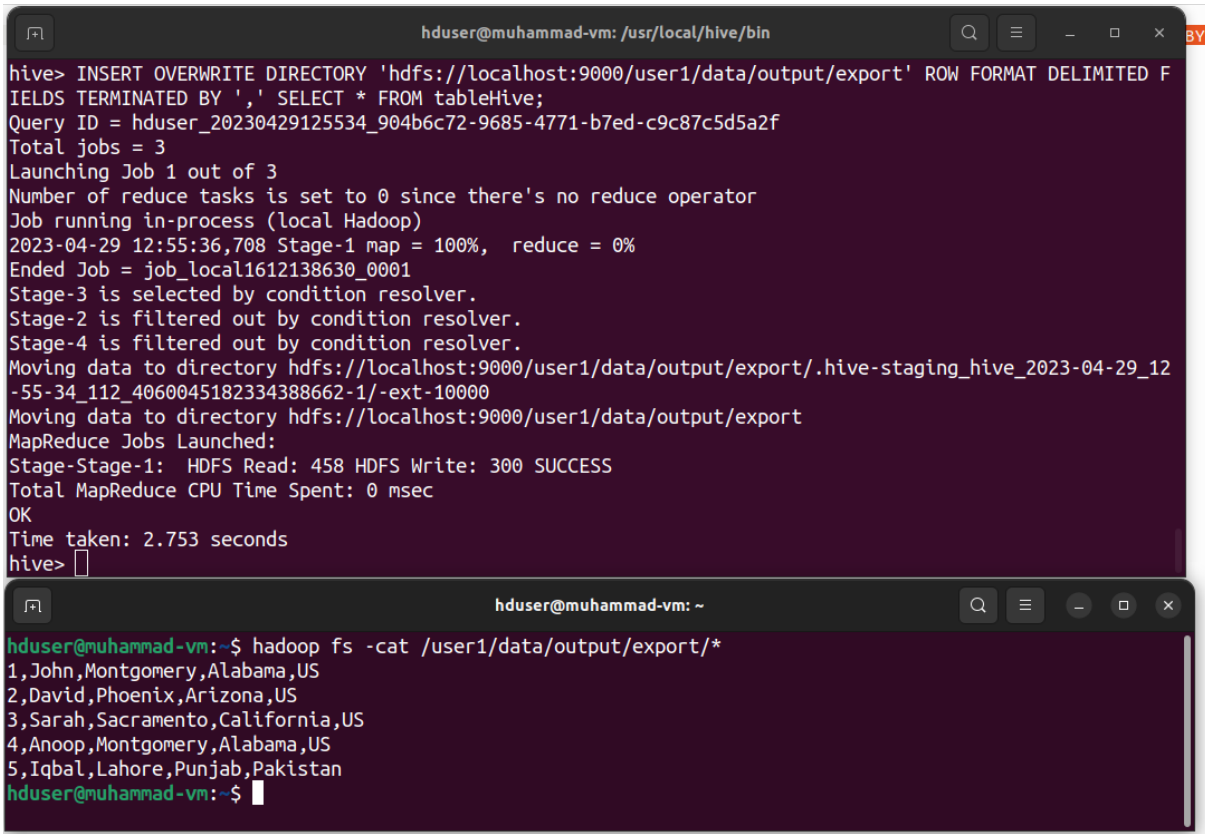
```
hduser@muhammad-vm:~$ cd /home/hduser/export
hduser@muhammad-vm:~/export$ ls
000000_0
hduser@muhammad-vm:~/export$ cat 000000_0
1,John,Montgomery,Alabama,US
2,David,Phoenix,Arizona,US
3,Sarah,Sacramento,California,US
4,Anoop,Montgomery,Alabama,US
5,Iqbal,Lahore,Punjab,Pakistan
hduser@muhammad-vm:~/export$
```

Export to HADOOP directory

```
hive>INSERT OVERWRITE DIRECTORY
'hdfs://localhost:9000/user1/data/output/export' ROW FORMAT
DELIMITED FIELDS TERMINATED BY ',' SELECT * FROM tableHive;
```

Open an new terminal and check the output obtained in the export directory.

```
$hadoop fs -cat /user1/data/output/export/*
```


hduser@muhammad-vm: /usr/local/hive/bin

```
hive> INSERT OVERWRITE DIRECTORY 'hdfs://localhost:9000/user1/data/output/export' ROW FORMAT DELIMITED F
IELDS TERMINATED BY ',' SELECT * FROM tableHive;
Query ID = hduser_20230429125534_904b6c72-9685-4771-b7ed-c9c87c5d5a2f
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Job running in-process (local Hadoop)
2023-04-29 12:55:36,708 Stage-1 map = 100%,  reduce = 0%
Ended Job = job_local1612138630_0001
Stage-3 is selected by condition resolver.
Stage-2 is filtered out by condition resolver.
Stage-4 is filtered out by condition resolver.
Moving data to directory hdfs://localhost:9000/user1/data/output/export/.hive-staging_hive_2023-04-29_12
-55-34_112_4060045182334388662-1/-ext-10000
Moving data to directory hdfs://localhost:9000/user1/data/output/export
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 458 HDFS Write: 300 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Time taken: 2.753 seconds
hive>
```

hduser@muhammad-vm: ~

```
hduser@muhammad-vm:~$ hadoop fs -cat /user1/data/output/export/*
1,John,Montgomery,Alabama,US
2,David,Phoenix,Arizona,US
3,Sarah,Sacramento,California,US
4,Anoop,Montgomery,Alabama,US
5,Iqbal,Lahore,Punjab,Pakistan
hduser@muhammad-vm:~$
```

Reference:

- https://phoenixnap.com/kb/install-hive-on-ubuntu
- https://hive.apache.org/
- Apache Hive Essentials, Dayong Du, Packt Publishing, June 2018.