1. Precondition: hadoop is installed
2. Install Hive
   1. Download Hive, store to your local (/usr/local/hive )
   2. Setting up environment for Hive with ~/.bashrc file

*export HIVE\_HOME=/usr/local/hive*

*export PATH=$PATH:$HIVE\_HOME/bin*

*export CLASSPATH=$CLASSPATH:/usr/local/Hadoop/lib/\*:.*

*export CLASSPATH=$CLASSPATH:/usr/local/hive/lib/\*:.*

Run: source ~/.bashrc to set path

* 1. Configuring Hive with Hadoop

To configure Hive with Hadoop, you need to edit the hive-env.sh file, which is placed in the $HIVE\_HOME/conf directory. The following commands redirect to Hive config folder and copy the template file:

*$ cd $HIVE\_HOME/conf*

*$ cp hive-env.sh.template hive-env.sh*

Edit the hive-env.sh file by appending the following line:

*export HADOOP\_HOME=/usr/local/hadoop*

* 1. Creating Hive warehouse directory in HDFS file system

In addition, we must use below HDFS commands to create /tmp and /user/hive/warehouse (hive.metastore.warehouse.dir) and set them chmod g+w before we can create a table in Hive:

*hduser@laptop:~$ hdfs dfs -mkdir /tmp*

*hduser@laptop:~$ hdfs dfs -mkdir /user/hive/warehouse*

*hduser@laptop:~$ hdfs dfs -chmod g+w /tmp*

*hduser@laptop:~$ hdfs dfs -chmod g+w /user/hive/warehouse*

* 1. Metastore schema initialization

Starting from Hive 2.1, we need to run the schematool command below as an initialization step. In our case, we use derby as db type (autoCreateSchema property is set false in hive-site.xml configuration file:

*hduser@laptop:/usr/local/apache-hive-2.1.0-bin/bin$ schematool -dbType derby -initSchema*

*SLF4J: Class path contains multiple SLF4J bindings.*

*SLF4J: Found binding in [jar:file:/usr/local/apache-hive-2.1.0-bin/lib/log4j-slf4j-impl-2.4.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]*

*SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.5.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 2.1.0*

*Initialization script hive-schema-2.1.0.derby.sql*

*Initialization script completed*

*schemaTool completed*

Note: If run error “Schema initialization FAILED! Metastore state would be inconsistent” then remove already metastore by below command:

*mv metastore\_db metastore\_db.tmp*

Then re-run:

*schematool -dbType derby -initSchema*

However, I can't find either metastore\_db or metastore\_db.tmp folder under install path, so I tried:

*find /usr/ -name hive-schema-2.0.0.derby.sql*

*vi /usr/local/Cellar/hive/2.0.1/libexec/scripts/metastore/upgrade/derby/hive-schema-2.0.0.derby.sql*

comment the 'NUCLEUS\_ASCII' function and 'NUCLEUS\_MATCHES' function

rerun schematool -dbType derby -initSchema, then everything goes well!

* 1. Verifying Hive Installation by running Hive CLI

To use the Hive command line interface (CLI) from the shell, issue bin/hive command to verify Hive:

*hduser@laptop:~$ $HIVE\_HOME/bin/hive*

We may get couple of errors when we try to start hive via **bin/hive** command. The followings are the errors and corresponding fixes:

Error #1:

*Exception in thread "main" java.lang.RuntimeException: Couldn't create directory ${system:java.io.tmpdir}/${hive.session.id}\_resources*

Fix #1: edit hive-site.xml:

*<property>*

*<name>hive.downloaded.resources.dir</name>*

*<!--*

*<value>${system:java.io.tmpdir}/${hive.session.id}\_resources</value>*

*-->*

*<value>/home/hduser/hive/tmp/${hive.session.id}\_resources</value>*

*<description>Temporary local directory for added resources in the remote file system.</description>*

*</property>*

Error #2:

*java*.net.URISyntaxException: Relative path in absolute URI: ${system:java.io.tmpdir%7D/$%7Bsystem:user.name%7D

Fix #2: replace ${system:java.io.tmpdir}/${system:user.name} by /tmp/mydir in hive-site.xml (see Confluence - AdminManual Configuration):

*<property>*

*<name>hive.exec.local.scratchdir</name>*

*<!--*

*<value>${system:java.io.tmpdir}/${system:user.name}</value>*

*-->*

*<value>/tmp/mydir</value>*

*<description>Local scratch space for Hive jobs</description>*

*</property>*

Now that we fixed the errors, let's start Hive CLI:

*hduser@laptop:/usr/local/apache-hive-2.1.0-bin/bin$ hive*

*SLF4J: Class path contains multiple SLF4J bindings.*

*SLF4J: Found binding in [jar:file:/usr/local/apache-hive-2.1.0-bin/lib/log4j-slf4j-impl-2.4.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]*

*SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.5.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]*

*Logging initialized using configuration in jar:file:/usr/local/apache-hive-2.1.0-bin/lib/hive-common-2.1.0.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, tez) or using Hive 1.X releases.*

To display all the tables:

*hive> show tables;*

*OK*

*Time taken*: 4.603 seconds

We can exit from that Hive shell by using exit command:

*hive> exit;*

*hduser@laptop*:/usr/local/apache-hive-2.1.0-bin/bin$