

-----HIVE Installation-----

1:

```
sudo mkdir /usr/hive
sudo cp apache-hive-2.3.6-bin.tar.gz /usr/hive/
sudo tar xvzf apache-hive-2.3.6-bin.tar.gz
remove apache-hive-2.3.6-bin.tar.gz
```

2:

Add to bashrc

```
export HIVE_HOME=/usr/hive/apache-hive-2.3.6-bin
export PATH=$PATH:$HIVE_HOME/bin:$HIVE_HOME/sbin
```

3.

```
source ~/.bashrc
```

4. Configuration

```
$ 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
```

Hive installation is completed successfully. Now you require an external database server to configure Metastore. We use Apache Derby database.

5.

```
$ wget
```

```
http://archive.apache.org/dist/db/derby/db-derby-10.4.2.0/db-derby-10.4.2.0-bin.tar.gz
```

```
$ tar zxvf db-derby-10.4.2.0-bin.tar.gz
```

Create derby folder

```
mv db-derby-10.4.2.0-bin /usr/local/derby
```

5.

5.Add to bashrc

```
export DERBY_HOME=/usr/local/derby
```

```
export PATH=$PATH:$DERBY_HOME/bin
```

```
export
```

```
CLASSPATH=$CLASSPATH:$DERBY_HOME/lib/derby.jar:$DERBY_HOME/lib/derbytools.jar
```

```
5.$ source ~/.bashrc
```

6.

```
$ mkdir $DERBY_HOME/data
```

Configuring Metastore means specifying to Hive where the database is stored. You can do this by editing the hive-site.xml file, which is in the \$HIVE_HOME/conf directory. First of all, copy the template file using the following command:

```
$ cd $HIVE_HOME/conf
```

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

```
<property>
```

```
  <name>javax.jdo.option.ConnectionURL</name>
```

```
  <value>jdbc:derby://localhost:1527/metastore_db;create=true </value>
```

```
  <description>JDBC connect string for a JDBC metastore </description>
```

```
</property>
```

7.

Create a file named jpox.properties and add the following lines into it:

```
javax.jdo.PersistenceManagerFactoryClass =
```

```
org.jpox.PersistenceManagerFactoryImpl
```

```
org.jpox.autoCreateSchema = false
```

```
org.jpox.validateTables = false
```

```
org.jpox.validateColumns = false
```

```
org.jpox.validateConstraints = false
```

```
org.jpox.storeManagerType = rdbms
```

```
org.jpox.autoCreateSchema = true
```

```
org.jpox.autoStartMechanismMode = checked
```

```
org.jpox.transactionIsolation = read_committed
```

```
javax.jdo.option.DetachAllOnCommit = true
```

```
javax.jdo.option.NontransactionalRead = true
```

```
javax.jdo.option.ConnectionDriverName =
```

```
org.apache.derby.jdbc.ClientDriver
```

```
javax.jdo.option.ConnectionURL =
```

```
jdbc:derby://hadoop1:1527/metastore_db;create = true
```

```
javax.jdo.option.ConnectionUserName = ash
```

```
javax.jdo.option.ConnectionPassword = youpassword
```

```
chmod g+w
```

```
$HADOOP_HOME/bin/hadoop fs -mkdir /tmp
```

```
$HADOOP_HOME/bin/hadoop fs -mkdir /user/hive/warehouse
```

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

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

Or

```
hdfs dfs -mkdir -p /user/hive/warehouse
```

```
hdfs dfs -mkdir -p /tmp/hive
```

Now give permission by giving the following command:-

```
hdfs dfs -chmod 777 /user/hive/warehouse
```

```
hdfs dfs -chmod 777 /tmp/
```

```
hdfs dfs -chmod 777 /tmp/hive
```

Since having multiple slf4j file will give errors we have to remove the same from hive lib directory by giving the following command:

Go to hive lib folder:

```
cd /usr/local/hive/lib
```

```
rm log4j-slf4j-impl-2.4.1.jar
```

Or both

- rm lib/hive-jdbc-2.0.0-standalone.jar
- rm lib/log4j-slf4j-impl-2.4.1.jar

Hive 2.1.1 installation is bit tricky here as in pervious version before 2.x the default database was derby and was initialized after the installation but in here we have to manually do the same by the following steps:-

1. Before you run hive for the first time, remove previous metastore information:

Go to hive bin directory and run the below command:

```
mv metastore_db metastore_db.tmp
```

2. Now run the schematool command:

Open a new terminal and give following command:-

```
schematool -initSchema -dbType derby
```

(If successful it will display success else error will be shown schematool not found, etc).

```
3.sudo chmod -R 777 /usr/hive  
mv db-derby-10.4.2.0-bin /usr/local/derby
```

:Refuse Connection Error

Change value for folder

```
<property>  
  <name>hive.downloaded.resources.dir</name>  
  <value>/tmp/hivetemp</value>  
  <description>Temporary local directory for added resources in the  
remote file system.</description>  
</property>
```