Step 1: Downloading Hive

We use hive-0.14.0 in this tutorial. You can download it by visiting the following link [http://apache.petsads.us/hive/hive-0.14.0/.](http://apache.petsads.us/hive/hive-0.14.0/) Let us assume it gets downloaded onto the /Downloads directory. Here, we download Hive archive named “apache-hive-0.14.0-bin.tar.gz” for this tutorial. The following command is used to verify the download:

$ cd Downloads

$ ls

On successful download, you get to see the following response:

apache-hive-0.14.0-bin.tar.gz

Step 2: Installing Hive

The following steps are required for installing Hive on your system. Let us assume the Hive archive is downloaded onto the /Downloads directory.

Extracting and verifying Hive Archive

The following command is used to verify the download and extract the hive archive:

$ tar zxvf apache-hive-0.14.0-bin.tar.gz

$ ls

On successful download, you get to see the following response:

apache-hive-0.14.0-bin apache-hive-0.14.0-bin.tar.gz

Copying files to /usr/local/hive directory

We need to copy the files from the super user “su -”. The following commands are used to copy the files from the extracted directory to the /usr/local/hive” directory.

Setting up environment for Hive

You can set up the Hive environment by appending the following lines to **~/.bashrc** file:

export HIVE\_HOME=/home/xxxx/hivefolder

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

export CLASSPATH=$CLASSPATH:$HADOOP\_HOME/lib/\*:.

export CLASSPATH=$CLASSPATH:$HIVE\_HOME/lib/\*:.

The following command is used to execute ~/.bashrc file.

$ source ~/.bashrc

Step 5: Configuring Hive

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=$HADOOP\_HOME

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

Step 6: Downloading and Installing Apache Derby

Follow the steps given below to download and install Apache Derby:

Downloading Apache Derby

The following command is used to download Apache Derby. It takes some time to download.

$ cd ~

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

The following command is used to verify the download:

$ ls

On successful download, you get to see the following response:

db-derby-10.4.2.0-bin.tar.gz

Extracting and verifying Derby archive

The following commands are used for extracting and verifying the Derby archive:

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

$ ls

On successful download, you get to see the following response:

db-derby-10.4.2.0-bin db-derby-10.4.2.0-bin.tar.gz

Setting up environment for Derby

You can set up the Derby environment by appending the following lines to **~/.bashrc** file:

export DERBY\_HOME=/home/xxx/derby-10.4….

export PATH=$PATH:$DERBY\_HOME/bin

export CLASSPATH=$CLASSPATH:$DERBY\_HOME/lib/derby.jar:$DERBY\_HOME/lib/derbytools.jar

The following command is used to execute **~/.bashrc** file:

$ source ~/.bashrc

Create a directory to store Metastore

Create a directory named data in $DERBY\_HOME directory to store Metastore data.

$ mkdir $DERBY\_HOME/data

Derby installation and environmental setup is now complete.

Step 7: Configuring Metastore of Hive

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

Edit **hive-site.xml** and append the following lines between the <configuration> and </configuration> tags:

<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>

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 = APP

javax.jdo.option.ConnectionPassword = mine

Step 8: Verifying Hive Installation

Before running Hive, you need to create the **/tmp** folder and a separate Hive folder in HDFS. Here, we use the **/user/hive/warehouse** folder. You need to set write permission for these newly created folders as shown below:

chmod g+w

Now set them in HDFS before verifying Hive. Use the following commands:

$ $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

The following commands are used to verify Hive installation:

$ cd $HIVE\_HOME

$ bin/hive

On successful installation of Hive, you get to see the following response:

Logging initialized using configuration in jar:file:/home/hadoop/hive-0.9.0/lib/hive-common-0.9.0.jar!/hive-log4j.properties

Hive history file=/tmp/hadoop/hive\_job\_log\_hadoop\_201312121621\_1494929084.txt

………………….

hive>

The following sample command is executed to display all the tables:

hive> show tables;

OK

Time taken: 2.798 seconds

hive>