1. Download the Hive packages using below command.

**sudo wget** [**http://archive.apache.org/dist/hive/hive-2.1.1/apache-hive-2.1.1-bin.tar.gz**](http://archive.apache.org/dist/hive/hive-2.1.1/apache-hive-2.1.1-bin.tar.gz)

1. Unzip the files

**sudo tar xvzf apache-hive-2.1.1-bin.tar.gz -C /usr/local/hive**

1. Add paths into .profile and match the final configuration with below written configuration:

**export** **JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64**

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

**export HADOOP\_HOME=/usr/local/hadoop/hadoop-2.7.1**

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

**export HADOOP\_CONF\_DIR=/usr/local/hadoop/hadoop-2.7.1/etc/hadoop**

**export HIVE\_HOME=/usr/local/hive/apache-hive-2.1.1-bin**

**export PATH=$PATH:/usr/local/hive/apache-hive-2.1.1-bin/bin**

**export HADOOP\_MAPRED\_HOME=/usr/local/hadoop/hadoop-2.7.1**

**export HADOOP\_COMMON\_HOME=/usr/local/hadoop/hadoop-2.7.1**

**export YARN\_HOME=/usr/local/hadoop/hadoop-2.7.1**

**export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native**

**export HADOOP\_OPTS="-Djava.library.path=$HADOOP\_HOME/lib"**

**export DERBY\_HOME=/usr/local/db-derby-10.13.1.1-bin**

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

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

1. Now activate the new setting:

**source ~/.profile**

1. Create Hive directories within HDFS following below commands:

**hdfs dfs -chmod g+w /user/hive/warehouse**

**hdfs dfs -chmod g+w /tmp**

**hdfs dfs -chmod g+w /user/hive/warehouse**

**hdfs dfs -chmod g+w /tmp**

1. Configure the hive-env,sh file.

**cd $HIVE\_HOME/conf**

**sudo 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/hadoop-2.7.1(“Hadoop Home Path”)**

1. Setup Apache Derby to run hive:

**cd /tmp**

**wget http://archive.apache.org/dist/db/derby/db-derby-10.13.1.1/db-derby-10.13.1.1-bin.tar.gz**

**sudo tar xvzf db-derby-10.13.1.1-bin.tar.gz -C /usr/local**

1. Match the ~/.profile with the configuration in step 3.
2. Create a directory named “data” in $DERBY\_HOME directory to store Metastore data.

**sudo mkdir $DERBY\_HOME/data**

1. Configure hive-site.xml:

**cd $HIVE\_HOME/conf**

**sudo vi hive-site.xml**

Add the following lines to the new hive-site.xml file configuration:

**<?xml version="1.0" encoding="UTF-8" standalone="no"?>**

**<?xml-stylesheet type="text/xsl" href="configuration.xsl"?><!--**

**Licensed to the Apache Software Foundation (ASF) under one or more**

**contributor license agreements. See the NOTICE file distributed with**

**this work for additional information regarding copyright ownership.**

**The ASF licenses this file to You under the Apache License, Version 2.0**

**(the "License"); you may not use this file except in compliance with**

**the License. You may obtain a copy of the License at**

**http://www.apache.org/licenses/LICENSE-2.0**

**Unless required by applicable law or agreed to in writing, software**

**distributed under the License is distributed on an "AS IS" BASIS,**

**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>**

**<property>**

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

**<value>jdbc:derby:;databaseName=/home/edureka/apache-hive-2.1.0-bin/metastore\_db;create=true</value>**

**<description>**

**JDBC connect string for a JDBC metastore.**

**To use SSL to encrypt/authenticate the connection, provide database-specific SSL flag in the connection URL.**

**For example, jdbc:postgresql://myhost/db?ssl=true for postgres database.**

**</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.uris</name>**

**<value/>**

**<description>Thrift URI for the remote metastore. Used by metastore client to connect to remote metastore.</description>**

**</property>**

**<property>**

**<name>javax.jdo.option.ConnectionDriverName</name>**

**<value>org.apache.derby.jdbc.EmbeddedDriver</value>**

**<description>Driver class name for a JDBC metastore</description>**

**</property>**

**<property>**

**<name>javax.jdo.PersistenceManagerFactoryClass</name>**

**<value>org.datanucleus.api.jdo.JDOPersistenceManagerFactory</value>**

**<description>class implementing the jdo persistence</description>**

**</property>**

**</configuration>**

1. 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**

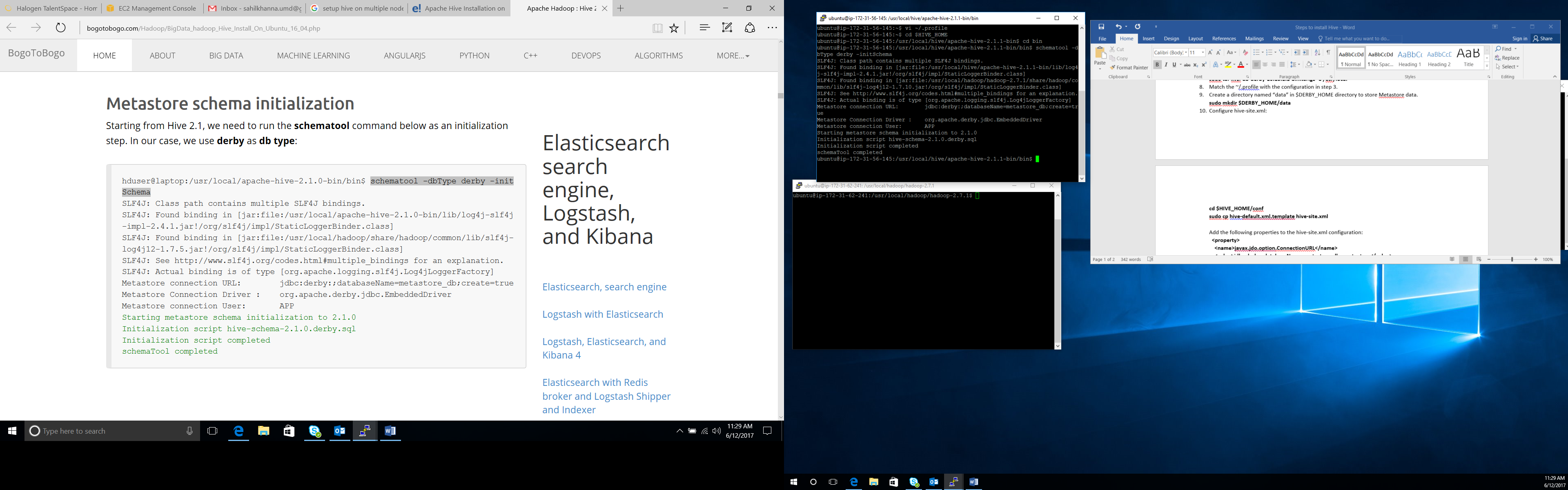
**javax.jdo.option.ConnectionPassword = mine**

1. Set the permission to the hive folder:

**sudo chown -R ubuntu apache-hive-2.1.1-bin**

1. 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:

**schematool -dbType derby –initSchema**



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

**$HIVE\_HOME/bin/hive**

