

[11:43, 1/29/2019] Sachin Patil Sir (Seven Mentor):
Make sure 8GB RAM and 40-50 GB free space...

Ubuntu .iso

Open Vbox --> Click on new --> Prompts Dialouge with Create Virtual Machine --> Write name to machine ex: Ubuntu1 , Type : Linux , Version: Ubuntu 64Bit ----> Make memory as 4096MB ---> Check default is Create Virtual Hard Disk now ---> Make File size more than 30GB and Make sure Virtual Disk Image is selected --> OK

Right click on that machine --> Settings --> Storage --> Empty --> Right side cd image and choose .iso file--> Ok

Double click on machine and start new installation steps--> Normal installation

Shared Folder..

Create shared Folder in ur main OS host..

VBox machine --> Right Click --> Settings --> SharedFolder --> Click on that cd image at right side --> Choose ur sharedfolder --> Select Auto mount --> OK

Start your virtual machine ...means ubuntu..

On top of VBox Devices ---> press insert guest edition ---> Run --> press enter to close

```
Terminal > mkdir ~/new
Terminal > sudo mount -t vboxsf SharedFolderName ~/new
Terminal > sudo gedit /etc/group
Last line check -> vboxsf:UserName
Terminal > sudo gedit /etc/fstab
Add following line
SharedFolderName /home/UserName/new vboxsf defaults 0 0
```

[11:56, 1/30/2019] Sachin Patil Sir (Seven Mentor): -----

[11:56, 1/30/2019] Sachin Patil Sir (Seven Mentor): Unzip the hadoop zip to home..

1. Configure hadoop for java home:

```
Terminal > readlink -f /usr/bin/java | sed "s:bin/java::"

/usr/lib/jvm/java-8-oracle/jre/
```

3 ways to set java home for hadoop.

1. Java_Home
2. Static
3. Dynamic

```
Terminal > sudo gedit hadoop-2.7.7/etc/hadoop/hadoop-env.sh
```

Add following:

```
export JAVA_HOME=/usr/lib/jvm/java-8-oracle/jre/
```

Save it and close

2. Setup password-less ssh (secure shell)

```
Terminal > sudo apt-get install openssh-server openssh-client
```

****If error like could not get the lock on dpkg or apt**

```
Terminal > ps aux | grep apt
```

```
Terminal > ps aux | grep dpkg
```

```
Terminal > sudo kill -9 processid's
```

```
Terminal > ssh-keygen -t rsa -P ""
```

```
Terminal > cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
```

Testing: Terminal > ssh localhost

Type Yes

Restart your machine...

```
Terminal > shutdown -r now
```

3. Setup Configurations..

bashrc work:

```
Terminal > sudo gedit ~/.bashrc
```

Add following lines in it...

#Hadoop Settings:

```
export HADOOP_PREFIX=/home/hduser/hadoop-2.7.7
export PATH=$PATH:$HADOOP_PREFIX/bin
export PATH=$PATH:$HADOOP_PREFIX/sbin
export HADOOP_MAPRED_HOME=${HADOOP_PREFIX}
export HADOOP_COMMON_HOME=${HADOOP_PREFIX}
export HADOOP_HDFS_HOME=${HADOOP_PREFIX}
export YARN_HOME=${HADOOP_PREFIX}
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_PREFIX/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_PREFIX/lib/native"
```

4. Core-site.xml

```
Terminal > sudo gedit hadoop-2.7.7/etc/hadoop/core-site.xml
```

Add following properties in it between <configuration> tags:

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
  </property>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>/home/hduser/hdata</value>
  </property>
</configuration>
```

5. Terminal > sudo mkdir /home/hduser/hdata
> sudo chmod 777 /home/hduser/hdata

6. Edit hdfs-site.xml

Terminal > sudo gedit hadoop-2.7.7/etc/hadoop/hdfs-site.xml

Add following in between <configuration>:

```
<property>
<name>dfs.replication</name>
<value>1</value>
</property>

<property>
<name>dfs.data.dir</name>
<value>/home/hduser/tmp/dfs/name/data</value>
<final>true</final>
</property>

<property>
<name>dfs.name.dir</name>
<value>/home/hduser/tmp/dfs/name</value>
<final>true</final>
</property>
```

7. Terminal > sudo mv hadoop-2.7.7/etc/hadoop/mapred-site.xml.template
hadoop-2.7.7/etc/hadoop/mapred-site.xml

Terminal > sudo gedit hadoop-2.7.7/etc/hadoop/mapred-site.xml

Add following property in <configuration>:

```
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
```

[12:18, 2/4/2019] Sachin Patil Sir (Seven Mentor): -----

13. Hive with Mysql

Apache Hive - DW tool --uses RDBMS for holding schema..

Hive -- Schema--> Mysql

Storage --> hdfs

1. Unzip the hive file in hadoop-2.7.7

2. Edit the bashrc

#Hive Setting

export PATH=\$PATH:/home/sachin/hadoop-2.7.7/apache-hive-2.3.4-bin/bin

export HIVE_HOME=/home/sachin/hadoop-2.7.7/apache-hive-2.3.4-bin

export HIVE_CLASSPATH=\$HADOOP_PREFIX/conf

Save the file and source ~/.bashrc

3. creating warehouse directory on hdfs:

Terminal > hdfs dfs -mkdir -p /user/hive/warehouse

Terminal > hdfs dfs -mkdir -p /tmp

Terminal > hdfs dfs -chmod g+w /user/hive/warehouse

Terminal > hdfs dfs -chmod g+w /tmp

```
4.hive-env.sh
sudo cp hive-env.sh.template hive-env.sh
```

```
/home/sachin/hadoop-2.7.7/apache-hive-2.3.4-bin/conf
```

```
sudo gedit hive-env.sh
```

5. Install mysql

```
Terminal > sudo apt-get update
```

```
Terminal > sudo apt-get install mysql-server
```

```
Error: Could not get lock on apt/dpkg
```

```
Terminal > ps -aux | grep apt
```

```
Terminal > sudo kill -9 processId's
```

```
6.  ** Terminal > sudo mysql -u root -p
mysql > create database hiveMetaStore;
mysql > use hiveMetaStore;
mysql > source /home/sachin/hadoop-2.7.7/apache-hive-2.3.4-
bin/scripts/metastore/upgrade/mysql/hive-schema-2.3.0.mysql.sql;
```

```
mysql > show tables;
mysql > create user 'hiveuser'@'%' identified by 'Hive@123';
mysql > grant all on . to 'hiveuser' @localhost identified by
'Hive@123';
mysql > flush privileges;
mysql > quit;
```

7. hive-site.xml

```
** > sudo cp hive-default.xml.template hive-site.xml
```

```
sudo gedit hadoop-2.7.7/apache-hive-2.3.4-bin/conf/hive-site.xml
```

```
USername, password, url, driver
```

```
querylog.location,scratchdir,downloaded.resources.dir -> /tmp/hive
```

8. Copy mysql connection jar file to lib of hive

```
Terminal > sudo cp Desktop/Hadoop_Setups/mysql-connector-java-5.1.22-
bin.jar hadoop-2.7.7/apache-hive-2.3.4-bin/lib/
```

```
Terminal > sudo chmod 777 hadoop-2.7.7/apache-hive-2.3.4-bin/lib/mysql-
connector-java-5.1.22-bin.jar
```

```
** Error: javabase.jdk internal loader class loader Can not be cast to
java.base
```

```
Terminal > sudo apt install openjdk-8-jdk
```

```
Terminal > sudo update-alternatives --config java
```

```
Output: Enter number: 2
```

Find out java path and replace in bashrc and hadoop-env.sh

```
Error: /tmp/hive on hdfs should be writable..
```

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

9. Terminal > hive

05.02.2019

Apache Sqoop: Import and export

RDBMS --> HDFS ---> RDBMS

1. Unzip the sqoop into hadoop-2.7.7

2. Configure bashrc

Add following into it:

```
#Sqoop
export SQOOP_HOME=/home/sachin/hadoop-2.7.7/sqoop-1.4.7
export PATH=$PATH:$SQOOP_HOME/bin
```

Save it and source ~/.bashrc

3. Configure Sqoop

```
Terminal > cd hadoop-2.7.7/sqoop-1.4.7/conf
> sudo cp sqoop-env-template.sh sqoop-env.sh
> sudo gedit sqoop-env.sh
```

Add following:

```
#Set path to where bin/hadoop is available
export HADOOP_COMMON_HOME=/home/sachin/hadoop-2.7.7

#Set path to where hadoop-*-core.jar is available
export HADOOP_MAPRED_HOME=/home/sachin/hadoop-2.7.7

#Set the path to where bin/hive is available
export HIVE_HOME=/home/sachin/hadoop-2.7.7/apache-hive-2.3.4-bin
```

Save it.

4. Mysql connector:

```
Terminal > sudo cp Desktop/Hadoop_Setups/mysql-connector-java-5.1.22-
bin.jar hadoop-2.7.7/sqoop-1.4.7/lib/
```

```
Terminal > sudo chmod 777 hadoop-2.7.7/sqoop-1.4.7/lib/mysql-connector-
java-5.1.22-bin.jar
```

5. Terminal > sqoop-version

6. Testing Sqoop commands:

Import any table from mysql to hdfs

06.2.2019

Terminal > hbase ----> bashrc ----> env.sh ---> conf xml ---> hadoop and java

HBase Installation:

HBase can be downloaded from <https://hbase.apache.org/downloads.html>

HBase - NoSQL DB relative data insertion and reading .. read/write, Column oriented DB and uses hash table. CAP Thoery (Consistent Availability Partition)

1. Unzip that hbase to hadoop-2.7.7

2. edit bashrc

Add following:

```
#HBase
```

```
export HBASE_HOME=/home/sachin/hadoop-2.7.7/hbase-2.1.1
```

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

Save it.

Terminal > source ~/.bashrc

3. Edit hbase-site.xml

Terminal > sudo gedit hadoop-2.7.7/hbase-2.1.1/conf/hbase-site.xml

Add following:

```
<configuration>
```

```
    <property>
```

```
        <name>hbase.rootdir</name>
```

```
        <value>hdfs://localhost:9000/hbase</value>
```

```
    </property>
```

```
    <property>
```

```
        <name>hbase.cluster.distributed</name>
```

```
        <value>true</value>
```

```
    </property>
```

```
    <property>
```

```
        <name>hbase.zookeeper.quorum</name>
```

```
        <value>localhost</value>
```

```
    </property>
```

```
    <property>
```

```
        <name>dfs.replication</name>
```

```
        <value>1</value>
```

```
    </property>
```

```
    <property>
```

```
        <name>hbase.zookeeper.property.clientPort</name>
```

```
        <value>2181</value>
```

```

    </property>

    <property>

    <name>hbase.zookeeper.property.dataDir</name>

    <value>/home/sachin/hadoop-2.7.7/hbase-2.1.1/zookeeper</value>

    </property>

</configuration>

```

* Error: /usr/bin path is not set in environment variable...
No command is working..

```

Terminal > export PATH = /usr/bin:/bin
Terminal > sudo gedit ~/.bashrc
Terminal > source ~/.bashrc

```

4. Edit hbase-env.sh

```
Terminal > sudo gedit hadoop-2.7.7/hbase-2.1.1/conf/hbase-env.sh
```

Add following:

```

export JAVA_HOME=/usr/lib/jvm/java-8-oracle/jre/
export HBASE_REGIONSERVERS=/home/sachin/hadoop-2.7.7/hbase-
2.1.1/conf/regionserver
export HBASE_MANAGES_ZK=true

```

Save it.

5. If cluster is not running then

```

Terminal > start-all.sh
Terminal > mr-jobhistory-daemon.sh start historyserver

```

HBase start:

```
Terminal > start-hbase.sh
```

JPS must show:

```

HRegionServer
HMaster
HQuorumPeer

```

3 of hbase+6 of hadoop = 9 services running..

```
Terminal > hbase shell
```

If error about zookeeper or hbase master and regionserver

```

Terminal > sudo service hbase-master restart
Terminal > sudo service hbase-regionserver restart

```

Master --> Slave

Master with hadoop and all setup --->

Master IP: Terminal > ifconfig --> 192.168.43.37

Slave IP: Terminal > ifconfig --> 192.168.43.206

Optional for Disable firewall

Terminal > service iptables stop

Open host file to add master and data node with their ip address in both machines:

Terminal > sudo gedit /etc/hosts

192.168.43.37 sachin-pc

192.168.43.206 hduser-VirtualBox

Terminal > service sshd restart

If Error:

Terminal > sudo apt-get install openssh-server

Create the ssh key in master:

Copy the key to master authorized keys:

Terminal > chmod 600 .ssh/id_rsa

Copy this key to slave:

sudo ssh-copy-id -i \$HOME/.ssh/id_rsa.pub hduser@hduser-VirtualBox -f

Testing:

Terminal > ssh hduser@hduser-VirtualBox

It suppose to login

Make sure slave is ready with java installation...

Terminal > javac -version

Copy hadoop setups to slave machine...

Unzip hadoop to home directory

Terminal > sudo gedit ~/.bashrc

#Hadoop Setting

```
export HADOOP_PREFIX=/home/hduser/hadoop-2.7.7
export PATH=$PATH:$HADOOP_PREFIX/bin
export PATH=$PATH:$HADOOP_PREFIX/sbin
export HADOOP_MAPRED_HOME=${HADOOP_PREFIX}
export HADOOP_COMMON_HOME=${HADOOP_PREFIX}
export HADOOP_HDFS_HOME=${HADOOP_PREFIX}
export YARN_HOME=${HADOOP_PREFIX}
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_PREFIX/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_PREFIX/lib/native"
```

Save it and source ~/.bashrc

Testing: Terminal > hadoop version

Create masters file and edit as follow in both master and slave machines as below:

```
Terminal > sudo gedit hadoop-2.7.7/etc/hadoop/masters
Add: sachin-pc
```

Create slaves file in Master and Slave machine and edit as follow

```
Terminal > sudo gedit hadoop-2.7.7/etc/hadoop/slaves
Add: hduser-VirtualBox
```

Edit core-site.xml on both machines...

```
sudo gedit hadoop-2.7.7/etc/hadoop/core-site.xml
```

Add following:

```
<property>
<name>fs.default.name</name>
<value>hdfs://sachin-pc:9000</value>
</property>
```

Edit hdfs-site.xml on master as below:

```
Terminal > sudo gedit hadoop-2.7.7/etc/hadoop/hdfs-site.xml
```

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>2</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>/home/sachin/hadoop-2.7.7/namenode</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>/home/sachin/hadoop-2.7.7/datanode</value>
  </property>
  <property>
    <name>dfs.permissions</name>
    <value>>false</value>
  </property>
</configuration>
```