MULTI-NODE HBASE CLUSTER ON HADOOP 2.6.0

TABLE OF CONTENTS

SETUP DETAILS:

Create 4 separate machines i.e., 1master and 3slaves with defined IP addresses
master 192.168.10.10
slave1 192.168.10.11
slave2 192.168.10.12
slave3 192.168.10.13

STEP 1: INSTALL JDK7

Before installing hadoop make sure you have java installed on all nodes of hadoop cluster systems.

Download JDK7 for Linux-x64 from official Oracle site.

```
[root@master]# cd ~/Download
[root@master]# yum localinstall jdk-7u80-linux-x64.rpm
[root@master]# alternatives --install /usr/bin/java java /usr/java/jdk1.7.0_80/bin/java 210000
```

To check java version and also alternatives

```
[root@master]# java –version
[root@master]# alternatives --display java
```

This is need to done all the 4 machines.

STEP 2: CREATE USER ACCOUNT

Create a system user account on both master and slave systems to use for hadoop installation

```
[root@master]# useradd huser
[root@master]# passwd huser
```

STEP 3: ADD FQDN MAPPING

Edit /etc/hosts file on master and slave machines and add following entries.

```
[root@master]# gedit /etc/hosts
```

Append the following lines at the end of the file:

```
192.168.10.10 master
192.168.10.11 slave1
192.168.10.12 slave2
192.168.10.13 slave3
```

STEP 4: CONFIGURING KEY BASED LOGIN

It’s required to set up hadoop user to ssh itself without password. Use following commands to configure auto login between all hadoop cluster servers.

```
[root@master]# su – huser
[root@huser]$ ssh-keygen
[root@huser]$ ssh-copy-id -i ~/.ssh/id_rsa.pub huser@192.168.10.10
[root@huser]$ ssh-copy-id -i ~/.ssh/id_rsa.pub huser@192.168.10.11
[root@huser]$ ssh-copy-id -i ~/.ssh/id_rsa.pub huser@192.168.10.12
[root@huser]$ ssh-copy-id -i ~/.ssh/id_rsa.pub huser@192.168.10.13
 [root@huser]$ chmod 0600 ~/.ssh/authorized_keys
[root@huser]$ exit
```

To avoid typing password for each time we login:

```
[root@master]# gedit /etc/ssh/ssh_config
```

And search for “StrickHostKeyChecking”
Remove “#” and make it like this “StrickHostKeyChecking no” without double quote and save it.

STEP 5: DOWNLOAD AND EXTRACT HADOOP SOURCE

Download Hadoop

```
[root@master]# cd ~/Downloads
[root@master]# wget http://www.eu.apache.org/dist/hadoop/common/hadoop-2.6.0/hadoop-2.6.0.tar.gz
[root@master]# mkdir /opt/Hadoop
[root@master]# cp ~/Downloads/hadoop-2.6.0.tar.gz /opt/hadoop
[root@master]# cd /opt/hadoop/
[root@master]# tar -xzf hadoop-2.6.0.tar.gz
[root@master]# chown -R huser /opt/hadoop
[root@master]# cd /opt/hadoop/hadoop-2.6.0/
```

Download Hbase

```
[root@master]# cd ~/Downloads
[root@master]# wget http://www-us.apache.org/dist/hbase/0.98.18/hbase-0.98.18-hadoop2-bin.tar.gz
[root@master]# mkdir /opt/hadoop/hbase
[root@master]# cp ~/Downloads/hbase-0.98.18-hadoop2-bin.tar.gz /opt/hadoop/hbase
[root@master]# cd /opt/hadoop/hbase
[root@master]# tar -xzf hbase-0.98.18-hadoop2-bin.tar.gz
[root@master]# cd /opt/hadoop/hbase/
```

Download Hive

```
[root@master]# cd ~/Downloads
[root@master]# wget http://mirror.tcpdiag.net/apache/hive/stable/apache-hive-1.2.0-bin.tar.gz
[root@master]# mkdir /opt/hadoop/hive-1.2.0
[root@master]# cp ~/Downloads/apache-hive-1.2.0-bin.tar.gz /opt/hadoop/hive-1.2.0
[root@master]# cd /opt/hadoop/hive-1.2.0
[root@master]# tar –xzf apache-hive-1.2.0-bin.tar.gz
```

STEP 6: CONFIGURE HADOOP

Edit hadoop configuration files and make following changes.

```
[root@master]# cd /opt/hadoop/hadoop-2.6.0/etc/hadoop/
```

6.1 - Edit core-site.xml

```
[root@master]# core-site.xml
```

Add the following inside the <configuration> tag

```
<configuration>
<property>
        <name>fs.defaultFS</name>
        <value>hdfs://master:9000/</value>
</property>
</configuration>
```

6.2 - Create Datanode and Namenode

Create HDFS DataNode data dirs on every node and change ownership of /opt/hadoop:

```
[root@master]# chown huser /opt/hadoop/ -R
[root@master]# chgrp huser /opt/hadoop/ -R
[root@master]# mkdir /opt/hadoop/datanode
[root@master]# chown huser /opt/hadoop/hdfs/datanode/
[root@master]# chgrp huser /opt/hadoop/hdfs/datanode/
```

Create HDFS NameNode data dirs on master:

```
[root@master]# mkdir /opt/hadoop/namenode
[root@master]# chown huser /opt/hadoop/hdfs/namenode/
[root@master]# chgrp huser /opt/hadoop/hdfs/namenode/
```

6.3 - Edit hdfs-site.xml

```
[root@master]# gedit hdfs-site.xml
```

Add the following inside the <configuration> tag

```
<configuration>
<property>
        <name>dfs.replication</name>
        <value>3</value>
</property>
<property>
        <name>dfs.permissions</name>
        <value>false</value>
</property>
<property>
        <name>dfs.datanode.data.dir</name>
        <value>/opt/hadoop/hdfs/datanode</value>
</property>
<property>
         <name>dfs.namenode.data.dir</name>
         <value>/opt/hadoop/hdfs/namenode</value>
</property>
</configuration>
```

6.4 - Edit mapred-site.xml

```
[root@master]# gedit mapred-site.xml
```

Add the following inside the <configuration> tag

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

6.5 - Edit yarn-site.xml

```
[root@master]# gedit yarn-site.xml
```

Add the following inside the <configuration> tag

```
<configuration>
<property>
        <name>yarn.resourcemanager.hostname</name>
        <value>master</value>
</property>



<property>
        <name>yarn.nodemanager.hostname</name>
        <value>master</value>       <!-- or slave1, slave2, slave3 -->
</property>

<property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
</property>
</configuration>
```

6.6 - Edit hadoop-env.sh

```
[root@master]# gedit hadoop-env.sh
Append the following lines at the end of the file:
export JAVA_HOME=/usr/java/jdk1.7.0_80
export HADOOP_OPTS=-Djava.net.preferIPv4Stack=true
export HADOOP_CONF_DIR=/opt/hadoop/hadoop-2.6.0/etc/hadoop
```

STEP 7: COPY HADOOP SOURCE TO SLAVE SERVERS

After updating above configuration, we need to copy the source files to all slave servers.

```
[root@master]# scp -rp /opt/hadoop slave1:/opt/
[root@master]# scp -rp /opt/hadoop slave2:/opt/
[root@master]# scp -rp /opt/hadoop slave3:/opt/
```

STEP 8: CONFIGURE HADOOP ON MASTER SERVER ONLY

Go to hadoop source folder on huser-master and do following settings.

```
[root@master]# su – huser
[root@huser]$ cd /opt/hadoop/hadoop-2.6.0/
```

```
[root@huser]$ gedit masters
```

And this line:

```
master
```

```
[root@huser]$ gedit slaves
```

Add this lines:

```
slave1
slave2
slave3
slave4
```

STEP 9: SETTING UP THE ENVIRONMENT FOR JAVA AND HADOOP

We need to source the environment files

```
[root@master]# su – huser
[root@huser]$ gedit ~/.bashrc
```

Append the following lines at the end of the file:

```
## JAVA env variables
export JAVA_HOME=/usr/java/jdk1.7.0_80
export PATH=$PATH:$JAVA_HOME/bin
export CLASSPATH=.:$JAVA_HOME/jre/lib:$JAVA_HOME/lib:$JAVA_HOME/lib/tools.jar
```

```
## HADOOP env variables
export HADOOP_HOME=/opt/hadoop/hadoop-2.6.0
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
```

```
##HBASE env variables
export HBASE_HOME=/opt/hadoop/hbase/hbase-0.98.18-hadoop2
export PATH=$HBASE_HOME/bin:$PATH
export HBASE_PID_DIR=/var/hadoop/pids
```

```
##HIVE env variables
export HIVE_HOME=/opt/hadoop/hive-1.2.0/hive
export PATH=$PATH:$HIVE_HOME/bin
```

```
[root@huser]$ source ~/.bashrc
[root@huser]$ exit
```

SCP to the ~/.bashrc to other slave machines

slave1

```
[root@master]# scp -rp /root/huser/.bashrc slave1:~/
[root@master]# ssh slave1
[root@slave1]$ source ~/.bashrc
[root@slave1]$ exit
```

slave2

```
[root@master]# scp -rp /root/huser/.bashrc slave2:~/
[root@master]# ssh slave2
[root@slave2]$ source ~/.bashrc
[root@slave2]$ exit
```

slave3

```
[root@master]# scp -rp /root/huser/.bashrc slave3:~/
[root@master]# ssh slave3
[root@slave3]$ source ~/.bashrc
[root@slave3]$ exit
```

STEP 10: FORMAT THE NODE

Format Name Node on Hadoop Master only

```
[root@master]# su – huser
[root@huser]$  hdfs namenode –format
```

STEP 11: START HADOOP

Enter the following command to start all HADOOP

```
[root@huser]$ start-all.sh
```

STEP 12: CONFIGURING HBASE

12.1 - Open hbase-env.sh and make following changes

```
[root@huser]$ cd /opt/hadoop/hbase/hbase-0.98.18-hadoop2/conf
[root@huser]$ gedit hbase-env.sh
                    export JAVA_HOME=/usr/java/jdk1.7.0_80
                    export HBASE_MANAGES_ZK=true
```

12.2 - Edit hbase-site.xml in Master Node

```
[root@huser]$ cd /opt/hadoop/hbase/hbase-0.98.18-hadoop2/conf
[root@huser]$ gedit hbase-site.xml
                   <configuration>
                   //Here you have to set the path where you want HBase to store its files.
                   <property>
                     <name>hbase.rootdir</name>
                     <value>hdfs://master:9000/hbase</value>
                   </property>

                   //Here you have to set the path where you want HBase to store its built in zookeeper files.
                   <property>
                     <name>hbase.zookeeper.property.dataDir</name>
                     <value>hdfs:///master:9000/zookeeper</value>
                   </property>

                   <property>
                     <name>hbase.cluster.distributed</name>
                     <value>true</value>
                   </property>

                   <property>
                     <name>hbase.zookeeper.quorum</name>
                     <value>slave1,slav2,slave3</value>
                   </property>

                   <property>
                     <name>hbase.zookeeper.property.clientPort</name>
                     <value>2181</value>
                   </property>
                   </configuration>
```

12.3 - Edit hbase-site.xml in Slaves Node

```
[root@huser]$ cd /opt/hadoop/hbase/hbase-0.98.18-hadoop2/conf
[root@huser]$ gedit hbase-site.xml
                <configuration>
                <property>
                  <name>hbase.rootdir</name>
                  <value>hdfs://slave1:9000/hbase</value>
                </property>

                <property>
                  <name>hbase.cluster.distributed</name>
                  <value>true</value>
                </property>
                </configuration>
```

This need to done in other slaves node

12.4 - Edit regionservers file on Master Node

```
[root@huser]$ cd /opt/hadoop/hbase/hbase-0.98.18-hadoop2/conf
[root@huser]$ gedit regionservers
                  master
                  slave1
                  slave2
                  slave3
```

12.5 - Start Hbase server

```
[root@huser]$ start-hbase.sh
```

Open web browser and type the following line
http://master:60010

STEP 13: TO START HIVE CLI, ENTER THE FOLLOWING COMMANDS

```
[root@huser]$ hadoop fs -mkdir /tmp
[root@huser]$ hadoop fs -mkdir /user/hive/warehouse
[root@huser]$ hadoop fs -chmod g+w /tmp
[root@huser]$ hadoop fs -chmod g+w /user/hive/warehouse
[root@huser]$ hive
```

STEP 14: CREATE DATABSE AND TABLE IN HIVE

14.1 - Create Internal Table

```
hive> CREATE DATABASE emp_db;
hive> USE emp_db;
hive> CREATE TABLE employee (e_id INT, e_name STRING)
        COMMENT ‘Employee Details’
        ROW FORMATE DELIMITED
        FIELDS TERMINATED BY ‘,’
        LINES TERMINATED BY ‘n’
        LOCATION ‘/opt/hadoop/hive/data/employee’;
hive> exit
```

14.2 - Create a .txt file to load the data in Hive

```
[root@huser]$ cd /opt/hadoop/hive/data/employee
[root@huser employee]$ gedit emp.txt
                              101    Ram
                              102    Lakshman
                              103    Krishna
                              104    Arjun
```

14.3 - Load the Data in Table

```
[root@huser employee]$ hive
hive> LOAD DATA ‘hdfs:/opt/hadoop/hive/data/employee/emp.txt’ OVERWRITE INTO TABLE
        employee;
```

14.4 - Viewing table

```
hive> CREATE VIEW AS emp_data
        SELECT * FROM employee
        WHERE e_id>0;

hive> SELECT * FROM emp_data;
```