PIG – HIVE – OOZIE
ON
HADOOP 2.6.0

TABLE OF CONTENTS

SETUP DETAILS:

Create 5 separate machines i.e., 1master and 5slaves with defined IP addresses
master 192.168.10.10
slave1 192.168.10.11
slave2 192.168.10.12
slave3 192.168.10.13
slave4 192.168.10.14

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
192.168.10.14 slave4
```

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]$ ssh-copy-id -i ~/.ssh/id_rsa.pub huser@192.168.10.14
[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 2.6.0

```
[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 Pig 0.15.0

```
[root@master]# cd ~/Downloads
[root@master]# wget http://apache.cs.utah.edu/pig/pig-0.15.0/pig-0.15.0.tar.gz
[root@master]# mkdir /opt/hadoop/pig
[root@master]# cp ~/Downloads/pig-0.15.0.tar.gz /opt/hadoop/pig
[root@master]# cd /opt/hadoop/pig
[root@master]# tar –xzf pig-0.15.0.tar.gz
```

Download Hive 1.2.0

```
[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/datanode/
[root@master]# chgrp huser /opt/hadoop/datanode/
```

Create HDFS NameNode data dirs on master:

```
[root@master]# mkdir /opt/hadoop/namenode
[root@master]# chown huser /opt/hadoop/namenode/
[root@master]# chgrp huser /opt/hadoop/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>4</value>
</property>
<property>
        <name>dfs.permissions</name>
        <value>false</value>
</property>
<property>
        <name>dfs.datanode.data.dir</name>
        <value>/opt/hadoop/datanode</value>
</property>
<property>
         <name>dfs.namenode.data.dir</name>
         <value>/opt/hadoop/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, slave4 -->
</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/
[root@master]# scp -rp /opt/hadoop slave4:/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, HADOOP, PIG AND HIVE

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
```

```
##PIG env variables
export PIG_HOME/opt/hadoop/pig/pig-0.15.0
export PATH=$PIG_HOME/bin:$PATH
```

```
##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]$ echo $HADOOP_HOME
[root@huser]$ echo $JAVA_HOME
[root@huser]$ exit
```

SCP to the ~/.bashrcto 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 slave1
[root@slave2]$ source ~/.bashrc
[root@slave2]$ exit
```

slave3

```
[root@master]# scp -rp /root/huser/.bashrc slave3:~/
[root@master]# ssh slave1
[root@slave3]$ source ~/.bashrc
[root@slave3]$ exit
```

slave4

```
[root@master]# scp -rp /root/huser/.bashrc slave4:~/
[root@master]# ssh slave4
[root@slave1]$ source ~/.bashrc
[root@slave1]$ 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, PIG, HIVE SERVICES

Enter the following command to start all HADOOP

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

Enter the following command to start PIG

```
[root@huser]$ pif -x
```

Before starting 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 12: CHECK RUNNING SERVICES

```
[root@huser]$ jps
```

Open browser and type on address bar “master:50070” without double quote and u can see 4 live nodes

STEP 13: OOZIE INSTALLATION

13.1 - OOZIE tarball extraction

```
[root@huser]$ su
[root@master]# cd /Download
[root@master]# wget http://apache.bytenet.in/oozie/4.1.0/oozie-4.1.0.tar.gz
[root@master]# mkdir -p /usr/huser/setups/oozie
[root@master]# cp oozie-4.1.0.tar.gz /usr/hduser/setups/oozie
[root@master]# cd /usr/huser/setups/oozie
[root@master]# tar –xzf oozie-4.1.0.tar.gz
```

13.2 - Maven installation

```
[root@master]# apt-get update
[root@master]# apt-get install maven
```

13.3 - OOZIE-Hadoop version configuration

```
[root@master]# cd oozie-4.1.0
[root@master]# gedit pom.xml
```

--Search for

```
<hadoop.version>1.1.1</hadoop.version>
```

--Replace it with

```
<hadoop.version>2.6.0</hadoop.version>
```

save pom.xml file

13.4 OOZIE Package Building

```
[root@master]# mvn clean package assembly:single -P hadoop-2 -DskipTests
[root@master]# cd /usr/local
[root@master]# mkdir oozie
[root@master]# cp -rf /usr/huser/setups/oozie/oozie-4.1.0/distro/target/oozie-4.1.0-distro/oozie-4.1.0/ oozie/
[root@master]# cd oozie/oozie-4.1.0
[root@master]# mkdir libext
[root@master]# cp -R /usr/huser/setups/oozie/oozie-4.1.0/hadooplibs/Hadoop-2/target/hadooplibs/hadooplib-2.3.0.oozie-4.1.0/* libext
[root@master]# wget -P libext http://dev.sencha.com/deploy/ext-2.2.zip
```

13.5 - OOZIE War Building

```
[root@master]# ./bin/oozie-setup.sh prepare-war
```

13.6 OOZIE-Hadoop configuration

```
[root@master]# gedit /opt/hadoop/hadoop-2.6.0/etc/hadoop/core-site.xml
```

And the following content in end of core-site.xml

```
<configuration>
<property>
        <name>hadoop.proxyuser.huser.hosts</name>
        <value>*</value>
</property>

<property>
        <name>hadoop.proxyuser.huser.groups</name>
        <value>*</value>
</property>
</configuration>
```

13.7 - Changing oozie directory owner and group

```
[root@master]# cd /usr/local
[root@master]# chown -R huser:huser oozie
```

13.8 - Creating Sharelib directory in HDFS

```
[root@master]# su huser
[root@huser]$ cd /usr/local/oozie/oozie-4.1.0
[root@huser]$ ./bin/oozie-setup.sh sharelib create -fs hdfs://localhost:54310
```

13.9 - Creating oozie database

```
[root@huser]$ ./bin/ooziedb.sh create -sqlfile oozie.sql -run
```

13.10 - Starting oozie server

```
[root@huser]$ ./bin/oozied.sh start
```

13.11 - Verify whether oozie server is up and running

```
[root@huser]$ http://localhost:11000/oozie/
```

13.12 - Stopping oozie server

```
[root@huser]$ ./bin/oozied.sh stop
```