######

Master: 172.29.75.250

Slave: 172.29.75.249, 172.29.75.251

Salve to be added later: 172.29.75.248

######

###########Install Java on each server###########################

sudo yum install java-1.8.0-openjdk java-1.8.0-openjdk-devel -y

vi ~/.bashrc

export JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk

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

source ~/.bashrc

#Check

java -version

########################### Install Scala on each server###########################

cd /opt/

sudo wget https://downloads.lightbend.com/scala/2.12.4/scala-2.12.4.tgz

sudo tar xvf scala-\*.tgz

whereis scala

vi ~/.bashrc

#use the path obtained above

export SCALA\_HOME=/opt/scala-2.12.4

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

source ~/.bashrc

#Check

scala -version

###Install requirements

sudo yum install bzip2 -y

########################### Install Python 3 (Anaconda) on each server###########################

cd /opt/

sudo wget https://repo.continuum.io/archive/Anaconda3-5.0.1-Linux-x86\_64.sh

bash Anaconda\*.sh

#Keep pressing enter and yes and path give as "/opt/anaconda3"

#Do you wish the installer to prepend the Anaconda3 install location to PATH in your /root/.bashrc ? [yes|no] : Select yes

source ~/.bashrc

#check this will show the Anaconda version of python 3.5

python

########################## Install MongoDB on master server/name node###########################

cd /opt/

curl -O https://fastdl.mongodb.org/linux/mongodb-linux-x86\_64-3.6.5.tgz

tar -zxvf mongodb-linux-x86\_64-3.6.5.tgz

#check path from

whereis mongo

vi ~/.bashrc

export PATH=/opt/mongodb-linux-x86\_64-3.6.5/bin:$PATH

source ~/.bashrc

mkdir -p /data/db

chmod -R 777 mongodb

#verify installation by typing mongod in the terminal. Keep this running.

mongod

#from another terminal write mongo and check

mongo

>show databases;

>exit;

######################### Setting up multi-node spark cluster on each server#############################

#On each server do the following

sudo vi /etc/hosts

172.29.75.250 master

172.29.75.249 slave01

172.29.75.251 slave02

Esc :wq

############ Setting up passwordles access from each of the servers : taking help from sysadmin ##########

#somewhat like

sudo yum install openssh-server openssh-client -y

##### From master 104 to slaves 108

su - tilguest

ssh-keygen -t rsa

ssh-copy-id -i ~/.ssh/id\_rsa.pub tilguest@master

ssh-copy-id -i ~/.ssh/id\_rsa.pub tilguest@slave01

chmod 0600 ~/.ssh/authorized\_keys

chmod 755 ~/.ssh

# Check bidirectional communication

ssh tilguest@slave01

ssh tilguest@master

########################## Install Hadoop on each server###########################

cd /opt/

sudo wget http://redrockdigimark.com/apachemirror/hadoop/common/hadoop-2.7.6/hadoop-2.7.6.tar.gz

tar -xzvf hadoop\*.tar.gz

#Reference: https://www.digitalocean.com/community/tutorials/how-to-install-hadoop-in-stand-alone-mode-on-ubuntu-16-04

whereis hadoop

vi ~/.bashrc

HADOOP\_HOME=/opt/hadoop-2.7.6

PATH=$PATH:$HADOOP\_HOME/bin

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

export PATH=$PATH:$HADOOP\_HOME/sbin

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\_PID\_DIR=$HADOOP\_HOME/pid/

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

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

export HADOOP\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop

export YARN\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop

export HDFS\_NAMENODE\_USER="root"

export HDFS\_DATANODE\_USER="root"

export HDFS\_SECONDARYNAMENODE\_USER="root"

export YARN\_RESOURCEMANAGER\_USER="root"

export YARN\_NODEMANAGER\_USER="root"

source ~/.bashrc

#check by typing hadoop on the terminal

###################### Setting up Hadoop multi-node cluster : on each server ###########################

#Reference: https://data-flair.training/blogs/hadoop-2-6-multinode-cluster-setup/

mkdir -p /data/hadoop\_data/namenode

chmod -R 777 /data/hadoop\_data/namenode

mkdir -p /data/hadoop\_data/datanode

chmod -R 777 /data/hadoop\_data/datanode

-----------------------------

cd /opt/hadoop-2.7.6/etc/hadoop/

-----------------------------

vi /opt/hadoop-2.7.6/etc/hadoop/hadoop-env.sh

export JAVA\_HOME=${JAVA\_HOME}

#make sure JAVA\_HOME is set

#OR : export JAVA\_HOME=<path-to-the-root-of-your-Java-installation> (eg: /usr/lib/jvm/java-8-oracle/

#OR : change ~/.bashrc accordingly

Esc : wq

------------------------------

vi /opt/hadoop-2.7.6/etc/hadoop/core-site.xml

#Add the following lines

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://master:9000</value>

</property>

<property>

<name>hadoop.tmp.dir</name>

<value>/tmp/</value>

</property>

</configuration>

Esc :wq

-------------------------

vi /opt/hadoop-2.7.6/etc/hadoop/hdfs-site.xml

#Add the following lines

<configuration>

<property>

     <name>dfs.replication</name>

     <value>1</value>

</property>

<property>

     <name>dfs.name.dir</name>

     <value>file:///data/hadoop\_data/namenode</value>

</property>

<property>

     <name>dfs.data.dir</name>

     <value>file:///data/hadoop\_data/datanode </value>

</property>

</configuration>

Esc : wq

------------------------------

cp /opt/hadoop-2.7.6/etc/hadoop/mapred-site.xml.template /opt/hadoop-2.7.6/etc/hadoop/mapred-site.xml

vi /opt/hadoop-2.7.6/etc/hadoop/mapred-site.xml

<configuration>

<property>

     <name>mapreduce.framework.name</name>

     <value>yarn</value>

  </property>

</configuration>

Esc : wq

------------------------------

vi /opt/hadoop-2.7.6/etc/hadoop/yarn-site.xml

<configuration>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

<property>

<name>yarn.resourcemanager.resource-tracker.address</name>

<value>master:8025</value>

</property>

<property>

<name>yarn.resourcemanager.scheduler.address</name>

<value>master:8030</value>

</property>

<property>

<name>yarn.resourcemanager.address</name>

<value>master:8040</value>

</property>

</configuration>

Esc : wq

----------------------------

vi /opt/hadoop-2.7.6/etc/hadoop/slaves

#Add the following lines

salve01

slave02

Esc : wq

################## Start Hadoop from Master/name node ################

#Format namenode from master/namenode

/opt/hadoop-2.7.6/bin/hdfs namenode -format

#Start dfs from master/namenode

/opt/hadoop-2.7.6/sbin/start-dfs.sh

#Start dfs from master/namenode

/opt/hadoop-2.7.6/sbin/start-yarn.sh

############### Stop Hadoop from Master/name node ##################

#Stop dfs from master/namenode

/opt/hadoop-2.7.6/sbin/stop-dfs.sh

#Stop dfs from master/namenode

/opt/hadoop-2.7.6/sbin/stop-yarn.sh

################Check if working properly from master/namenode

jps

#Output

15377 NameNode

16001 ResourceManager

15640 SecondaryNameNode

16303 Jps

###############Check if working properly from datanodes/slaves

jps

#Output

5081 NodeManager

5241 Jps

4926 DataNode

########################## Install Spark on each server###########################

cd /opt/

sudo wget http://www-eu.apache.org/dist/spark/spark-2.3.0/spark-2.3.0-bin-hadoop2.7.tgz

tar xvf spark\*.tgz

#check path from

whereis pyspark

vi ~/.bashrc

export SPARK\_HOME=/opt/spark-2.3.0-bin-hadoop2.7/

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

source ~/.bashrc

#type pyspark and verify

################## Making spark run on multiple nodes on each servers/nodes##############################

mkdir /data/spark\_logs

cd /opt/spark-2.3.0-bin-hadoop2.7/conf/

cp /opt/spark-2.3.0-bin-hadoop2.7/conf/spark-env.sh.template /opt/spark-2.3.0-bin-hadoop2.7/conf/spark-env.sh

vi /opt/spark-2.3.0-bin-hadoop2.7/conf/spark-env.sh

export SCALA\_HOME=/opt/scala-2.12.4

export SPARK\_WORKER\_MEMORY=30g

export SPARK\_WORKER\_INSTANCES=1

export SPARK\_WORKER\_DIR=/opt/spark-2.3.0-bin-hadoop2.7/

export SPARK\_MASTER\_IP=master

export SPARK\_LOCAL\_DIRS=/data/spark\_logs/

Esc : wq

-----------------------------------------------

cp /opt/spark-2.3.0-bin-hadoop2.7/conf/slaves.template /opt/spark-2.3.0-bin-hadoop2.7/conf/slaves

vi /opt/spark-2.3.0-bin-hadoop2.7/conf/slaves

master

slave01

slave02

Esc: wq

---------------------------------

cp /opt/spark-2.3.0-bin-hadoop2.7/conf/spark-defaults.conf.template /opt/spark-2.3.0-bin-hadoop2.7/conf/spark-defaults.conf

vi /opt/spark-2.3.0-bin-hadoop2.7/conf/spark-defaults.conf

spark.master                     spark://master:7077

Esc: wq

----------------------------------

########################### Start spark from master/namenode #######################

###On namenode/master

cd /opt/spark-2.3.0-bin-hadoop2.7/sbin/

/opt/spark-2.3.0-bin-hadoop2.7/sbin/start-all.sh

########################### Stop spark from master/namenode #######################

###On namenode/master

cd /opt/spark-2.3.0-bin-hadoop2.7/sbin/

/opt/spark-2.3.0-bin-hadoop2.7/sbin/stop-all.sh

################Check if working properly from master/namenode

jps

#Output

17329 SecondaryNameNode

17762 ResourceManager

18762 Jps

17070 NameNode

18591 Master

#Note : Here Master is added by spark

###############Check if working properly from datanodes/slaves

jps

#Output

5603 NodeManager

6084 Worker

6167 Jps

5438 DataNode

#Note : Here Worker is added by spark

######Check from browser http://172.29.75.250:8080/ ##########

########################## Install Sqoop on master server###########################

#Download sqoop from : https://sqoop.apache.org/

cd /opt/

sudo wget http://redrockdigimark.com/apachemirror/sqoop/1.4.7/sqoop-1.4.7.bin\_\_hadoop-2.6.0.tar.gz

tar -xvf /opt/sqoop-1.4.7.bin\_\_hadoop-2.6.0.tar.gz

vi ~/.bashrc

export SQOOP\_HOME=/opt/sqoop-1.4.7.bin\_\_hadoop-2.6.0/

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

source ~/.bashrc

mv /opt/sqoop-1.4.7.bin\_\_hadoop-2.6.0/conf/sqoop-env-template.sh /opt/sqoop-1.4.7.bin\_\_hadoop-2.6.0/conf/sqoop-env.sh

vi /opt/sqoop-1.4.7.bin\_\_hadoop-2.6.0/conf/sqoop-env.sh

export HADOOP\_COMMON\_HOME=/opt/hadoop-2.7.6

export HADOOP\_MAPRED\_HOME=/opt/hadoop-2.7.6

Esc : wq

#Check

sqoop-version

#### Download and Configure db2-connector-java

#download and put db2 jar in /opt/sqoop-1.4.7.bin\_\_hadoop-2.6.0/lib

#Reference: https://www.tutorialspoint.com/sqoop/sqoop\_installation.htm , https://data-flair.training/blogs/sqoop-installation/

sqoop import   --connect jdbc:db2://172.16.84.205:60000/tjcandb   --username tcuser --password jobusr --query "select emailid from tcuser.FAKE\_EMAIL where \$CONDITIONS and status = 'N'"   --m 1 --target-dir /fake\_emails

#Error: ERROR tool.ImportTool: Import failed: org.apache.hadoop.ipc.RemoteException(java.io.IOException): File /tmp/hadoop-yarn/staging/root/.staging/job\_1529505610700\_0001/libjars/parquet-generator-1.6.0.jar could only be replicated to 0 nodes instead of minReplication (=1).  There are 0 datanode(s) running and no node(s) are excluded in this operation.