# SQOOP AND MYSQL
# ON
# HADOOP 2.6.0

# TABLE OF CONTENTS

SETUP DETAILS: 4
STEP 1: INSTALL JDK7 4
STEP 2: CREATE USER ACCOUNT 4
STEP 3: ADD FQDN MAPPING 5
STEP 4: CONFIGURING KEY BASED LOGIN 5
STEP 5: DOWNLOAD AND INSTALLATION 6
Download Hadoop 2.6.0 6
Download SQOOP 1.4.6 6
Download and Install MySQL 7
STEP 6: CONFIGURE HADOOP 8
6.1 - Edit core-site.xml 8
6.2 - Create Datanode and Namenode 9
6.3 - Edit hdfs-site.xml 9
6.4 Edit mapred-site.xml 10
6.5 Edit yarn-site.xml 10
6.6 Edit hadoop-env.sh 11
STEP 7: COPY HADOOP SOURCE TO SLAVE SERVERS 11
STEP 8: CONFIGURE HADOOP ON MASTER SERVER ONLY 12
STEP 9: SETTING UP THE ENVIRONMENT FOR JAVA, HADOOP AND SQOOP 12

STEP 10: CONFIGURE SQOOP ....................14
10.1 - Download and Configure mysql-connector-java .................... 14
STEP 11: FORMAT THE NODE ....................15
STEP 12: START HADOOP ....................15
STEP 13: CHECK RUNNING SERVICES ....................15
STEP 14: CREATE A DATABASE, TABLE AND INSERT SOME VALUES ....................15
STEP 15: SQOOP IMOPRT....................16
15.1 - Importing a table into HDFS .................... 16
15.2 Import all rows of a table in MySQL, but specific columns of the table.................... 17
15.3 Import all columns, filter rows using where clause.................... 17

## SETUP DETAILS:

Create 5 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 INSTALLATION

### 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 SQOOP 1.4.6

```
[root@master]# cd ~/Downloads
[root@master]# wget http://apache.proserve.nl/sqoop/1.4.6/sqoop-1.4.6.bin__hadoop-2.0.4-alpha.tar.gz
[root@master]# tar -xzf sqoop-1.4.6.bin__hadoop-2.0.4-alpha.tar.gz
[root@master]# mv sqoop-1.4.6.bin__hadoop-2.0.4-alpha sqoop-1.4.6
[root@master]# mkdir /opt/hadoop/sqoop
[root@master]# cp ~/Downloads/ sqoop-1.4.6 /opt/hadoop/sqoop
```

## Download and Install MySQL

```
[root@master]# cd ~/Downloads
```

Adding the MySQL Yum Repository

```
[root@master]# wget http://dev.mysql.com/get/mysql57-community-release-el6-7.noarch.rpm
```

Installing downloaded package

```
[root@master]# yum localinstall mysql57-community-release-el6-7.noarch.rpm
```

Installing MySQL

```
[root@master]# yum install mysql-community-server
```

Installing MySQL Release Series

```
[root@master]# yum-config-manager --disable mysql57-community
[root@master]# yum-config-manager --enable mysql56-community
```

Starting the MySQL Server

```
[root@master]# service mysqld start
```

Verifying the status of the MySQL server

```
[root@master]# service mysqld status
```

Verifying installed MySQL version

```
[root@master]# mysql –version
```

Securing the MySQL installation

below command to see the password before running mysql secure command

```
[root@master]# grep 'temporary password' /var/log/mysqld.log
```

Once you know the password you can now run following command to secure your MySQL installation

```
[root@master]# mysql_secure_installation

Connecting to MySQL Server
[root@master]# mysql -u root -p

Updating MySQL
[root@master]# yum update mysql-server
```

## 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>3</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>
<property>
            <name>dfs.nameservices</name>
            <value>ns1, ns2 </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, HADOOP AND SQOOP

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
## SQOOP env variables
export SQOOP_HOME=/opt/hadoop/sqoop
export PATH=$PATH:$SQOOP_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 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
```

## STEP 10: CONFIGURE SQOOP

```
[root@master]# cd $SQOOP_HOME/conf
[root@master]# mv sqoop-env-template.sh sqoop-env.sh

Open sqoop-env.sh and edit the following lines
[root@master]# gedit sqoop-env.sh
export HADOOP_COMMON_HOME=/opt/hadoop/hadoop-2.6.0/
export HADOOP_MAPRED_HOME=/opt/hadoop/hadoop-2.6.0/
```

### 10.1 - Download and Configure mysql-connector-java

```
[root@master]# cd ~/Download
[root@master]# wget http://ftp.ntu.edu.tw/MySQL/Downloads/Connector-J/mysql-connector-java-5.1.40.tar.gz
[root@master]# tar –xzf mysql-connector-java-5.1.40.tar.gz
[root@master]# cd mysql-connector-java-5.1.40
[root@master]# mv mysql-connector-java-5.1.40-bin.jar /opt/hadooop/sqoop/lib
[root@master]# cd $SQOOP_HOME/bin
[root@master]# sqoop-version
```

## STEP 11: FORMAT THE NODE

Format Name Node on Hadoop Master only

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

## STEP 12: START HADOOP

Enter the following command to start all HADOOP

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

## STEP 13: CHECK RUNNING SERVICES

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

Open browse and type on address bar “master:50070” without double quote and u can see 3 live nodes

## STEP 14: CREATE A DATABASE, TABLE AND INSERT SOME VALUES

```
[root@huser]$ mysql -u root –p
        mysql> CREATE DATABASE test;
        mysql> USE test;
        mysql> CREATE TABLE student (s_id INT, s_name VARCHAR(20));
        mysql> INSERT INTO student (s_id, s_name) VALUES (101, “Ram”);
        mysql> INSERT INTO student (s_id, s_name) VALUES (102, “Sita”);
        mysql> INSERT INTO student (s_id, s_name) VALUES (103, “Lakshman”);
        mysql> INSERT INTO student (s_id, s_name) VALUES (104, “Krishna”);
        mysql> INSERT INTO student (s_id, s_name) VALUES (105, “Arjun”);
        mysql> SELECT * FROM student;
        mysql> exit;
```

## STEP 15: SQOOP IMOPRT

### 15.1 - Importing a table into HDFS

```
[root@huser]$ cd $HOME
```

Create a config file $HOME/import.txt add following to the config file

```
[root@huser]$ gedit import.txt
        import
        --connect
        jdbc:mysql://localhost/test
        --username
        root
        --password
        1212
```

Execute the sqoop import

```
[root@huser]$ sqoop --options-file /home/huser/import.txt --table student -m 1
```

Once import is done you can find student.jar, student.class and student.java at following location /tmp/sqoop-huser/compile/—-/student.jar

Files created in HDFS

```
[root@huser]$ hadoop dfs -ls -R student
```

Data file contents

```
[root@huser]$ hadoop dfs -cat /user/huser/student/part-m-00000
101,Ram
102,Sita
103,Lakshman
104,Krishna
105,Arjun
```

### 15.2 Import all rows of a table in MySQL, but specific columns of the table

```
[root@huser]$ sqoop import --connect jdbc:mysql://localhost/test --username root --password 1212 --table student --columns "s_name" -m 1

Data file contents
[root@huser]$ hadoop dfs -cat  /user/huser/student/part-m-00000
Ram
Sita
Lakshman
Krishna
Arjun
```

### 15.3 Import all columns, filter rows using where clause

```
[root@huser]$ sqoop import --connect jdbc:mysql://localhost/test --username root --password 1212 --table student --where "s_id>101" -m 1 --target-dir
/user/huser/ar

Data file contents
[root@huser]$ hadoop dfs -cat  /user/huser/ar/part-m-00000
102,Sita
```