usernameusedforlogin ALL=(ALL) NOPASSWD:ALL

Step 1: Download Hadoop

```
wget https://archive.apache.org/dist/hadoop/common/hadoop-3.2.1/hadoop-3.2.1.tar.gz
```

Step 2: Untar the tar file

```
tar -xvzf hadoop-3.2.1.tar.gz
mv hadoop-3.2.1 hadoop
sudo apt-get update -y
sudo apt install openjdk-8-jdk -y
```

Step 3: Update the .bashrc file

```
gedit/vi .bashrc
```

Step 4: Add the Hadoop configurations and save

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_HOME=/home/bigdata/hadoop
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export HADOOP_YARN_HOME=$HADOOP_HOME
export
HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS="-
Djava.library.path=$HADOOP_HOME/lib/native"
```

Step 5: Execute the following command

```
source .bashrc
```

Step 6: edit the Hadoop env file to set the java path

```
gedit $HADOOP_HOME/etc/hadoop/hadoop-env.sh
add
```

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

```
ssh-keygen -t rsa
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
chmod 640 ~/.ssh/authorized_keys
```

```
sudo apt-get remove openssh-client openssh-server -y

sudo apt-get install openssh-client openssh-server -y
```

Step 7: Create Hadoop directories

```
mkdir -p ~/hadoopdata/hdfs/namenode
mkdir -p ~/hadoopdata/hdfs/datanode
chmod 777 -R hadoopdata
```

Step 8: Edit /home/bigdata/hadoop/etc/hadoop/core-site.xml and add the following

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

Step 9: Edit /home/bigdata/hadoop/etc/hadoop/hdfs-site.xml and add the following

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

    <property>
        <name>dfs.name.dir</name>
        <value>file:///home/bigdata/hadoopdata/hdfs/namenode
</value>
    </property>

    <property>
        <name>dfs.data.dir</name>
        <value>file:///home/bigdata/hadoopdata/hdfs/datanode
</value>
    </property>
```

Step 10: Edit /home/spark/hadoop/etc/hadoop/mapred-site.xml and add the following

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

<property>
<name>yarn.app.mapreduce.am.env</name>
<value>HADOOP_MAPRED_HOME=${HADOOP_HOME}</value>
</property>
<property>
<name>mapreduce.map.env</name>
<value>HADOOP_MAPRED_HOME=${HADOOP_HOME}</value>
</property>
<property>
```

```
<name>mapreduce.reduce.env</name>
<value>HADOOP_MAPRED_HOME=${HADOOP_HOME}</value>
</property>
```

Step 11: Edit /home/spark/hadoop/etc/hadoop/yarn-site.xml and add the following

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

Step 12: Format name node

```
hdfs namenode -format
```

Step 13: Start hadoop

```
start-all.sh
```

Step 14: Adjust firewall

```
sudo firewall-cmd --permanent --add-port=9870/tcp
sudo firewall-cmd --permanent --add-port=8088/tcp
sudo firewall-cmd --reload
```

Step 15: Access NN and YARN

http://localhost:9870 --- Name Node
http://localhost:8088 -- Resource Manager

Step 16: Create hdfs directory for current user

```
hdfs dfs -mkdir -p /user/bigdata
```

Install mysql server

$ sudo apt update
$ sudo apt upgrade -y
$ sudo apt install mysql-server -y
$ sudo mysql_secure_installation
Set password and answer Y to all questions

$ sudo systemctl status mysql
$ sudo systemctl enable mysql

$ sudo mysql -u root

Download and copy the mysql driver to hive /lib directory

wget https://repo1.maven.org/maven2/mysql/mysql-connector-java/8.0.22/mysql-connector-java-8.0.22.jar

sudo mysql -uroot -e "create user 'training'@'%' identified by 'training'";
sudo mysql -uroot -e "create database moviesdb";
sudo mysql -uroot -e "grant all privileges on moviesdb.* to 'training'@'%'";

sudo mysql -u root moviesdb < moviesdb.sql;

Install Sqoop

wget http://archive.apache.org/dist/sqoop/1.4.7/sqoop-1.4.7.bin__hadoop-2.6.0.tar.gz
mv sqoop-1.4.7.bin__hadoop-2.6.0 sqoop
vi .bashrc and add
export SQOOP_HOME=/home/ubuntu/sqoop
export PATH=$PATH:$SQOOP_HOME/bin

$ cd $SQOOP_HOME/conf

$ mv sqoop-env-template.sh sqoop-env.sh
Edit sqoop-env.sh and modify the following

```
#Set path to where bin/hadoop is available
export HADOOP_COMMON_HOME=/home/bigdata/hadoop

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

cp mysql-connector-java-8.0.22.jar $SQOOP_HOME/lib
cp commons-lang-2.6 $SQOOP_HOME/lib
```

Installation of Hive

Step 1: Download hive

```
$ wget https://dlcdn.apache.org/hive/hive-3.1.2/apache-hive-3.1.2-bin.tar.gz
$ tar -xvf apache-hive-3.1.2-bin.tar.gz
$ mv apache-hive-3.1.2-bin hive
$ sudo nano .bashrc
Add the following at the last
export HIVE_HOME=/home/bigdata/hive
export PATH=$PATH:$HIVE_HOME/bin
export CLASSPATH=$CLASSPATH:$HIVEHOME/lib

$ source .bashrc
$ hdfs dfs -mkdir -p /user/hive/warehouse
$hdfs dfs -chmod g+w /user/hive/warehouse
```

Step 2: Modify hive-env.sh

```
$ cd hive/conf
$ sudo nano hive-env.sh
Add the following
```

```
export HADOOP_HOME=/home/bigdata/hadoop
export HIVE_CONF_DIR=/home/bigdata/hive/conf
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64

Save and exit
```

Step 4: Configure metastore

```
mysql> create database metastore;
mysql> use metastore;
mysql> SOURCE hive/scripts/metastore/upgrade/mysql/hive-
schema-3.1.0.mysql.sql;
mysql>CREATE USER 'hiveuser'@'%' IDENTIFIED BY 'Metastore1.0';
mysql >GRANT all privileges on metastore.* to 'hiveuser'@'%' ;
mysql>flush privileges;
mysql>exit;
sudo cp ~/mysql-connector-java-8.0.22.jar hive/lib/
sudo cp ~/hadoop/share/hadoop/hdfs/lib/guava-30.0-jre.jar
hive/lib/
sudo rm ~/hive/lib/guava-19.0.jar
```

Step 5: Configure hive-site.xml

```
$ sudo nano hive/config/hive-site.xml
Add the following
<configuration>
  <property>
    <name>javax.jdo.option.ConnectionURL</name>

<value>jdbc:mysql://localhost/metastore?createDatabaseIfNotExis
t=true</value>
    <description>metadata is stored in a MySQL
server</description>
```

```
  </property>
  <property>
    <name>javax.jdo.option.ConnectionDriverName</name>
    <value>com.mysql.cj.jdbc.Driver</value>
    <description>MySQL JDBC driver class</description>
  </property>
  <property>
    <name>javax.jdo.option.ConnectionUserName</name>
    <value>hiveuser</value>
    <description>user name for connecting to mysql
server</description>
  </property>
  <property>
    <name>javax.jdo.option.ConnectionPassword</name>
    <value>Metastore1.0</value>
    <description>hivepassword for connecting to mysql
server</description>
  </property>
</configuration>
```

Step 6: Add the following in mapred-site.xml if not already added

```
<property>
<name>yarn.app.mapreduce.am.env</name>
<value>HADOOP_MAPRED_HOME=${HADOOP_HOME}</value>
</property>
<property>
<name>mapreduce.map.env</name>
<value>HADOOP_MAPRED_HOME=${HADOOP_HOME}</value>
</property>
<property>
<name>mapreduce.reduce.env</name>
<value>HADOOP_MAPRED_HOME=${HADOOP_HOME}</value>
</property>
```

Step 8: Add hive commons jar to sqoop lib

```
cp hive/lib/hive-common-3.1.2.jar sqoop/lib/
```

HBase Installation

```
wget https://dlcdn.apache.org/hbase/1.7.1/hbase-1.7.1-bin.tar.gz
tar xvf hbase-1.7.1-bin.tar.gz
add below like to hbase/conf/hbase-env.sh
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
Add below to .bashrc

export HBASE_HOME=/home/ubuntu/hbase

export PATH=$PATH:$HBASE_HOME/bin


source .bashrc

start-hbase.sh
```

Add following to hbase/conf/hbase-site.xml

```
<property>
    <name>hbase.cluster.distributed</name>
    <value>true</value>
  </property>
<property>
      <name>hbase.rootdir</name>
      <value>hdfs://172.31.89.196:9000/hbase</value>
   </property>
```

```
<property>
  <name>hbase.zookeeper.property.dataDir</name>
  <value>/hadoop/zookeeper</value>
</property>

<property>
  <name>hbase.unsafe.stream.capability.enforce</name>
  <value>false</value>
</property>
```

```
mkdir /hadoop/zookeeper
chmod -R 777 /hadoop/zookeeper
```

Install Spark/Kafka

```
sudo apt update -y
pip3 install boto3

sudo apt install python3-pip
pip3 install jupyter
sudo apt install jupyter-notebook jupyter-core -y
sudo apt-get update
sudo apt install openjdk-8-jdk -y
wget https://downloads.lightbend.com/scala/2.12.8/scala-2.12.8.tgz
tar -xvf scala-2.12.8.tgz
pip3 install py4j
download spark
wget https://dlcdn.apache.org/spark/spark-3.1.2/spark-3.1.2-bin-hadoop3.2.tgz
```

```
wget https://archive.apache.org/dist/kafka/2.8.0/kafka_2.12-2.8.0.tgz
tar -xvf spark-3.1.2-bin-hadoop3.2.tgz
tar -xvf kafka_2.12-2.8.0.tgz

wget https://dlcdn.apache.org/maven/maven-3/3.8.4/binaries/apache-maven-3.8.4-bin.tar.gz
tar -xvf apache-maven-3.8.4-bin.tar.gz
export SPARK_HOME='/home/bigdata/spark-3.1.2-bin-hadoop3.2'
export PATH=$SPARK_HOME:$PATH
export PYTHONPATH=$SPARK_HOME/python:$PYTHONPATH
export PYSPARK_PYTHON=python3
export PATH=$PATH:$SPARK_HOME/bin:/home/bigdata/apache-maven-3.8.2/bin:/home/bigdata/scala-2.12.8/bin:/home/bigdata/spark-3.1.2-bin-hadoop3.2/bin:/home/bigdata/kafka_2.12-2.8.0/bin
source .bashrc
```