**Commands used to run code on pseudo and fully distributed node**

* Step by step instruction.
* Launch an instance that is Amazon Linux AMI.
* Now we see new instance under instances section on AWS.
* Using its public dns log in to that instance into putty.
* Check java version by giving command "java -version".
* Install hadoop having latest version of hadoop-2.6.1 and unzip that hadoop using command "tar -zxvf hadoop-2.6.1.tar.gz".
* Install oozie having latest version of oozie-4.1.0 and unzip it using "tar -zxvf oozie-4.1.0.tar.gz"
* Install maven having latest version of apache maven 3.3.3 and unzip it using "tar -zxvf apache-maven-3.3.3.tar.gz"

**COMMAND FOR BUILDING OOZIE**

To build oozie run the command "bin/mkditro.sh –DskipTests"

This command should build all the projects included in your oozie source folder and should create a binary distribution of oozie that we will use to install client and server.

Then create a new folder called oozie so that we can move our binary distribution to an individual folder.

Copy the folder to new folder oozie by command "cp -R oozie-4.1.0/distro/target/oozie-4.1.0-distro/oozie-4.1.0/ oozie"

Make a new directory into new oozie folder "mkdir libext"

Copy the hadooplibs to libext by command "cp oozie-4.1.0/hadooplibs/hadooplib-2.3.0.oozie-4.1.0/\* libext/"

Update the file in hadoop by /hadoop/etc/hadoop/core-site.xml

<property>

<name>hadoop.proxyuser.ubuntu.hosts</name>

<value>localhost</value>

</property>

<property>

<name>hadoop.proxyuser.ubuntu.groups</name>

<value>hadoop</value>

</property>

To install oozie use the command "/bin/oozie-setup.sh prepare-war" (to install oozie)

Create sharelib on HDFS by command "./bin/oozie-setup.sh sharelib create -fs hdfs://localhost:54310"

This will set up

setting CATALINA\_OPTS="$CATALINA\_OPTS -Xmx1024m"

the destination path for sharelib is: /user/hduser/share/lib

TO create the SQL file "./bin/ooziedb.sh create -sqlfile oozie.sql –run"

To start oozie run the command "./bin/oozied.sh start"

To run oozie use the command "./bin/oozied.sh run"

To check the status of Oozie run the following command: "./bin/oozie admin -oozie http://localhost:11000/oozie -status"

**COMMANDS FOR FULLY DISTRIBUTED MODE**

When you launch an instance of Amazon Linux make sure that you launch 2 instances for cluster setup.

Do the same procedure as you did in single or pseudo mode.

Name each instance as first as namenode and other as slavenode.

Launch every instance using putty and check the IP addresses by giving command "ifconfig" and note down the ip addresses of each instance.

Modify the hostname by giving command "sudo hostname public-dns" on every instance.

To modify it permanently change settings by going into

"sudo vi /etc/hosts".

Add there "private ip" address and you "public dns".

Add .pem file to every instance to "/home/ubuntu" using command line or using filezilla you can directly transfer to it.

Check java version by giving command

"java -version".

Change the hadoop name by

"mv hadoop-2.6.1 hadoop".

**CHANGE THE CONFIGURATION SETTINGS**

Set up the environment for ubuntu user by going to bash file.

Give command vi .bashrc and in that at the end of file set and after that save it by :wq.

export HADOOP\_CONF=/home/ubuntu/hadoop/conf

export HADOOP\_PREFIX=/home/ubuntu/hadoop

export JAVA\_HOME=/usr/lib/jvm/java-7-oracle

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

Give commands to check.

source ~/.bashrc

echo $HADOOP\_PREFIX

echo $HADOOP\_CONF

Add the .pem file to ssh by giving command

chmod 644 .ssh/authorized\_keys

chmod 400 .pem file

eval `ssh-agent -s`

ssh-add .pem file

**CHANGE THE CONFIGURATIONS OF HADOOP**

After that go to cd hadoop/etc/hadoop. In that change the configurations of hadoop-env.sh, core-site.xml, hdfs-site.xml, mapred-site.xml.template and save all the configurations by giving command wq.

Set java\_home in hadoop-env.sh by export JAVA\_HOME=/usr/lib/jvm/java-7-oracle

After that change the configurations core-site.xml :- but befor that make temporary directory to home by command "mkdir hdfstmp"

<configuration>

<property>

<name>fs.default.name</name>

<value>hdfs://public dns:8020</value>

</property>

<property>

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

<value>/home/ubuntu/hdfstmp</value>

</property>

</configuration>

After change the configurations of hdfs-site.xml

<configuration>

<property>

<name>dfs.replication</name>

<value>2</value>

</property>

<property>

<name>dfs.permissions</name>

<value>false</value>

</property>

</configuration>

After that change the configuations of mapred-site.xml.template

<configuration>

<property>

<name>mapred.job.tracker</name>

<value>hdfs://public dns:8021</value>

</property>

</configuration>

Copy all these configurations settings to your slaves node using command

"scp hadoop-env.sh core-site.xml hdfs-site.xml mapred-site.xml.template ubuntu@public dns:/home/ubuntu/hadoop/etc/hadoop" one by one.

Change the masters and slaves file by giving the public dns according to their public dns.

**RUN THE NAMENODES AND DATANODES**

After doing these things format the files by giving command

"hdfs namenode -format"

"hdfs datanode -format".

Go to cd hadoop in that "sbin/start-dfs.sh". After that give command "jps". That will show you namenode and datanodes running on instance.

Pass your input file to /user/ubuntu by command "hdfs dfs -mkdir /user" -> "hdfs dfs -mkdir /user/ubuntu"

Put your input into that by "hdfs dfs -put input input1"

To run your program use the command "hadoop jar ccproject14.jar ccproject.airlines input output1 output2 output3"

To check your output use the command "Hadoop fs –cat output1/part-r-00000" for different outputs.