**Single Node cluster creation in AWS Educate EC2**

Step-1: Open <https://aws.amazon.com/education/awseducate/> and **click on Login to AWS Educate**

If you don’t have account on aws create one using KL Mail

Step-2: Navigate to AWS account then click on **AWS Educate Starter Account** then goto **AWS Console**

Step-3: Goto to EC2 which is under Services/compute 🡪services🡪compute🡪EC2

Step-4: Click on Launch Instance . under search bar type **Ubuntu** then select **Ubuntu Server 18.04 LTS instance**

Step-5: Select one of the type of instance (**General purpose -t2 medium** type is recommended) then click on **Review and Launch** 🡪**Launch**

Step-6: A popup window appear you can select **choose an existing key pair** and **browse the pair** and **launch instance**

If you are creating instance first time then **create a new key pair** 🡪 give keypair name of your wish then download it safely on your system (Key pair is mandatory to login to your instance)

Then click on view instance.

Step-7: Select your instance click on Connect then check on **A standard SSH client** .

Step-8: Open Command prompt on your windows then navigate to the path of key pair which you had already downloaded (Refer step-6)

Step-9: Connect to your instance using its public DNS: copy and past the ssh command shown on your aws Connect to your instance window.

Example: **ssh -i "tarunsai.pem" ubuntu@ec2-18-232-129-119.compute-1.amazonaws.com**

Note : .pem file name , instance username differs from one another

A screenshot of a computer screen

Description automatically generated

Now you logged in to your instance using ssh connectivity.

Step-10: Update and upgrade pakages in ubuntu using

$ **sudo apt-get update && sudo apt-get upgrade command**

**Step-11:** Start installing Hadoop on ubuntu terminal

1. Install java on ubuntu🡪$ sudo apt-get install default-jdk
2. Generate SSH key for Hadoop 🡪$ ssh-keygen -t rsa -P ""
3. enable SSH access to your virally created machine with this newly created key. 🡪cat $HOME/.ssh/id\_rsa.pub >> $HOME/.ssh/authorized\_keys
4. Test your connectivity to local host 🡪 ssh localhost
5. exit from the localhost 🡪 $exit
6. Download Hadoop 🡪$ wget <https://archive.apache.org/dist/hadoop/common/hadoop-2.8.5/hadoop-2.8.5.tar.gz>
7. Extract Hadoop tar file 🡪$ tar -xzvf hadoop-2.8.5.tar.gz
8. Edit bashrc 🡪 $ nano ./.bashrc
9. Paste these export statements at the end of the file

* export JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-amd64
* export HADOOP\_HOME=/home/ubuntu/hadoop-2.8.5
* 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 PATH=$PATH:$HADOOP\_HOME/sbin:$HADOOP\_HOME/bin
* export HADOOP\_OPTS="-Djava.library.path=$HADOOP\_HOME/lib/native"
* save and exit by CTRl+O 🡪Enter 🡪CTRL + X

1. source the bashrc file 🡪 $ source ~/.bashrc
2. edit the Hadoop-env.sh file 🡪$ nano /home/ubuntu/hadoop-2.8.5/etc/had

* modify export JAVA\_HOME path to 🡪export JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-amd64
* modify export HADOOP\_CONF\_DIR to 🡪export HADOOP\_CONF\_DIR=${HADOOP\_CONF\_DIR:-"/home/ubuntu/hadoop-2.8.5/etc/hadoop"}
* save and exit by CTRl+O 🡪Enter 🡪CTRL + X

1. Edit core-site.xml file configuration🡪 $ nano /home/ubuntu/hadoop-2.8.5/etc/hadoop/core-site.xml

* Add these configuration to core-site.xml file

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://localhost:9000</value>

</property>

<property>

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

<value>/home/ubuntu/hadooptmpdata</value>

</property>

</configuration>

* Save and exit

1. Create these directories

* $ mkdir hadooptmpdata
* $ mkdir -p hdfs/datanode
* $ mkdir -p hdfs/datanode

1. Edit hdfs-site.xml file🡪$ nano /home/ubuntu/hadoop-2.8.5/etc/hadoop/hdfs-site.xml

* Add these configuration to hdfs-site.xml file

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

<name>dfs.name.dir</name>

<value>file:///home/hadoop/hdfs/namenode</value>

<name>dfs.data.dir</name>

<value>file:///home/hadoop/hdfs/datanode</value>

</property>

</configuration>

1. Copy mapred template 🡪 $ cp hadoop-2.8.5/etc/hadoop/mapred-site.xml.template hadoop-2.8.5/etc/hadoop/mapred-site.xml
2. Edit mapred-site.xml file🡪$ nano /home/ubuntu/hadoop-2.8.5/etc/hadoop/mapred-site.xml

* Add these configuration to mapred-site.xml file

<configuration>

<property>

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

<value>yarn</value>

</property>

</configuration>

1. Edit yarn-site.xml file 🡪$nano /home/ubuntu/hadoop-2.8.5/etc/hadoop/yarn-site.xml

* Add these configuration to yarn-site.xml file

<configuration>

<property>

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

<value>mapreduce\_shuffle</value>

</property>

</configuration>

1. Format the namenode before using it 🡪$ hdfs namenode -format
2. Start the services of hadoop 🡪$ start-all.sh
3. Check the started services🡪jps
4. If every services of hadoop starts then exit from the ubuntu connection 🡪$ exit

**Go back to aws console in the browser and select the created instance 🡪Actions🡪Instance state🡪stop**

**NOTE: Active internet connection is required while using AWS instances and must stop the running instances before signing out from aws console**