**Programming Assignment 2, Using of spark mllib on 4 ec2 instances with docker and container environment.**

**# Namae: Jalaj Sharma**

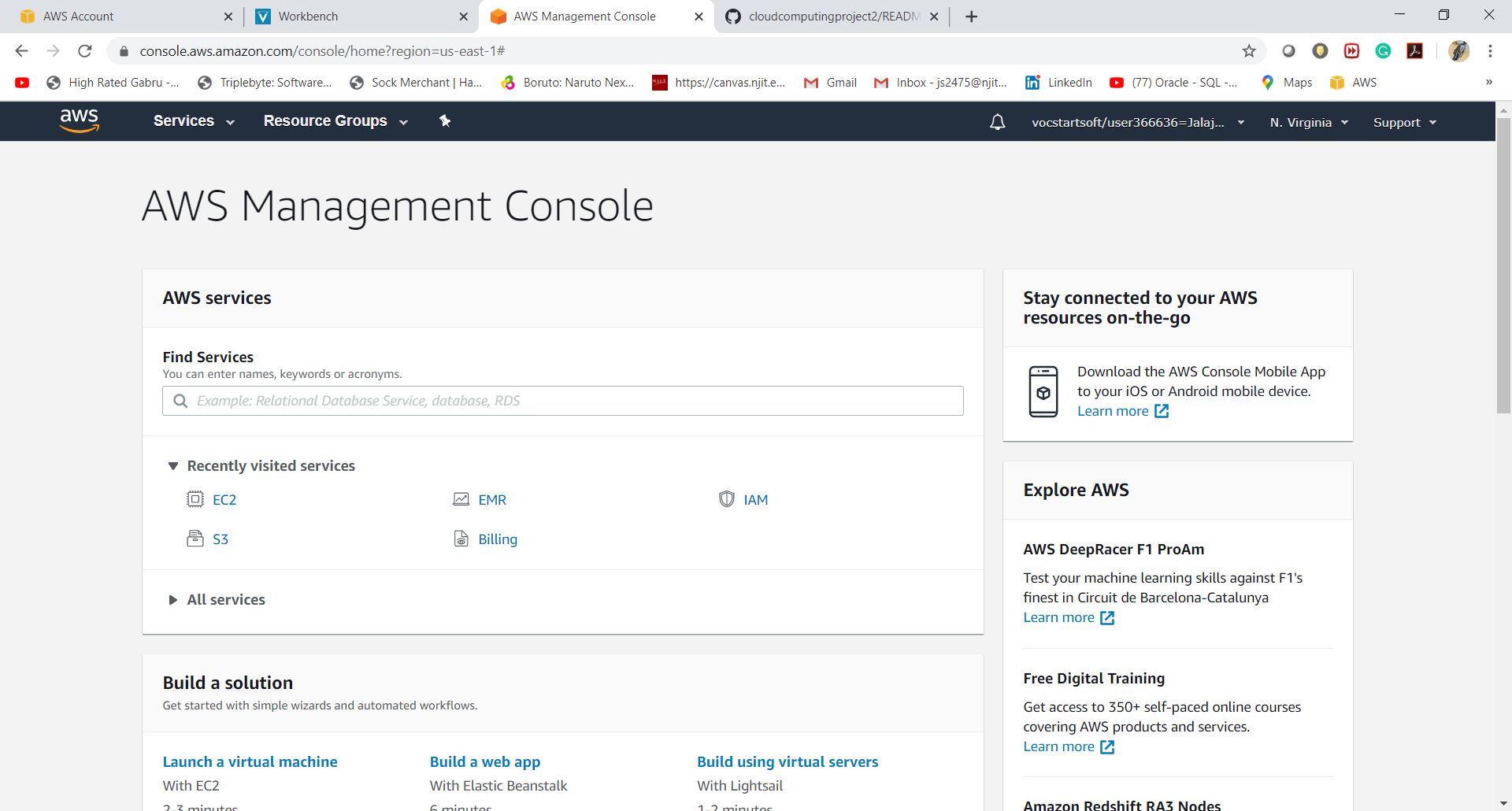
**# UCID : js2475**

**#github link: https://github.com/jalajsharma93/cloudcomputingproject2**

**#docker link : https://hub.docker.com/u/jalajsharma93**

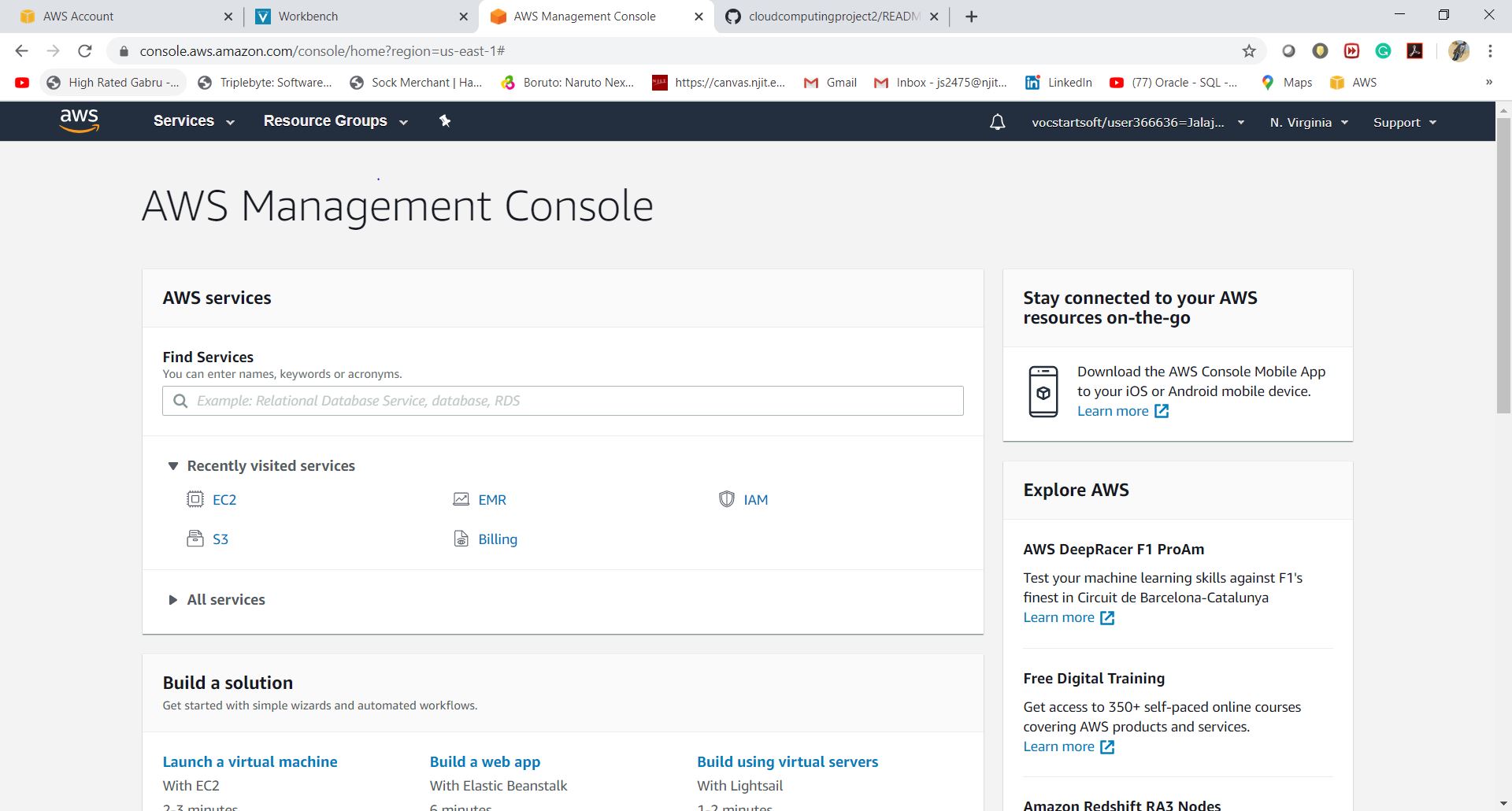
**Creating EMR cluster,**

For creating cluster got to AWS account



**Click on EMR,**

if emr is not there search on search bar and select it.

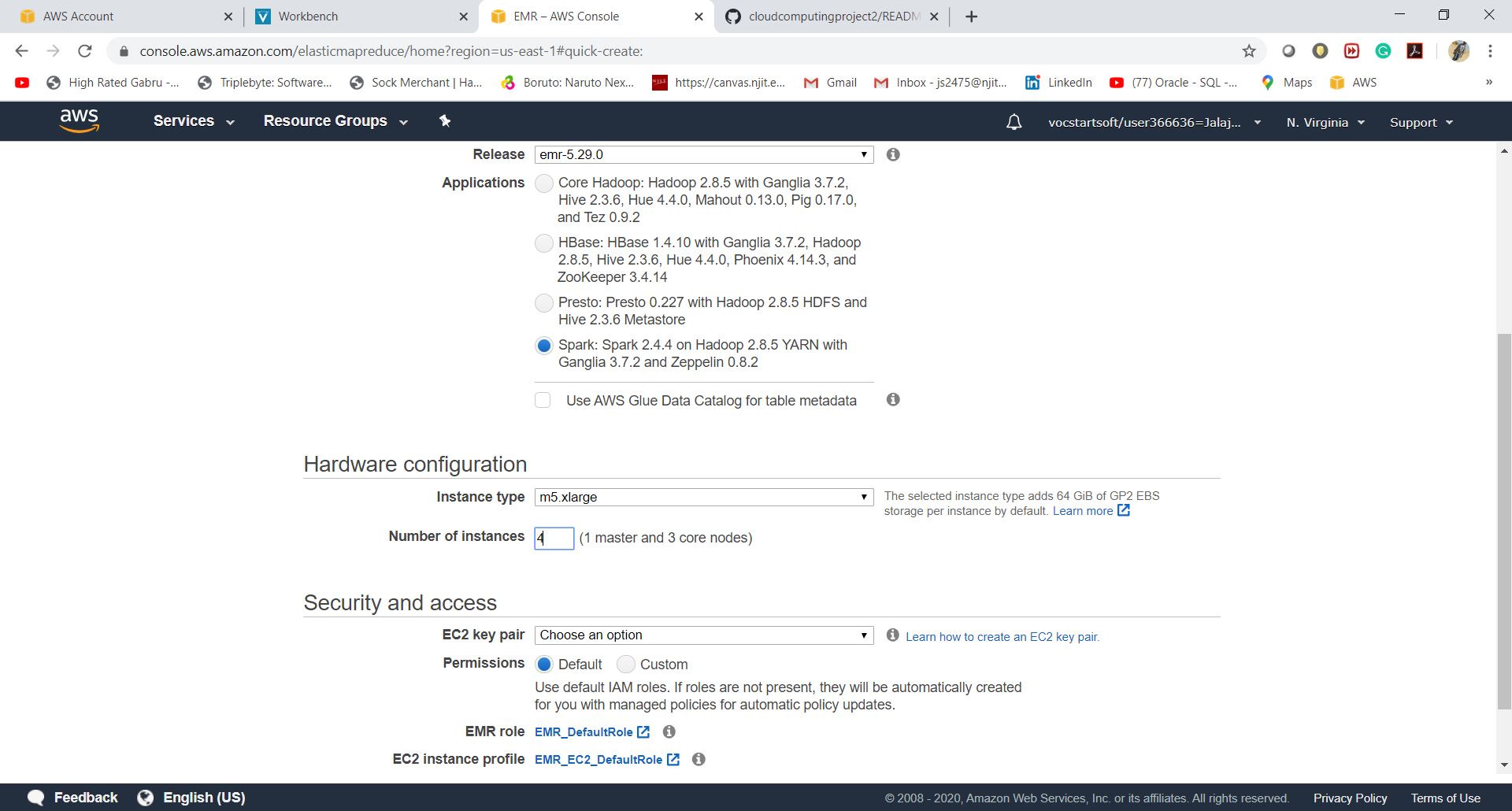
**Click on create cluster.**

**Select if next click on create cluster and then a page apper with config.**

**Name cluster with name you want for project.**

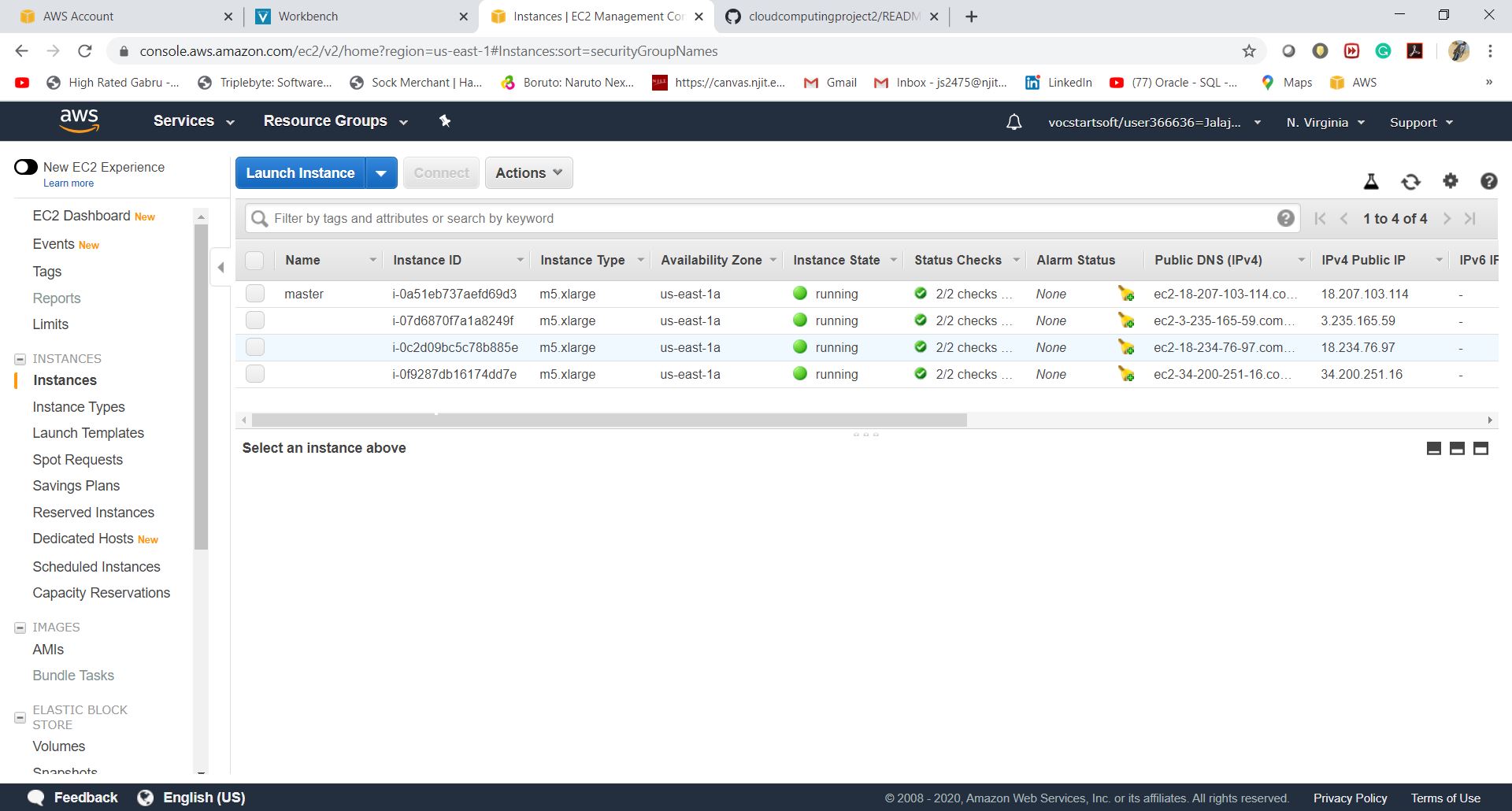
**Select spark Hadoop, yarn with ganglia and Zeppline**

**Bellow select number of Instance to 4.**

****

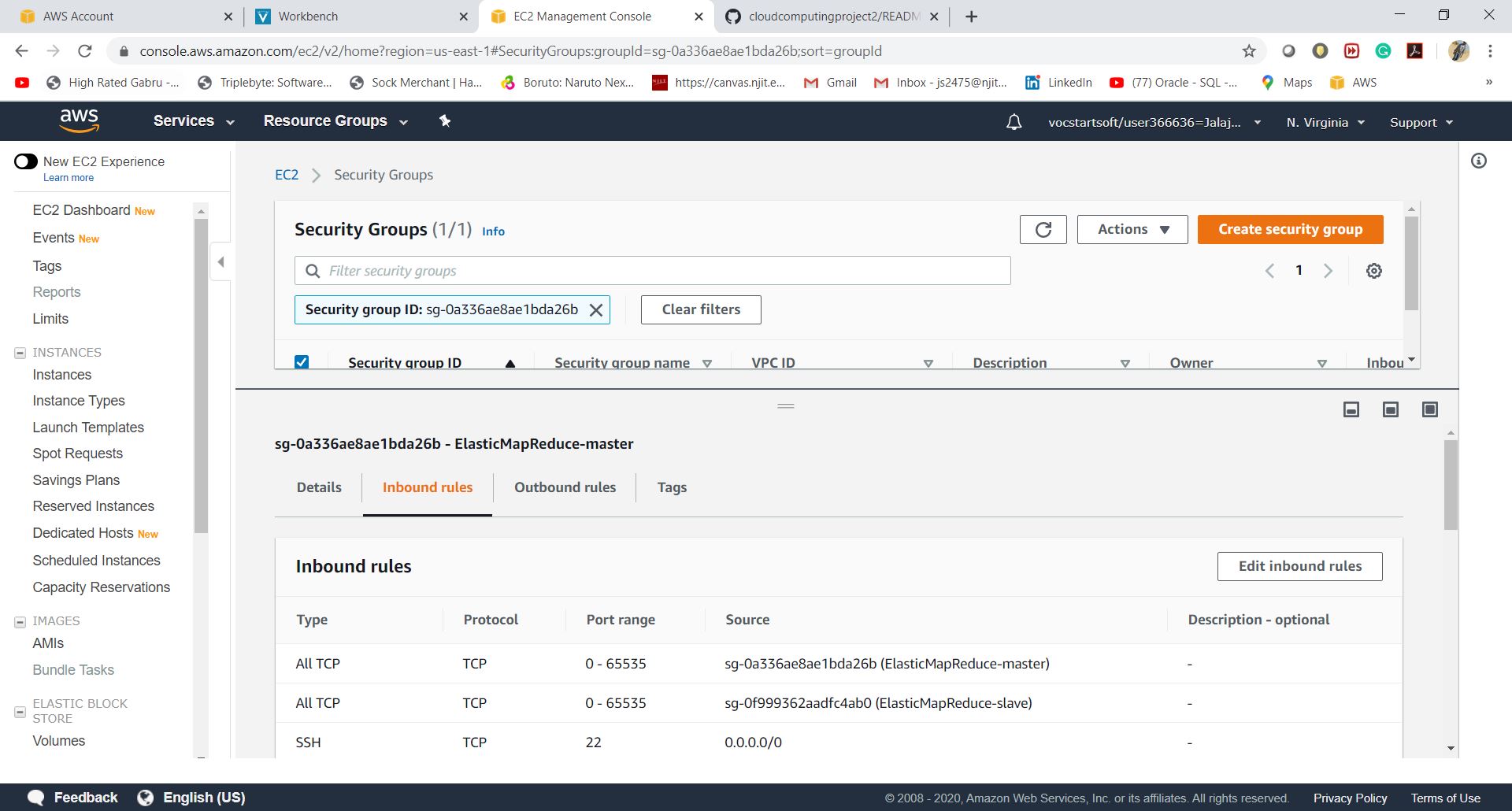
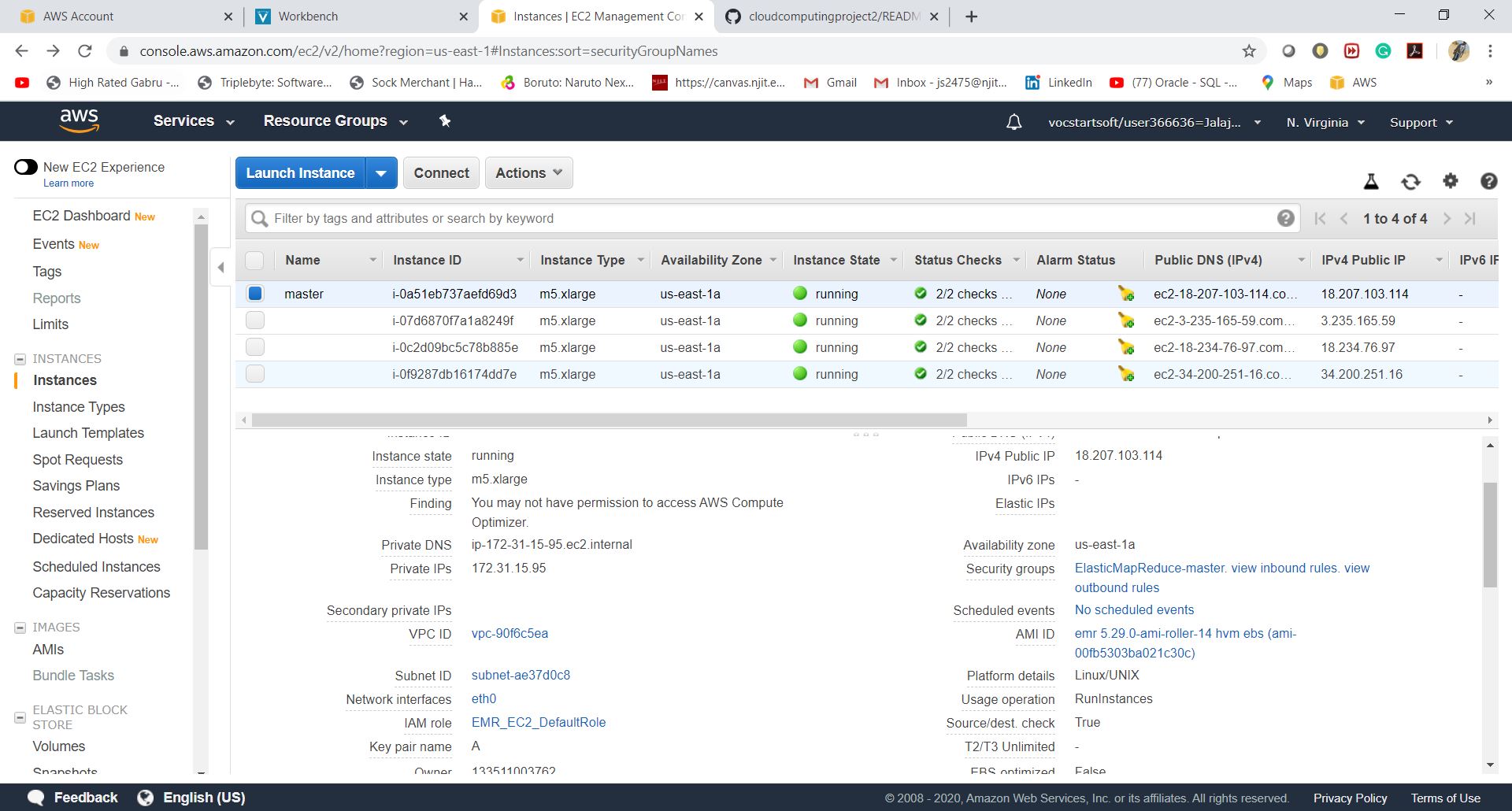
**After that you can see your 4 Intances with 1 master and 3 slave are ready in EC2 instances,**

**For that go to EC2 same as you search EMR and now Search EC2**

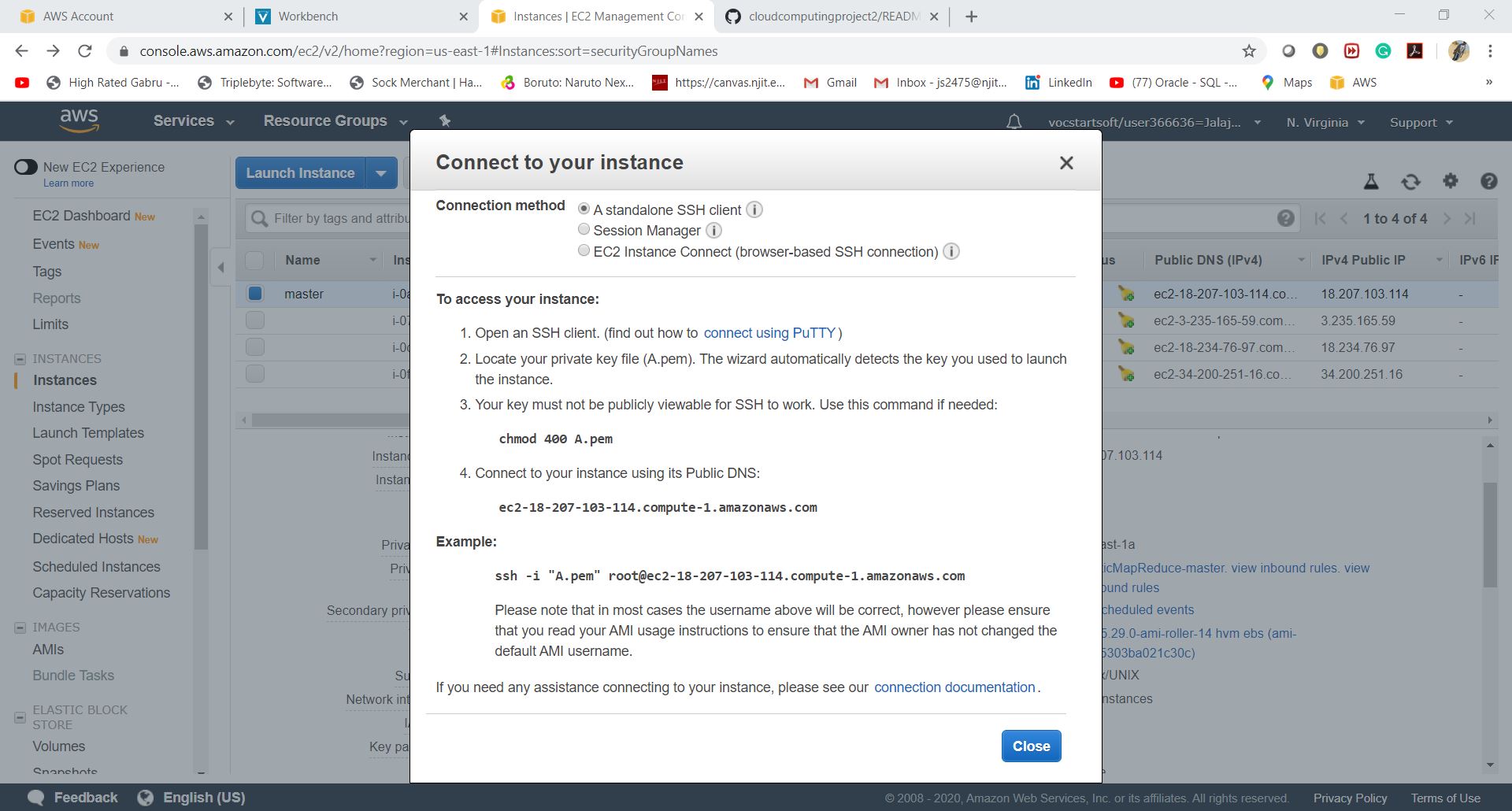
****

**Now we have to change the inbound rule to allow ssh connection. For that select master on EC2 instance, scroll down and in description select *ElasticMapReduce* under security groups.**

**Inside It edit inboud rule on top right corner and there add SSH.**

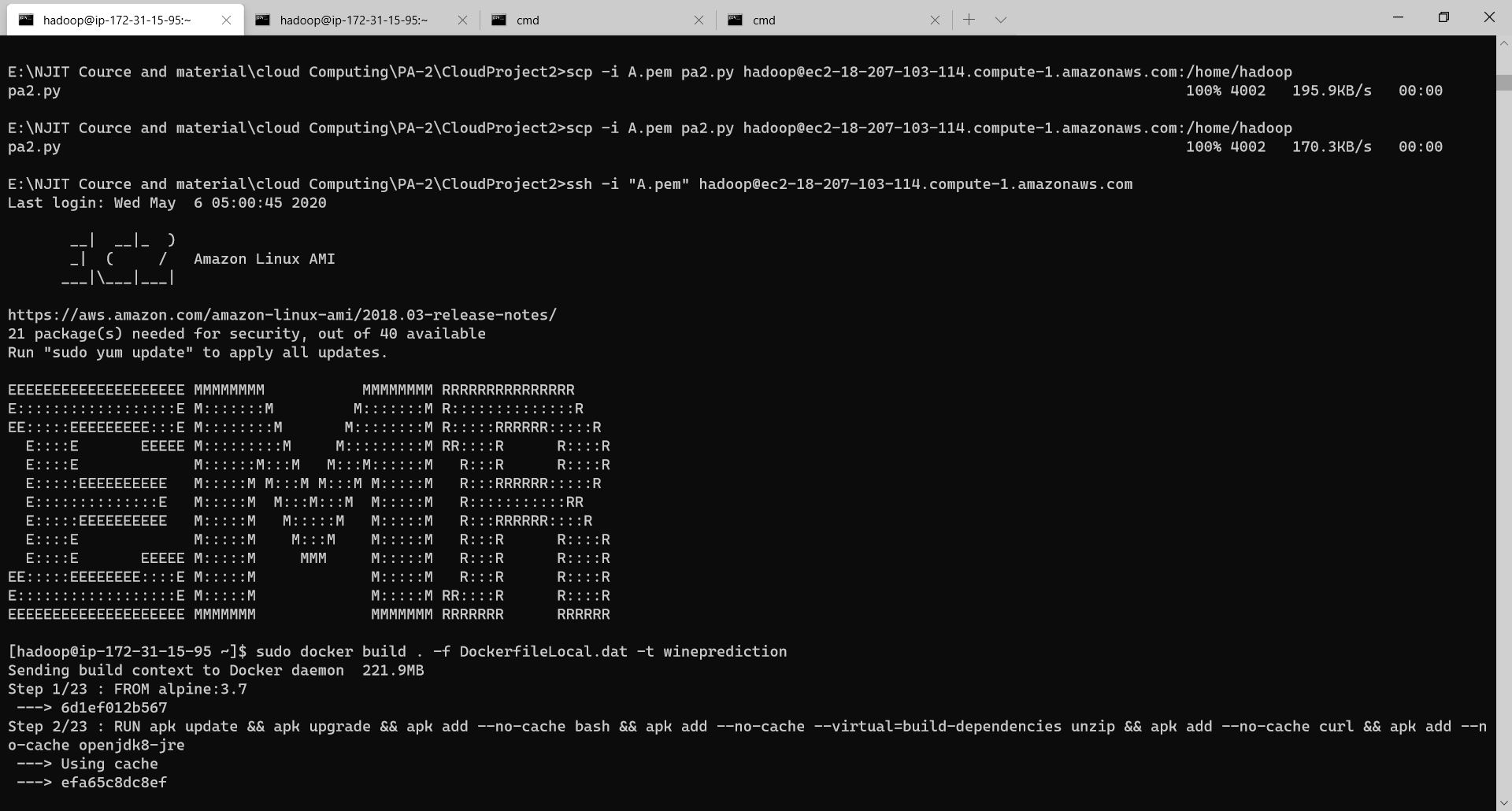
****

**Now again select master and click on connect on top you will see window like this.**

****

**Now we will connect to EMR cluster.**

**ssh -i "A.pem" hadoop@<your address from above screenhot point 4 in TO access your instance.**

****

**After connection from local terminal upload all data to instance using SCP command**

**Scp -i A.key \* hadopp@<address from ec2 instance connect> :/home/Hadoop/**

**Then all file will be uploaded.**

**Make sure your key have correct permissions.**

**Now we have to install python, pyspark and docker**

**They need dependencies also.**

**For that we have to install in order to train model in EMR and Predict on the cluster.**

**Sudo yum install python3**

**pip3 install wheel**

**pip3 install pyspark==2.4.5 --no-cache-dir**

**pip3 install findspark**

**if pip3 don’t work try pip or try sudo pip or sudo pip3.**

**After that install docker**

**Sudo yum install docker**

**Now we are ready to run our application for that we have to train our model on hdfs system**

**For that**

**Use command**

1. hadoop fs -put TrainingDataset.csv /
2. hadoop fs -put ValidationDataset.csv

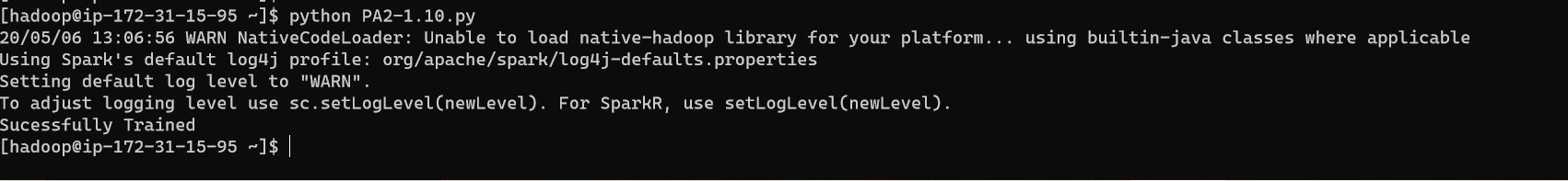
**This will add file to hdfs system for training and validation.**

**After that we have to run our training for that run command**

**python PA2-1.10.py**

**--python <training file name>.py**

**This will train the model, which we have to make sure that we upload it into our doker Image.**

****

**Now we have to build our doker image using command**

**sudo docker build . -f DockerfileLocal.dat -t jalajsharma93/cloudcomputingpa2**

**this will create our image with name** *cloudcomputingpa2*

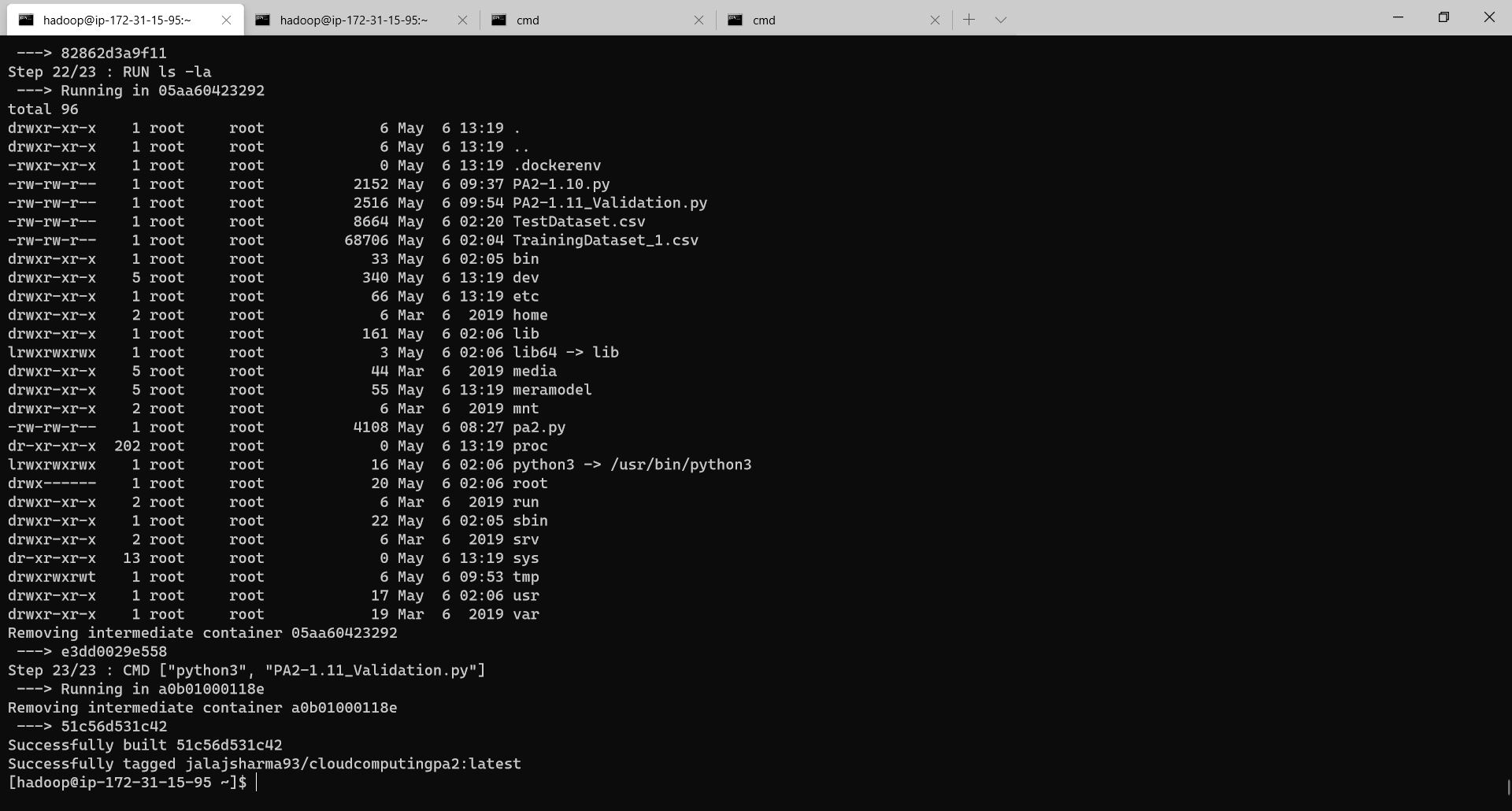
**jalajsharma93 is the username for docker. So if you have different user name please edit it, in last cloudcomputingpa2 is repository name so you can edit it as you want.**

**Now we have to run our dockker image.**

**sudo docker run -t jalajsharma93/cloudcomputingpa2**

**above command will run it.**

**This will give output.**

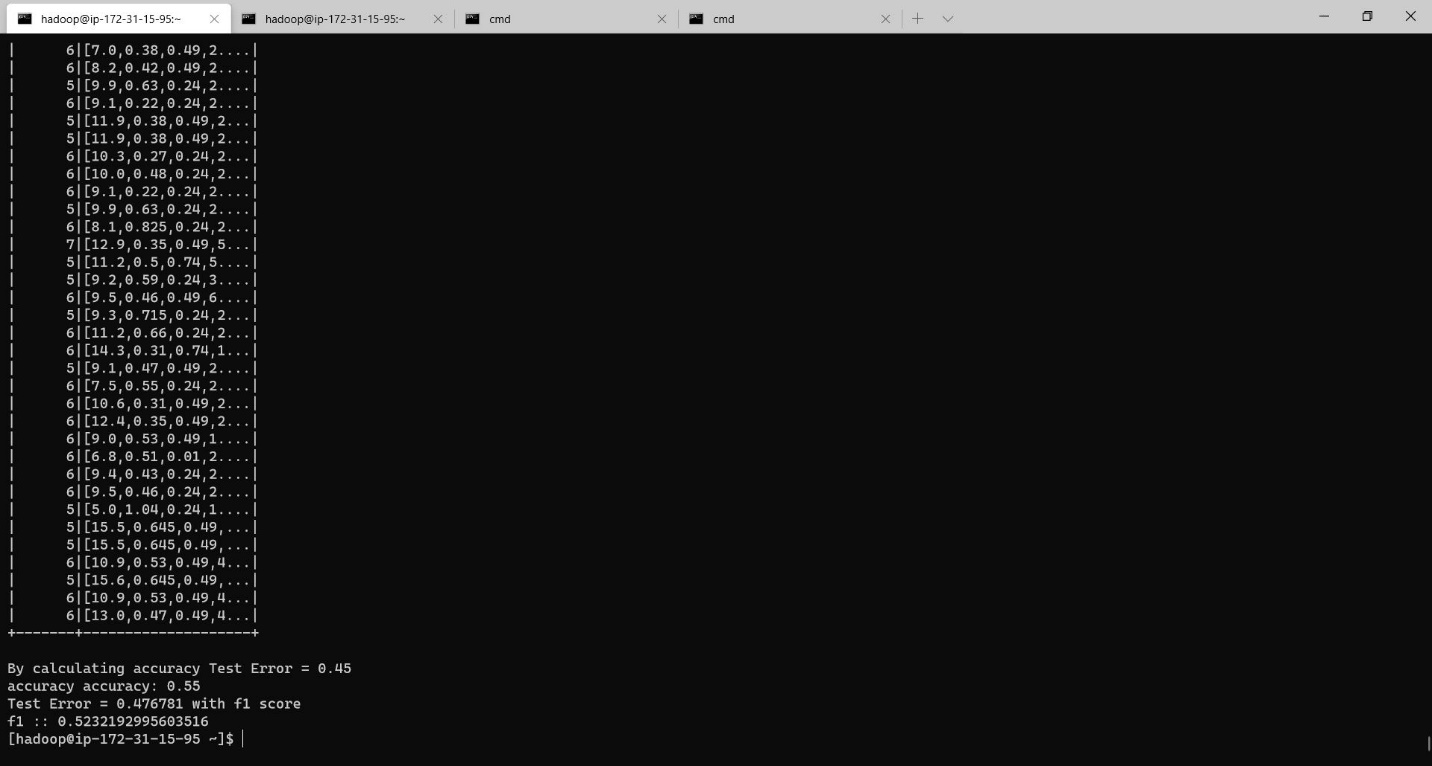
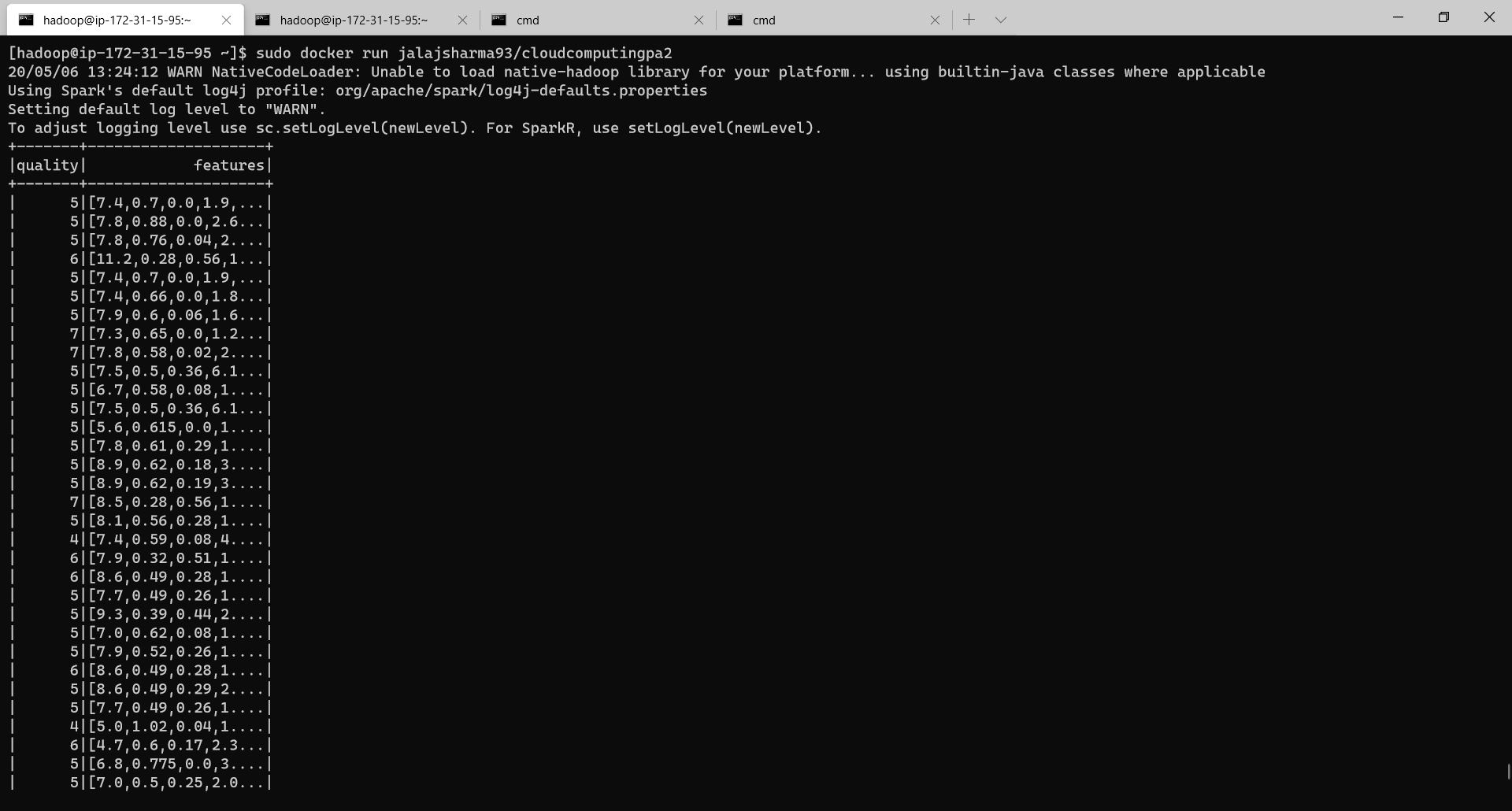
****

**Running Output:**

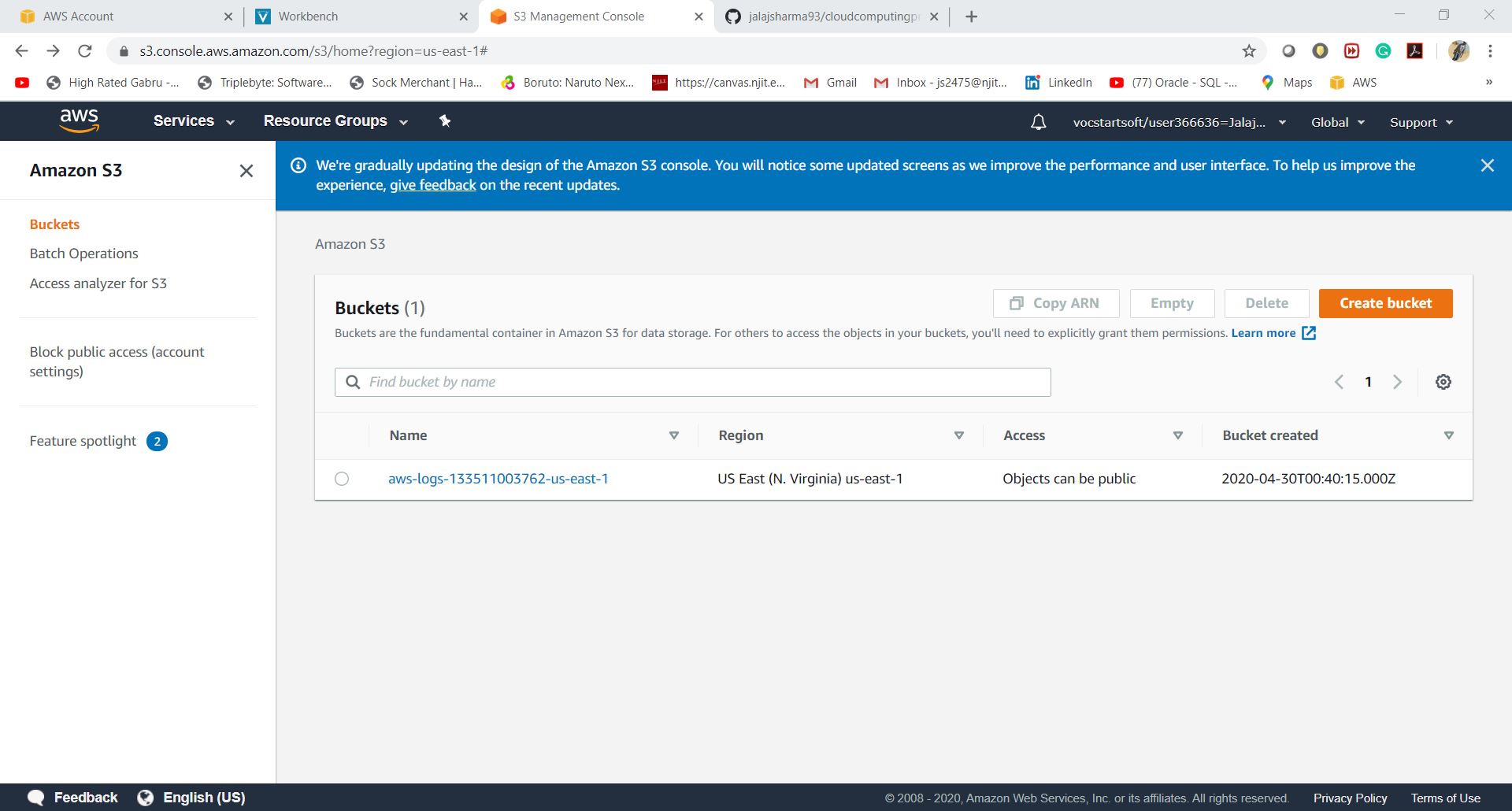
Showing only some commands please run code for full output

Please comment if you don’t need features and quality. And accuracy.

As it mentioned in the code **PA2-1.11\_Validation.py**

****

**The final output is ranging between 53 to 65% in term of accuracy percentage or in score .54 to .65 for accuracy and same for f1 score**

****

**Logs for training can be found here.**

**Now we can push our build file to Docker,**

**Run commad**

**Sudo docker login -u <username>**

**//it will ask for password please type it.**

**And then**

**Sudo docker push.**

**It will push file to docker.**

**Now we are ready to upload everything in git hub.**

**For that please look for command on git hub docs or git hub have nice user interface please look upload and download it from there.**

**Commands might help**

**# cloudcomputingproject2**

**We have several steps to follow which includes sub commands also**

**a)Set up EMR with 3 slave and 1 Master --- Required by professor**

**b)uploads all the file to EMR**

**c)Install python, pyspark, docker with all dependencies required to run pyspark**

**d)Create dokerFile, start docker, build, run and push to dockehub**

**## Setting up EMR cluster**

**please check document with extention.pdf or word dockehub**

**#copy file to cluster**

**Scp -i A.key \* hadopp@<address from ec2 instance connect> :/home/hadoop/**

**Scp -i <key.pem with path> <directory or file with path> hadopp@<address from ec2 instance connect> :/home/Hadoop/**

**#Connect to EMR cluster**

**ssh -i A.key hadoop@<address from ec2 instance>**

**## Installing python**

**sudo pip install --upgrade pip**

**sudo apt install python3-pip**

**#installing pyspark, docker and dependencies**

**sudo pip install --upgrade pip**

**sudo pip install wheel**

**sudo pip install pyspark --no-cache-dir**

**sudo pip install findspark**

**sudo pip install numpy**

**sudo yum install -y docker**

**## sudo service docker start**

**sudo docker build . -f Dockerfile -t jalajsharma93/cloudcomputingpa2**

**sudo docker run -t cloudcomputingpa2**

**##for Pushing file to docker**

**sudo docker login -u <user\_name>**

**<type password it will ask for it>**

**sudo docker push**

**For more please check docker file it will give you brief idea what to install because it has them all in one place**

**References.**

1. <https://medium.com/@dhiraj.p.rai/logistic-regression-in-spark-ml-8a95b5f5434c>
2. <https://medium.com/@dhiraj.p.rai/logistic-regression-in-spark-ml-8a95b5f5434c>
3. <https://spark.apache.org/docs/latest/mllib-dimensionality-reduction>
4. <https://datascience.stackexchange.com/questions/9424/spark-mllib-multiclass-logistic-regression-how-to-get-the-probabilities-of-all/11444>
5. <https://towardsdatascience.com/predict-customer-churn-using-pyspark-machine-learning-519e866449b5>
6. <https://spark.apache.org/docs/latest/ml-classification-regression.html#random-forest-classifier>
7. <https://spark.apache.org/docs/2.2.0/mllib-evaluation-metrics.html>
8. <https://stackoverflow.com/questions/43835504/error-attributeerror-py4jerror-object-has-no-attribute-message-building-de>
9. <https://mapr.com/blog/churn-prediction-pyspark-using-mllib-and-ml-packages/>

<https://runawayhorse001.github.io/LearningApacheSpark/pyspark.pdf>

1. <https://stackoverflow.com/questions/30063907/using-docker-compose-how-to-execute-multiple-commands>
2. <https://mlinproduction.com/docker-for-ml-part-4/>
3. <https://medium.com/@thiagolcmelo/submitting-a-python-job-to-apache-spark-on-docker-b2bd19593a06>
4. <https://docs.docker.com/engine/reference/commandline/run/>