

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>

Please comment out line no. 13 for testing on ValidationDataset.csv,

And comment in 14 for same, if want to test on TestDataset.csv please do not remove anything from **PA2-1.11_Validation.py**

Creating EMR cluster,

For creating cluster got to AWS account

The screenshot shows the AWS Management Console interface. At the top, there's a navigation bar with the AWS logo, 'Services' and 'Resource Groups' dropdowns, a user profile 'vocstartsoft/user366636=Jalaj...', the region 'N. Virginia', and a 'Support' link. Below the navigation bar, the main heading 'AWS Management Console' is displayed. The left sidebar contains 'AWS services' with a 'Find Services' search bar (placeholder: 'Example: Relational Database Service, database, RDS') and a 'Recently visited services' section showing icons for EC2, EMR, IAM, S3, and Billing. Below this is a link to 'All services'. The main content area features a 'Build a solution' section with the text 'Get started with simple wizards and automated workflows.' and three cards: 'Launch a virtual machine' (With EC2, 2-2 minutes), 'Build a web app' (With Elastic Beanstalk, 6 minutes), and 'Build using virtual servers' (With Lightsail, 1-2 minutes). On the right, there are three promotional boxes: 'Stay connected to your AWS resources on-the-go' (promoting the AWS Console Mobile App), 'Explore AWS' (featuring 'AWS DeepRacer F1 ProAm' and 'Free Digital Training'), and 'Amazon Redshift RA3 Nodes'.

AWS services

Find Services
You can enter names, keywords or acronyms.

Q Example: Relational Database Service, database, RDS

▼ Recently visited services

- EC2
- EMR
- IAM
- S3
- Billing

► All services

Build a solution
Get started with simple wizards and automated workflows.

- Launch a virtual machine**
With EC2
2-2 minutes
- Build a web app**
With Elastic Beanstalk
6 minutes
- Build using virtual servers**
With Lightsail
1-2 minutes

Stay connected to your AWS resources on-the-go

Download the AWS Console Mobile App to your iOS or Android mobile device.
[Learn more](#)

Explore AWS

AWS DeepRacer F1 ProAm
Test your machine learning skills against F1's finest in Circuit de Barcelona-Catalunya
[Learn more](#)

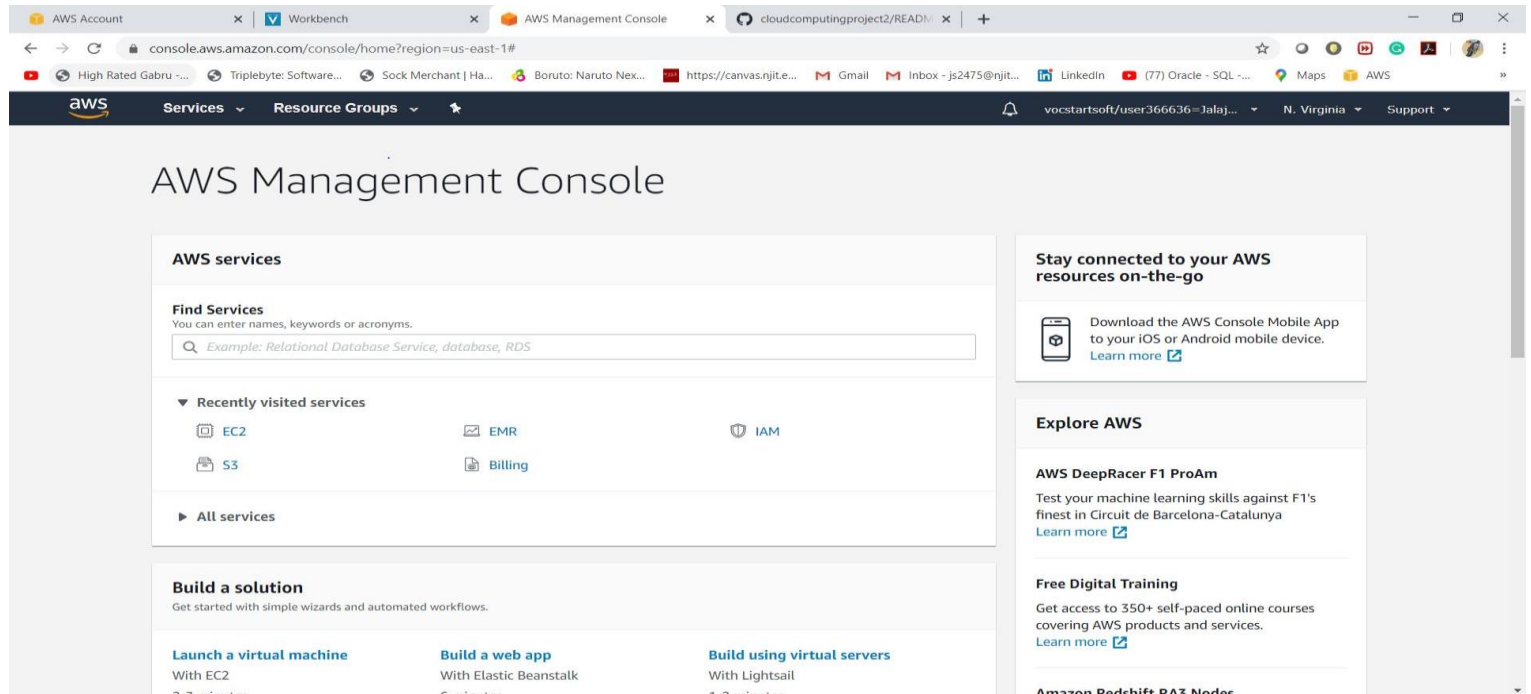
Free Digital Training
Get access to 350+ self-paced online courses covering AWS products and services.
[Learn more](#)

Amazon Redshift RA3 Nodes

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 Zepline

Bellow select number of Instance to 4.

Release: **emr-5.29.0**

Applications:

- ☐ Core Hadoop: Hadoop 2.8.5 with Ganglia 3.7.2, Hive 2.3.6, Hue 4.4.0, Mahout 0.13.0, Pig 0.17.0, and Tez 0.9.2
- ☐ HBase: HBase 1.4.10 with Ganglia 3.7.2, Hadoop 2.8.5, Hive 2.3.6, Hue 4.4.0, Phoenix 4.14.3, and ZooKeeper 3.4.14
- ☐ Presto: Presto 0.227 with Hadoop 2.8.5 HDFS and Hive 2.3.6 Metastore
- ☒ Spark: Spark 2.4.4 on Hadoop 2.8.5 YARN with Ganglia 3.7.2 and Zeppelin 0.8.2
- ☐ Use AWS Glue Data Catalog for table metadata

Hardware configuration

Instance type: **m5.xlarge** The selected instance type adds 64 GiB of GP2 EBS storage per instance by default. [Learn more](#)

Number of instances: **4** (1 master and 3 core nodes)

Security and access

EC2 key pair: **Choose an option** [Learn how to create an EC2 key pair.](#)

Permissions: ☒ Default ☐ Custom
Use default IAM roles. If roles are not present, they will be automatically created for you with managed policies for automatic policy updates.

EMR role: [EMR_DefaultRole](#)

EC2 instance profile: [EMR_EC2_DefaultRole](#)

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

Instances | EC2 Management Console

Filter by tags and attributes or search by keyword

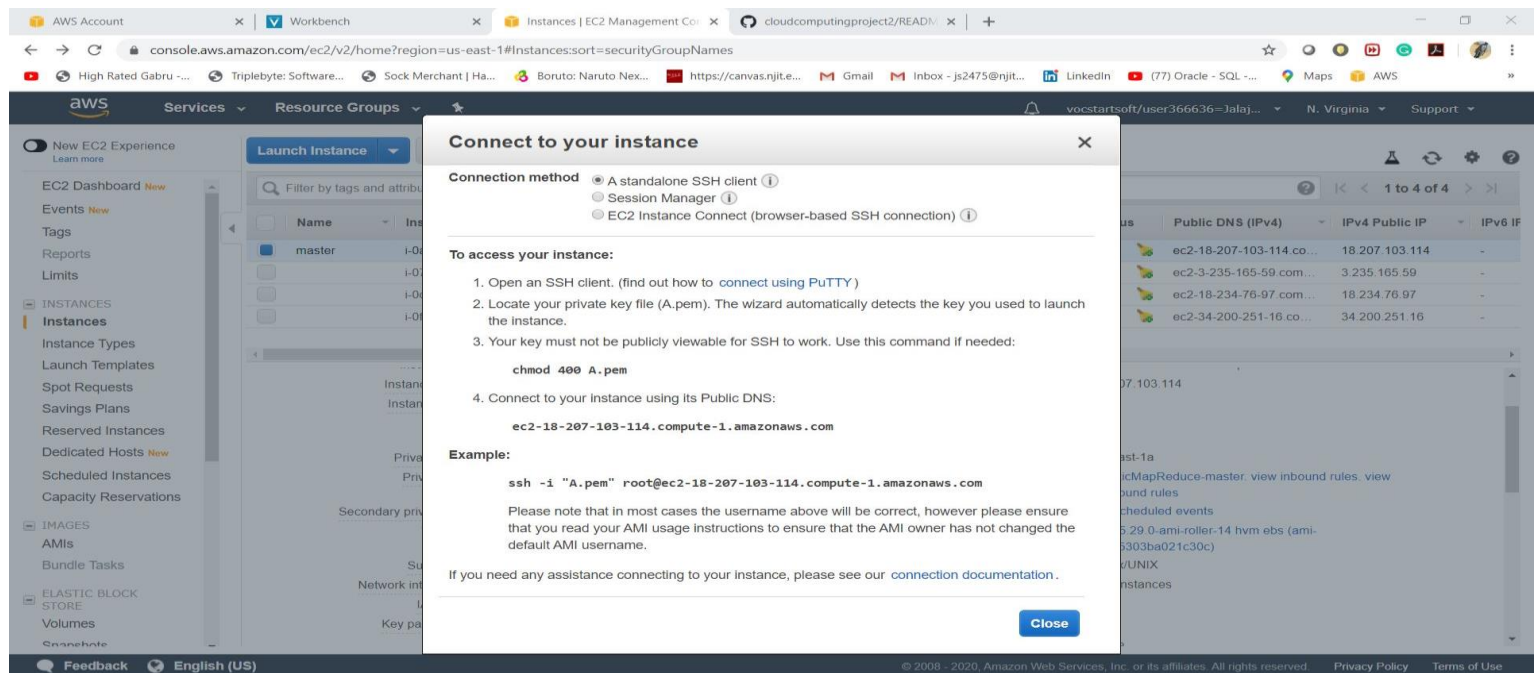
| Name | Instance ID | Instance Type | Availability Zone | Instance State | Status Checks | Alarm Status | Public DNS (IPv4) | IPv4 Public IP | IPv6 IP |
|--------|---------------------|---------------|-------------------|----------------|----------------|--------------|--------------------------|----------------|---------|
| master | i-0a51eb737aef069d3 | m5.xlarge | us-east-1a | running | 2/2 checks ... | None | ec2-18-207-103-114.co... | 18.207.103.114 | - |
| | i-07d68707a1a8249f | m5.xlarge | us-east-1a | running | 2/2 checks ... | None | ec2-3-235-165-59.com... | 3.235.165.59 | - |
| | i-0c2d09bc5c78b885e | m5.xlarge | us-east-1a | running | 2/2 checks ... | None | ec2-18-234-76-97.com... | 18.234.76.97 | - |
| | i-0f9287db16174dd7e | m5.xlarge | us-east-1a | running | 2/2 checks ... | None | ec2-34-200-251-16.co... | 34.200.251.16 | - |

Select an instance above

Inside It edit inbound 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 screenshot point 4 in TO access your instance.
```

```
hadoop@ip-172-31-15-95:~$ scp -i A.pem pa2.py hadoop@ec2-18-207-103-114.compute-1.amazonaws.com:/home/hadoop
pa2.py
E:\NJIT Course and material\cloud Computing\PA-2\CloudProject2>scp -i A.pem pa2.py hadoop@ec2-18-207-103-114.compute-1.amazonaws.com:/home/hadoop
pa2.py
E:\NJIT Course and material\cloud Computing\PA-2\CloudProject2>ssh -i "A.pem" hadoop@ec2-18-207-103-114.compute-1.amazonaws.com
Last login: Wed May 6 05:00:45 2020

--|  --|  )
_|  C  /   Amazon Linux AMI
---|\---|---|

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
21 package(s) needed for security, out of 40 available
Run "sudo yum update" to apply all updates.

EEEEEEEEEEEEEEEEEEEE MMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M::: M M::: M R::: R
EE::::::::::::::::::E M::: M M::: M R::: RRRRRR:::R
E:::E EEEEE M::: M M::: M RR:::R R:::R
E:::E M::: M M::: M M::: M R:::R R:::R
E:::EEEEEEEEEE M::: M M::: M M::: M R::: RRRRRR:::R
E::: M::: M M::: M M::: M R::: RRRRRR:::R
E:::EEEEEEEEEE M::: M M::: M M::: M R::: RRRRRR:::R
E:::E M::: M M::: M M::: M R:::R R:::R
E:::E EEEEE M::: M M M M::: M R:::R R:::R
EE::::::::::::::::::E M::: M M::: M R:::R R:::R
E::::::::::::::::::E M::: M M::: M RR:::R R:::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM RRRRRRR RRRRRR

[hadoop@ip-172-31-15-95 ~]$ sudo docker build . -f DockerfileLocal.dat -t wineprediction
Sending build context to Docker daemon 221.9MB
Step 1/23 : FROM alpine:3.7
--> 6d1ef012b567
Step 2/23 : RUN apk update && apk upgrade && apk add --no-cache bash && apk add --no-cache --virtual=build-dependencies unzip && apk add --no-cache curl && apk add --no-cache openjdk8-jre
--> Using cache
--> efa65c8dc8ef
```

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 docker Image.

```
[hadoop@ip-172-31-15-95 ~]$ python PA2-1.10.py
20/05/06 13:06:56 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Sucessfully Trained
[hadoop@ip-172-31-15-95 ~]$ |
```

Now we have to build our docker 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.

```
hadoop@ip-172-31-15-95:~ X hadoop@ip-172-31-15-95:~ X cmd X + v - [X]
---> 82862d3a9f11
Step 22/23 : RUN ls -la
---> Running in 05aa60423292
total 96
drwxr-xr-x 1 root root      6 May 6 13:19 .
drwxr-xr-x 1 root root      6 May 6 13:19 ..
-rwxr-xr-x 1 root root      0 May 6 13:19 .dockerenv
-rw-rw-r-- 1 root root    2152 May 6 09:37 PA2-1.10.py
-rw-rw-r-- 1 root root    2516 May 6 09:54 PA2-1.11_Validation.py
-rw-rw-r-- 1 root root    8664 May 6 02:20 TestDataset.csv
-rw-rw-r-- 1 root root   68706 May 6 02:04 TrainingDataset_1.csv
drwxr-xr-x 1 root root      33 May 6 02:05 bin
drwxr-xr-x 5 root root     340 May 6 13:19 dev
drwxr-xr-x 1 root root      66 May 6 13:19 etc
drwxr-xr-x 2 root root      6 Mar 6 2019 home
drwxr-xr-x 1 root root     161 May 6 02:06 lib
lrwxrwxrwx 1 root root      3 May 6 02:06 lib64 -> lib
drwxr-xr-x 5 root root      44 Mar 6 2019 media
drwxr-xr-x 5 root root      55 May 6 13:19 meramodel
drwxr-xr-x 2 root root      6 Mar 6 2019 mnt
-rw-rw-r-- 1 root root    4108 May 6 08:27 pa2.py
dr-xr-xr-x 202 root root      0 May 6 13:19 proc
lrwxrwxrwx 1 root root     16 May 6 02:06 python3 -> /usr/bin/python3
drwx----- 1 root root      20 May 6 02:06 root
drwxr-xr-x 2 root root      6 Mar 6 2019 run
drwxr-xr-x 1 root root     22 May 6 02:05 sbin
drwxr-xr-x 2 root root      6 Mar 6 2019 srv
dr-xr-xr-x 13 root root      0 May 6 13:19 sys
drwxrwxrwt 1 root root      6 May 6 09:53 tmp
drwxr-xr-x 1 root root     17 May 6 02:06 usr
drwxr-xr-x 1 root root     19 Mar 6 2019 var
Removing intermediate container 05aa60423292
---> e3dd0029e558
Step 23/23 : CMD ["python3", "PA2-1.11_Validation.py"]
---> Running in a0b01000118e
Removing intermediate container a0b01000118e
---> 51c56d531c42
Successfully built 51c56d531c42
Successfully tagged jalajsharma93/cloudcomputingpa2:latest
[hadoop@ip-172-31-15-95 ~]$ |
```

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**

```
hadoop@ip-172-31-15-95:~$ sudo docker run jalajsharma93/cloudcomputingpa2
20/05/06 13:24:12 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
+-----+
|quality|          features|
+-----+
5|[7.4,0.7,0.0,1.9,...|
5|[7.8,0.88,0.0,2.6,...|
5|[7.8,0.76,0.04,2.0,...|
6|[11.2,0.28,0.56,1.0,...|
5|[7.4,0.7,0.0,1.9,...|
5|[7.4,0.66,0.0,1.8,...|
5|[7.9,0.6,0.06,1.6,...|
7|[7.3,0.65,0.0,1.2,...|
7|[7.8,0.58,0.02,2.0,...|
5|[7.5,0.5,0.36,6.1,...|
5|[6.7,0.58,0.08,1.0,...|
5|[7.5,0.5,0.36,6.1,...|
5|[5.6,0.615,0.0,1.0,...|
5|[7.8,0.61,0.29,1.0,...|
5|[8.9,0.62,0.18,3.0,...|
5|[8.9,0.62,0.19,3.0,...|
7|[8.5,0.28,0.56,1.0,...|
5|[8.1,0.56,0.28,1.0,...|
4|[7.4,0.59,0.08,4.0,...|
6|[7.9,0.32,0.51,1.0,...|
6|[8.6,0.49,0.28,1.0,...|
5|[7.7,0.49,0.26,1.0,...|
5|[9.3,0.39,0.44,2.0,...|
5|[7.0,0.62,0.08,1.0,...|
5|[7.9,0.52,0.26,1.0,...|
6|[8.6,0.49,0.28,1.0,...|
5|[8.6,0.49,0.29,2.0,...|
5|[7.7,0.49,0.26,1.0,...|
4|[5.0,1.02,0.04,1.0,...|
6|[4.7,0.6,0.17,2.3,...|
5|[6.8,0.775,0.0,3.0,...|
5|[7.0,0.5,0.25,2.0,...|
+-----+

6|[7.0,0.38,0.49,2.0,...|
6|[8.2,0.42,0.49,2.0,...|
5|[9.9,0.63,0.24,2.0,...|
6|[9.1,0.22,0.24,2.0,...|
5|[11.9,0.38,0.49,2.0,...|
5|[11.9,0.38,0.49,2.0,...|
6|[10.3,0.27,0.24,2.0,...|
6|[10.0,0.48,0.24,2.0,...|
6|[9.1,0.22,0.24,2.0,...|
5|[9.9,0.63,0.24,2.0,...|
6|[8.1,0.825,0.24,2.0,...|
7|[12.9,0.35,0.49,5.0,...|
5|[11.2,0.5,0.74,5.0,...|
5|[9.2,0.59,0.24,3.0,...|
6|[9.5,0.46,0.49,6.0,...|
5|[9.3,0.715,0.24,2.0,...|
6|[11.2,0.66,0.24,2.0,...|
6|[14.3,0.31,0.74,1.0,...|
5|[9.1,0.47,0.49,2.0,...|
6|[7.5,0.55,0.24,2.0,...|
6|[10.6,0.31,0.49,2.0,...|
6|[12.4,0.35,0.49,2.0,...|
6|[9.0,0.53,0.49,1.0,...|
6|[6.8,0.51,0.01,2.0,...|
6|[9.4,0.43,0.24,2.0,...|
6|[9.5,0.46,0.24,2.0,...|
5|[5.0,1.04,0.24,1.0,...|
5|[15.5,0.645,0.49,...|
5|[15.5,0.645,0.49,...|
6|[10.9,0.53,0.49,4.0,...|
5|[15.6,0.645,0.49,...|
6|[10.9,0.53,0.49,4.0,...|
6|[13.0,0.47,0.49,4.0,...|
+-----+

By calculating accuracy Test Error = 0.45
accuracy accuracy: 0.55
Test Error = 0.476781 with f1 score
f1 :: 0.5232192995603516
[hadoop@ip-172-31-15-95 ~]$
```

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

The screenshot shows the AWS S3 console interface. At the top, there's a navigation bar with the AWS logo, 'Services', 'Resource Groups', and a user profile. A blue notification banner at the top states: 'We're gradually updating the design of the Amazon S3 console. You will notice some updated screens as we improve the performance and user interface. To help us improve the experience, give feedback on the recent updates.' The left sidebar contains a menu with 'Amazon S3' (selected), 'Buckets', 'Batch Operations', 'Access analyzer for S3', 'Block public access (account settings)', and 'Feature spotlight 2'. The main content area is titled 'Amazon S3' and shows 'Buckets (1)'. It includes a search bar 'Find bucket by name', a table of buckets, and a 'Create bucket' button. The table has columns for Name, Region, Access, and Bucket created. One bucket is listed: 'aws-logs-133511003762-us-east-1' in 'US East (N. Virginia) us-east-1' region, with 'Objects can be public' access and a creation time of '2020-04-30T00:40:15.000Z'. The bottom of the page has a footer with 'Feedback', 'English (US)', and copyright information.

| Name | Region | Access | Bucket created |
|---------------------------------|---------------------------------|-----------------------|--------------------------|
| aws-logs-133511003762-us-east-1 | US East (N. Virginia) us-east-1 | Objects can be public | 2020-04-30T00:40:15.000Z |

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.raai/logistic-regression-in-spark-ml-8a95b5f5434c>
- 2) <https://medium.com/@dhiraj.p.raai/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/>
- 10) <https://runawayhorse001.github.io/LearningApacheSpark/pyspark.pdf>
- 11) <https://stackoverflow.com/questions/30063907/using-docker-compose-how-to-execute-multiple-commands>
- 12) <https://mlinproduction.com/docker-for-ml-part-4/>
- 13) <https://medium.com/@thiagolcmelo/submitting-a-python-job-to-apache-spark-on-docker-b2bd19593a06>
- 14) <https://docs.docker.com/engine/reference/commandline/run/>