Programming Assignment-2 CS643-Cloud Computing

Submitted By: Akash Shrivastava UCID: AS57

Section-1

GitHub URLs:

Parallel training implementation-

https://github.com/akatast/Wine quality prediction with Spark/blob/main/wineQPredModelTraining.py

Single machine prediction application-

https://github.com/akatast/Wine quality prediction with Spark/blob/main/wineQPredMo delValidation.py

Docker hub URL:

Docker container for prediction application-

https://hub.docker.com/repository/docker/as5721/as57dockerpublic

Command to execute docker container using input file-

 $sudo\ docker\ run\ -it\ -v\ `pwd`/TestDataset.csv:/dataset/TestDataset.csv\ as 5721/as 57 docker public: test-wine-qp\ /dataset/TestDataset.csv$

Please make sure docker is started and input file TestDataset.csv is available at the same directory where this command is being submitted.

Section-2

AWS Cloud environment set-up

2.1 AWS educate account:

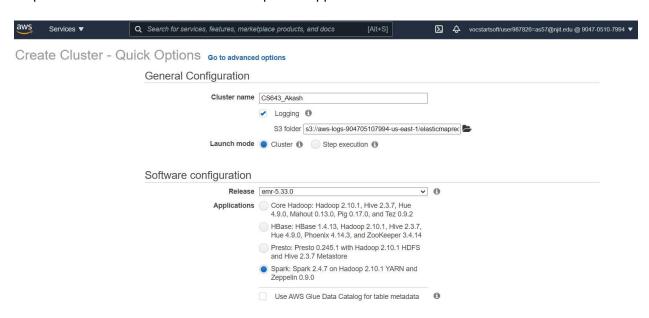
As very first step, login to AWS educate classroom (vocareum) and click on AWS Console.

2.2 Create EMR cluster:

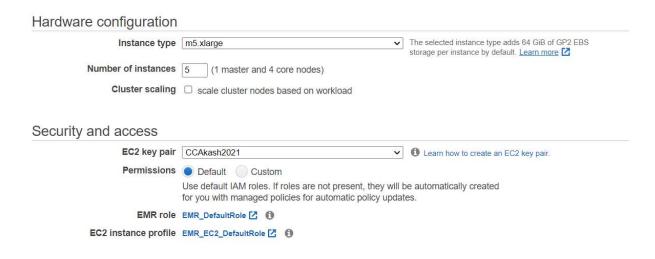
Step 1: Search for the EMR (Amazon Elastic MapReduce) and click on create cluster:



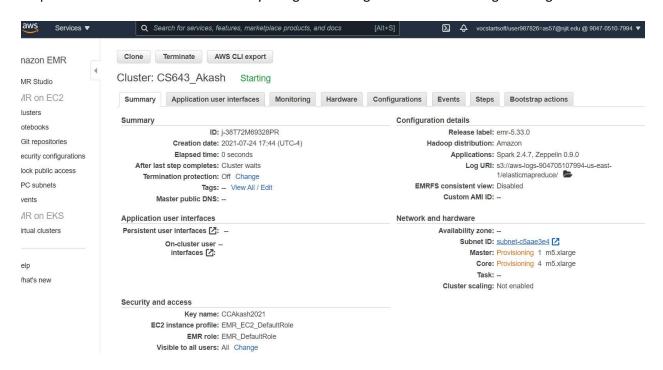
Step 2: Give the cluster name and select Spark as application:



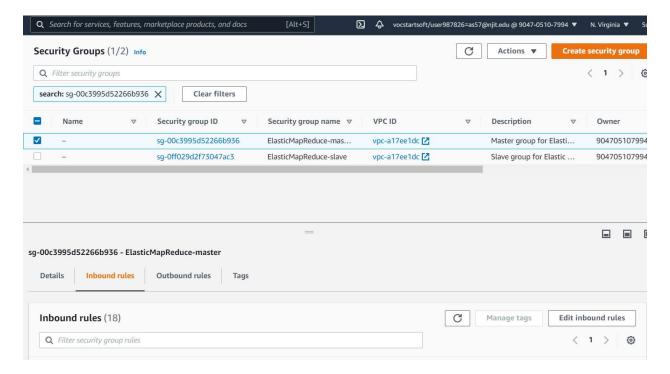
As mentioned in the requirement, use number of instances as 4 (one master and four core) and select the key pair:



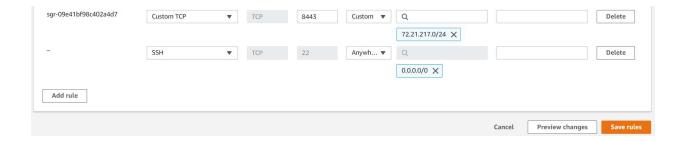
Step 3: Then click on create cluster and you'll get following screen with "Starting" message:



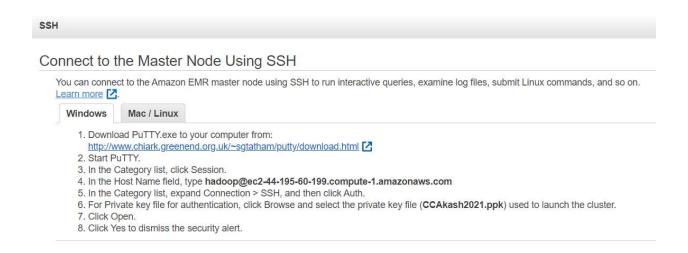
Step 4: Go to security group of master and click on inbound rules:



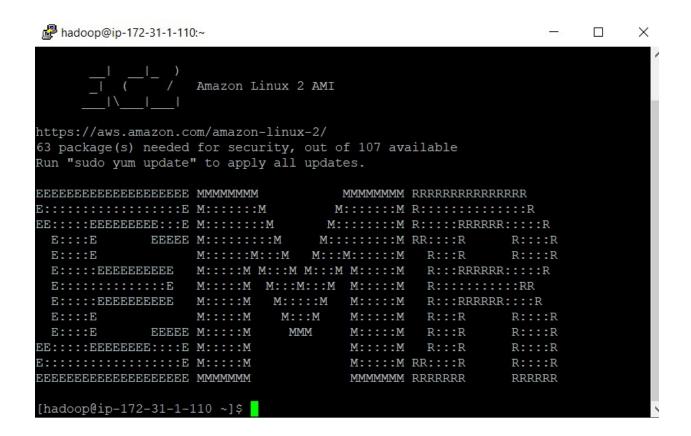
Click on 'Edit inbound rules' and add a new rule with type = SSH, port range=22 and source = anywhere



Step 5: AWS gives the steps to Connect to the Master Node Using SSH. Use the following steps to connect to the master node:



Step 6: After successful connection we'll get following screen:



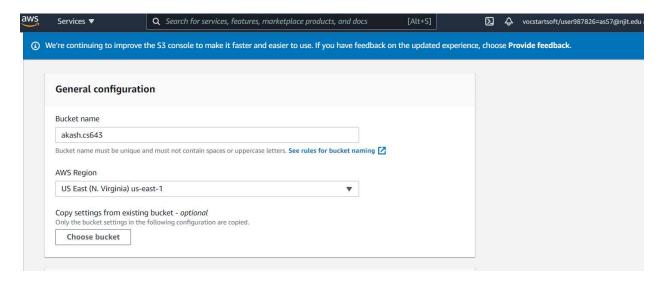
Section-3

S3 bucket and file storage

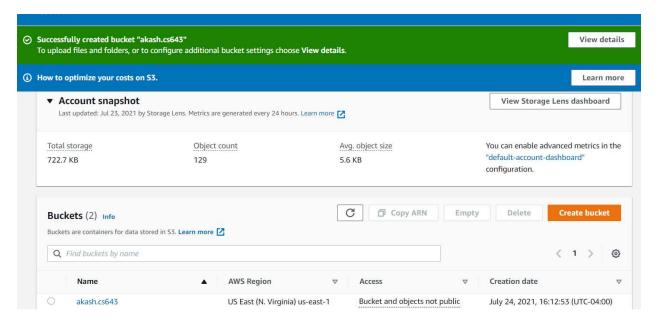
After termination of EMR cluster we lose all the data so we can use S3 bucket to store the files beyond one session/longer-term.

3.1 Create a new S3 bucket:

Create a S3 bucket to store the files-

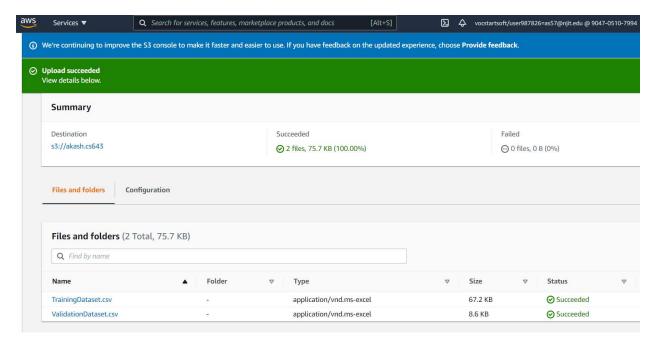


We'll get following success message-



3.2 Upload the files:

Upload the training and validation datasets to S3 bucket-



Section-4

Model training and validation

4.1 Parallel training implementation (Train the model):

Connect to the Master node, move the model training program to Master node and give the execute permission.

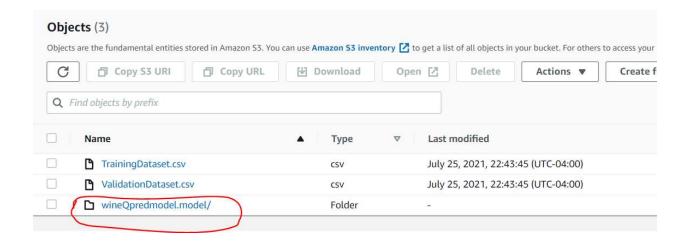
Please check page-1 for GitHub link of Parallel training implementation.

Execute the training program using following command-

Spark-submit wineQPredModelValidation.py

I am printing training error and F1 score after completion of training.

After successful execution, check the S3 bucket if model has been stored-



4.2 Single machine prediction application (without docker):

Please check page-1 for GitHub link of Single machine prediction application.

Execute the validation program:

spark-submit wineQPredModelValidation.py

Section-5 Create Docker Container

Connect to EC-2 instance of master and perform following steps:

```
Install spark-
```

pip install --user pyspark

Set the path-

export PYSPARK_PYTHON=/usr/bin/python3
export PYSPARK DRIVER PYTHON=/usr/bin/python3

Install and start the Docker-

sudo yum update -y
sudo amazon-linux-extras install docker
sudo yum install docker
sudo service docker start
sudo usermod -a -G docker Hadoop

Change the permission-

sudo chmod 666 /var/run/docker.sock

Build the image (before executing this command, make sure the trained model, model testing program and the dockerfile are present on same directory)-

```
sudo docker build -t imagename .
```

List the available images and verify-

sudo docker image ls

Login to docker hub using following command, username, and password-docker login

Go to dockerhub website and create a new public repo.

On the EC2 terminal tag the created docker image-

docker tag imagename: version username/repo:tagname

From EC2 push the image to docker hub repo-

docker push username/repo:tagname

Now docker container is ready, anyone can issue a pull request and execute it in same OS (Linux) using the input file TestDataset.csv

Docker container for prediction application:

Following is command to execute the container-

sudo docker run -it -v `pwd`/TestDataset.csv:/dataset/TestDataset.csv
as5721/as57dockerpublic:test-wine-qp /dataset/TestDataset.csv