CS643859
Programming Assignment 2
Cloud Computing

Name - Siddharth Pradhan
UCID - sp2882

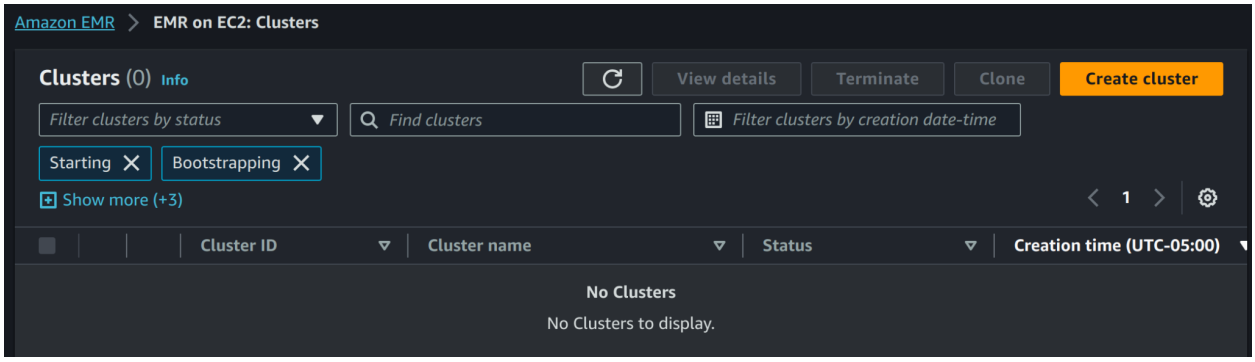
GitHub Link - https://github.com/siddharthpradhan20/CS643_Programming_assignment_2
DockerHub Link -
https://hub.docker.com/repository/docker/sidp20/cs643-programming-assignment-2/general

1. Login to AWS Console.

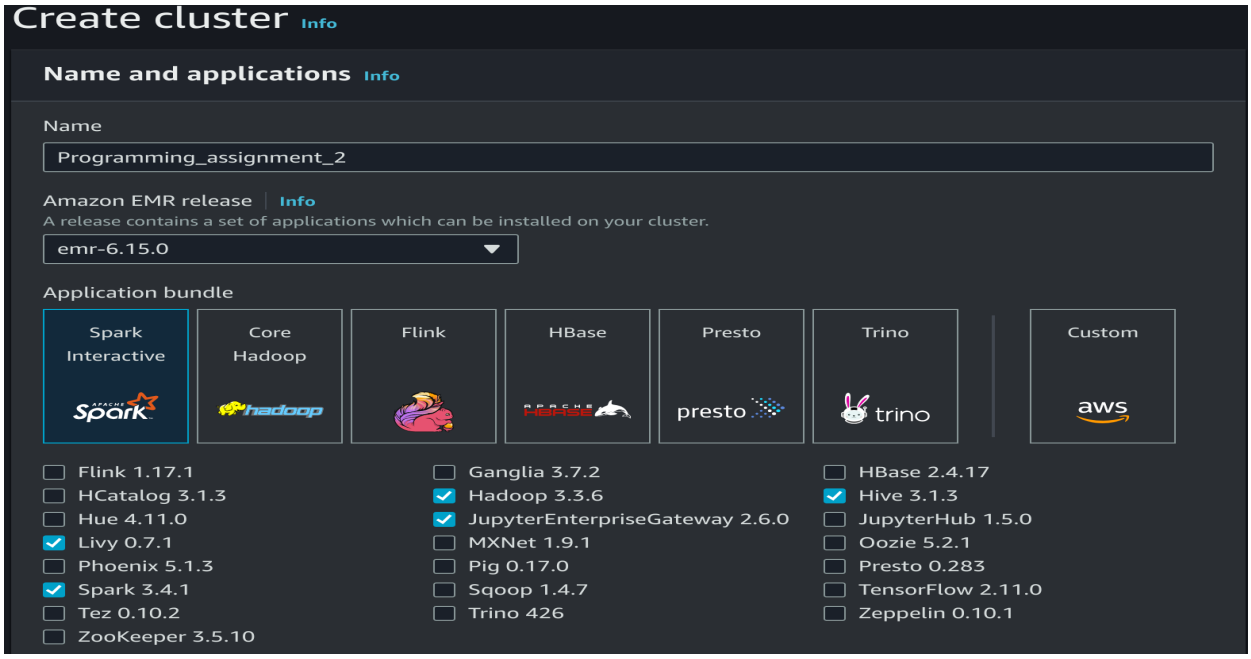
2. From the list of AWS services, select AWS EMR service and then select EMR on EC2 Clusters.

3. The Clusters page will appear. You can see that there are no active clusters. We have to create a cluster.

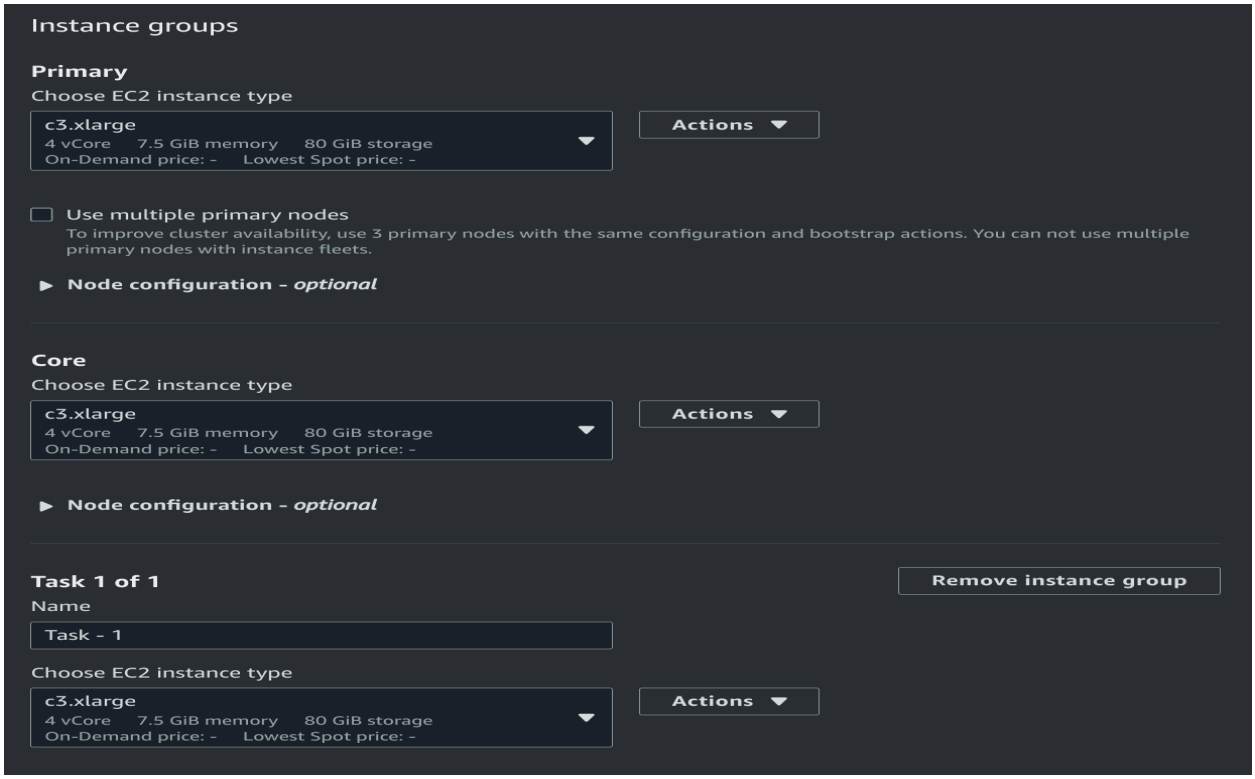
4. To create a cluster, click on the Create cluster button.



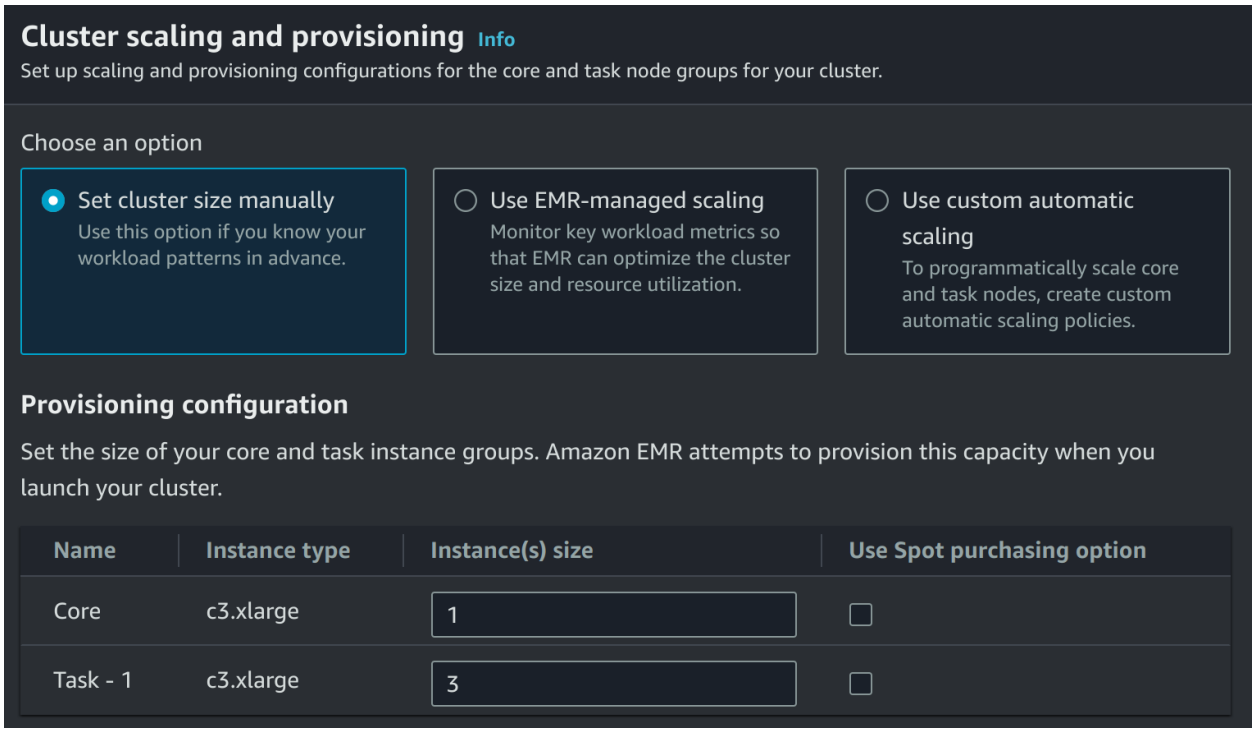
5. Give any name to the cluster you want. Also, make sure that the EMR version should be the latest version as shown in figure below. Also, select Spark Interactive under the Application bundle option.



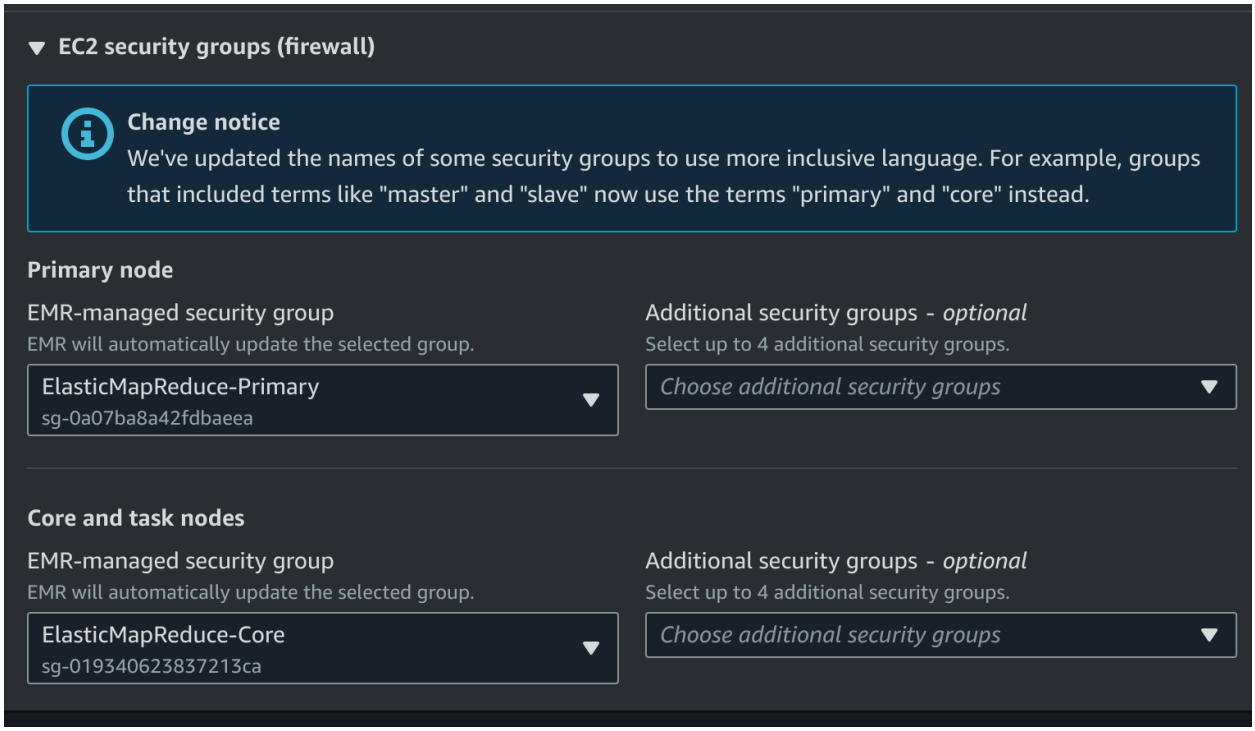
6. For instance groups in Primary, Core and Task, I have chosen c3.xlarge. You can choose any instance type.



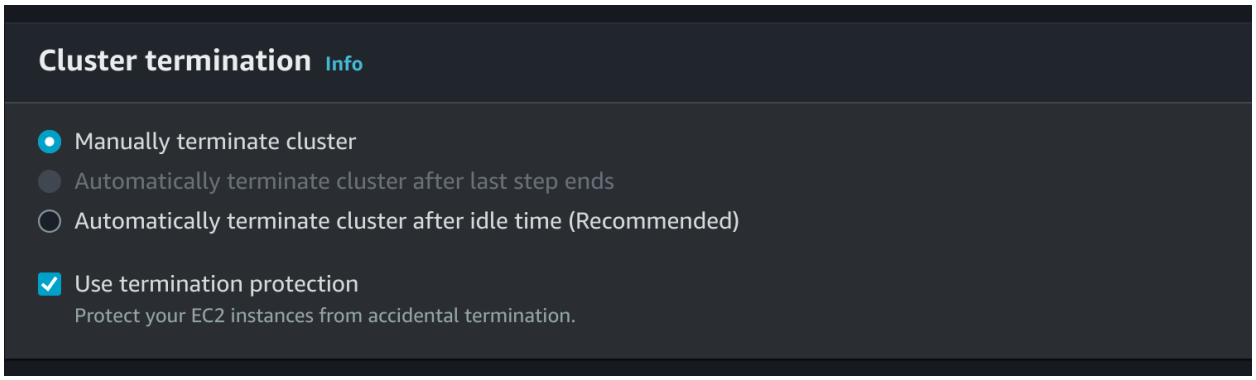
7. For cluster scaling and provisioning, let the Instance size for Core be 1 and for Task-2 be 3.



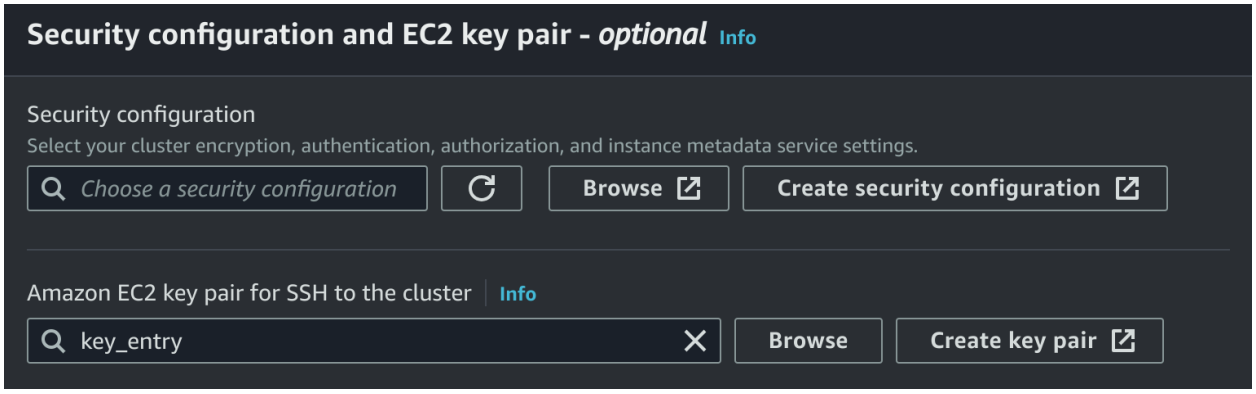
8. For EC2 security groups, select the security groups for Primary node and Core and task nodes as shown below.



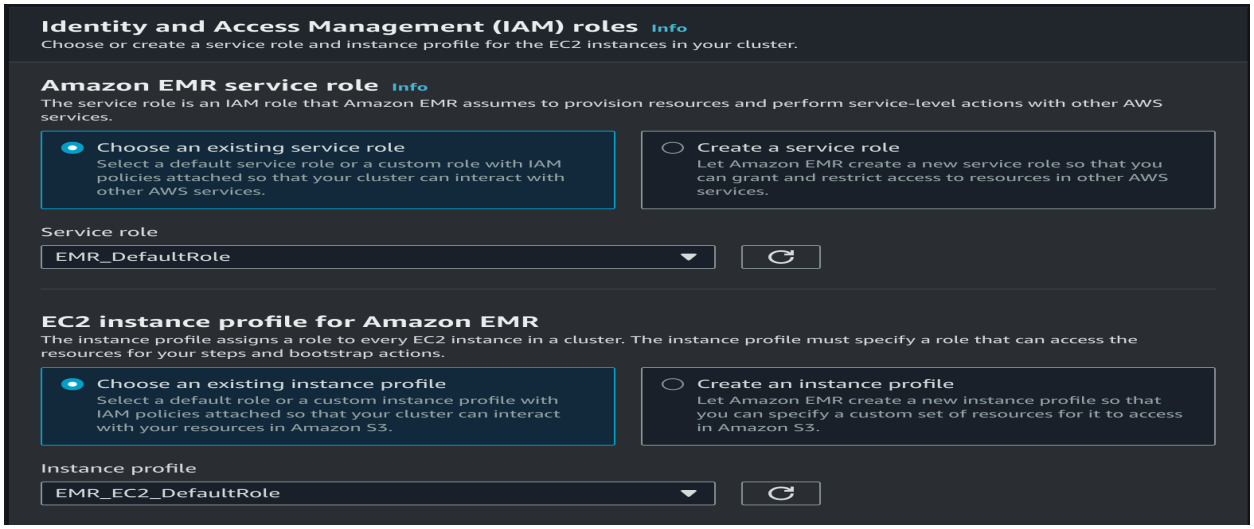
9. Make sure to select Manually terminate cluster option in order to prevent automatic termination of cluster. Although, this option is not recommended.



10. I have already created a key pair named “key_entry” and selected it as the key pair which enabled me to login to the cluster using SSH. You can create your own key pair and select here accordingly. Also, make sure to keep the .pem file of your key pair somewhere safe.

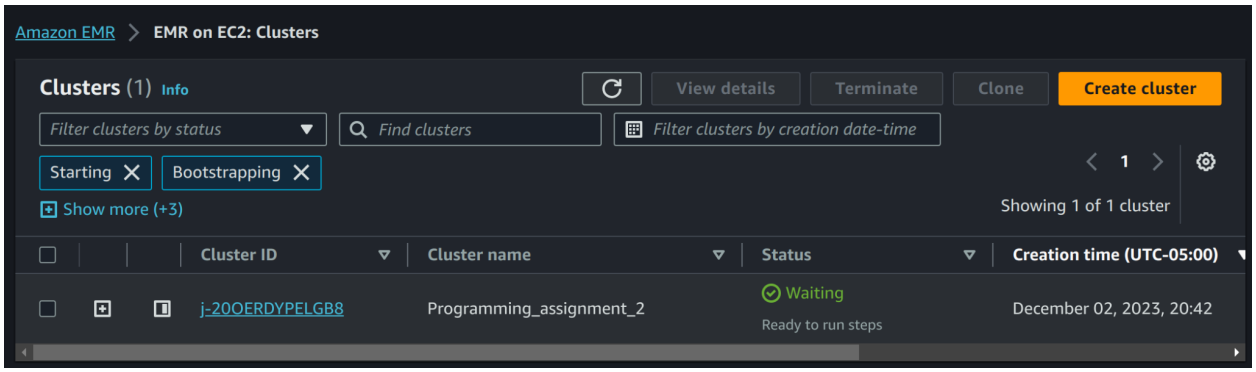


11. Select the IAM roles as shown below.

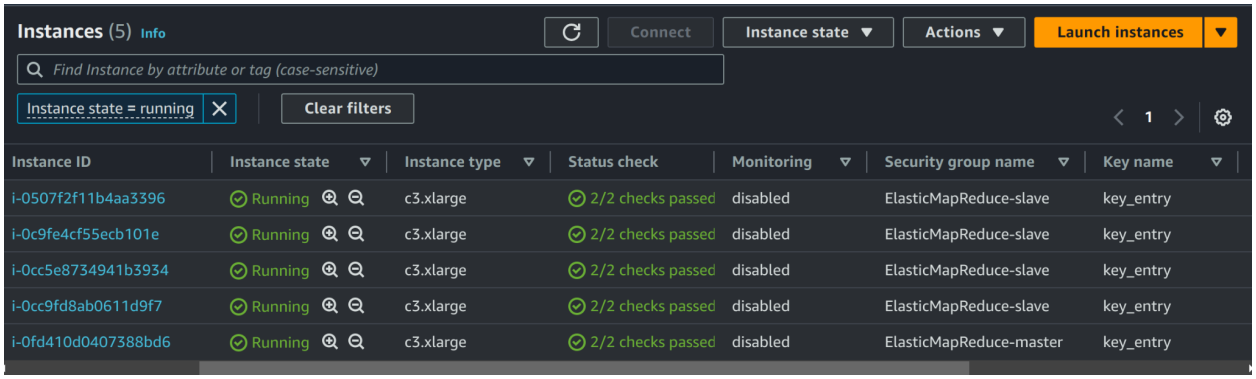


12. Click on the Create Cluster button to create the clusters.

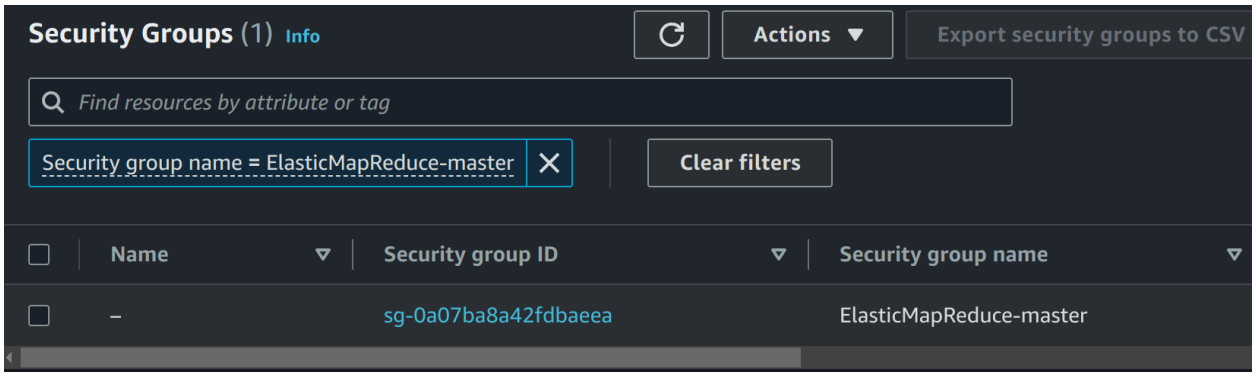
13. You can see that the cluster has been created on the Clusters page. Initially, the status of the cluster will be “Starting” and after some time, it will get changed to “Waiting” as shown below.



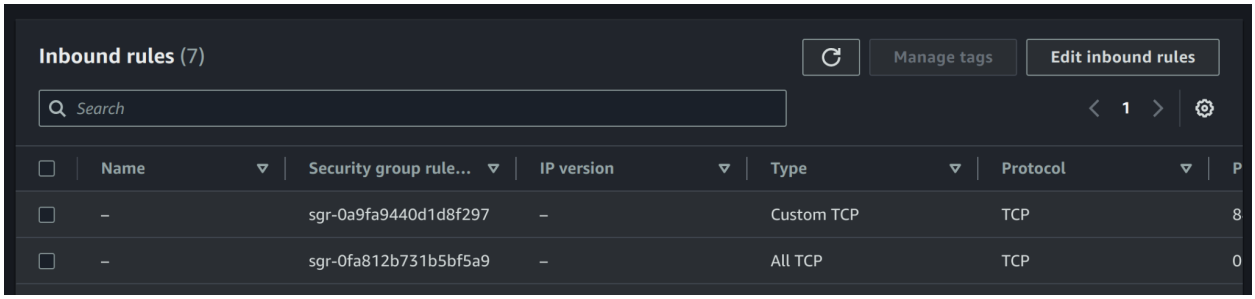
14. Go to the EC2 instances page. Here you can see that in total there are 5 EC2 instances launched and out of those, one is Master node while other four are Slave nodes as shown below.



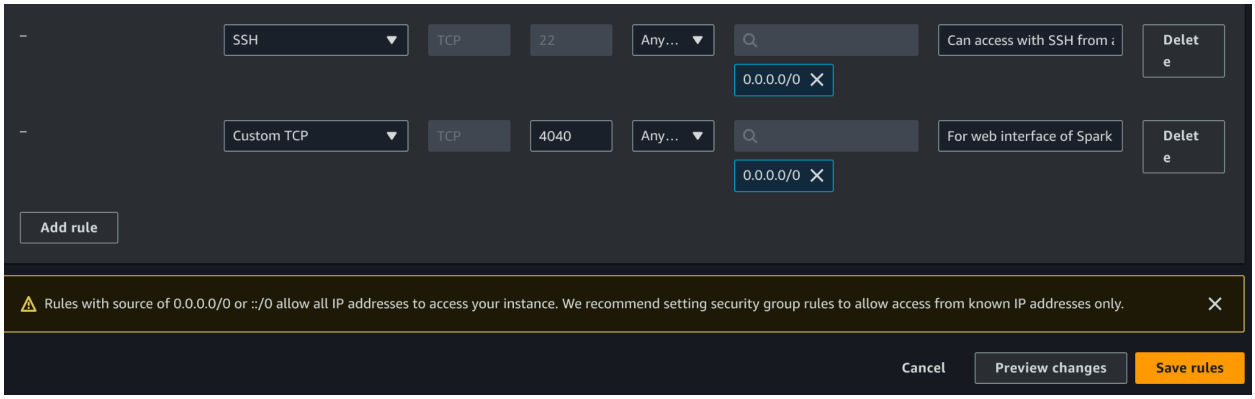
15. Now, in EC2 service, find “ElasticMapReduce-Master” security group and click on its respective Security ID.



16. In the Inbound Rules section, click Edit inbound rules.



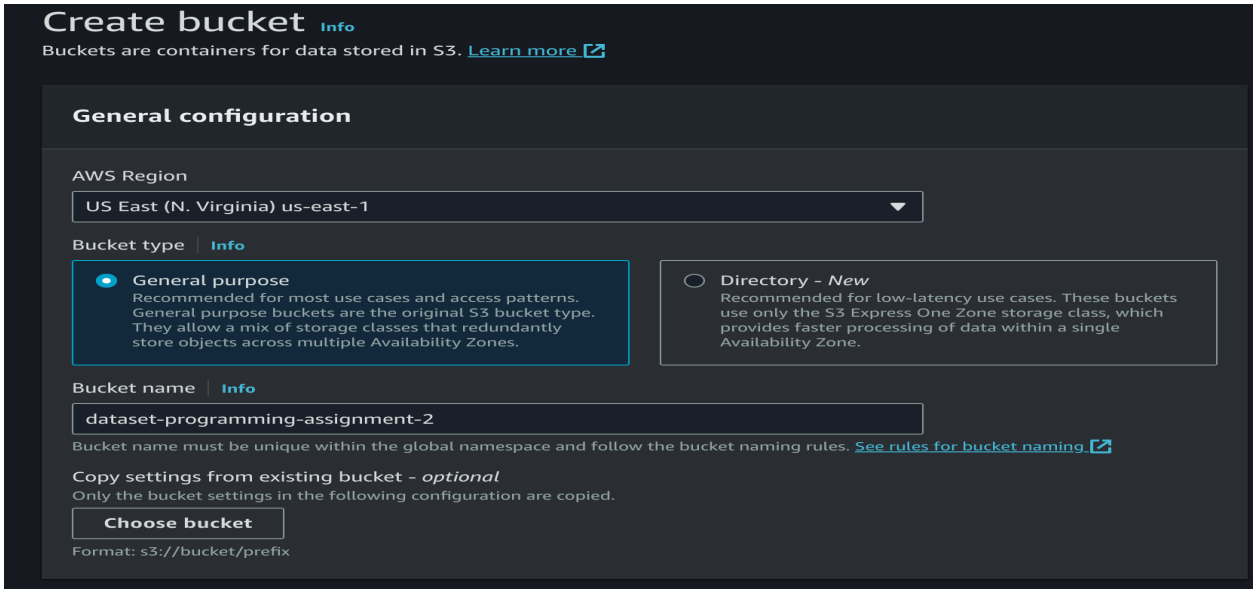
17. Click on Add rule button and add the 2 port numbers, 22 and 4040 with settings shown below and then click on Save rules.



18. Now, in AWS services, go to S3 to create an S3 bucket to store the dataset.

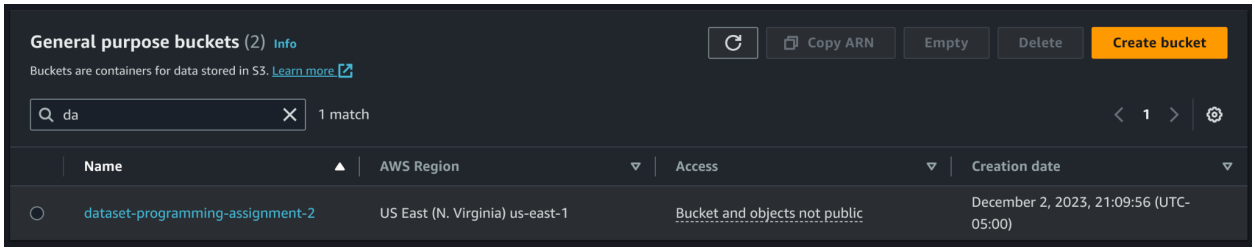
19. Click on “Create Bucket”.

20. Give your bucket name as “dataset-programming-assignment-2”. Scroll down the page and click on the “Create bucket” button.

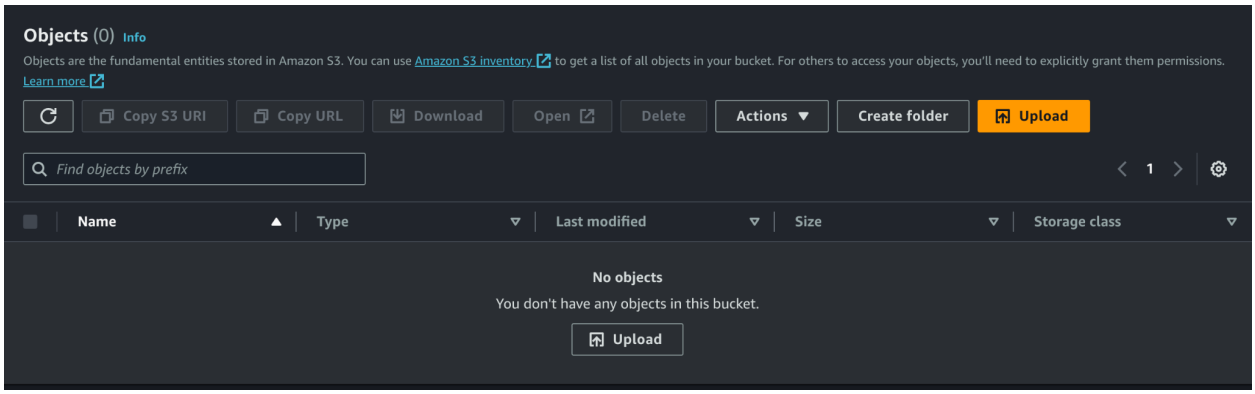


21. In the buckets page, you can see the bucket that you have created.

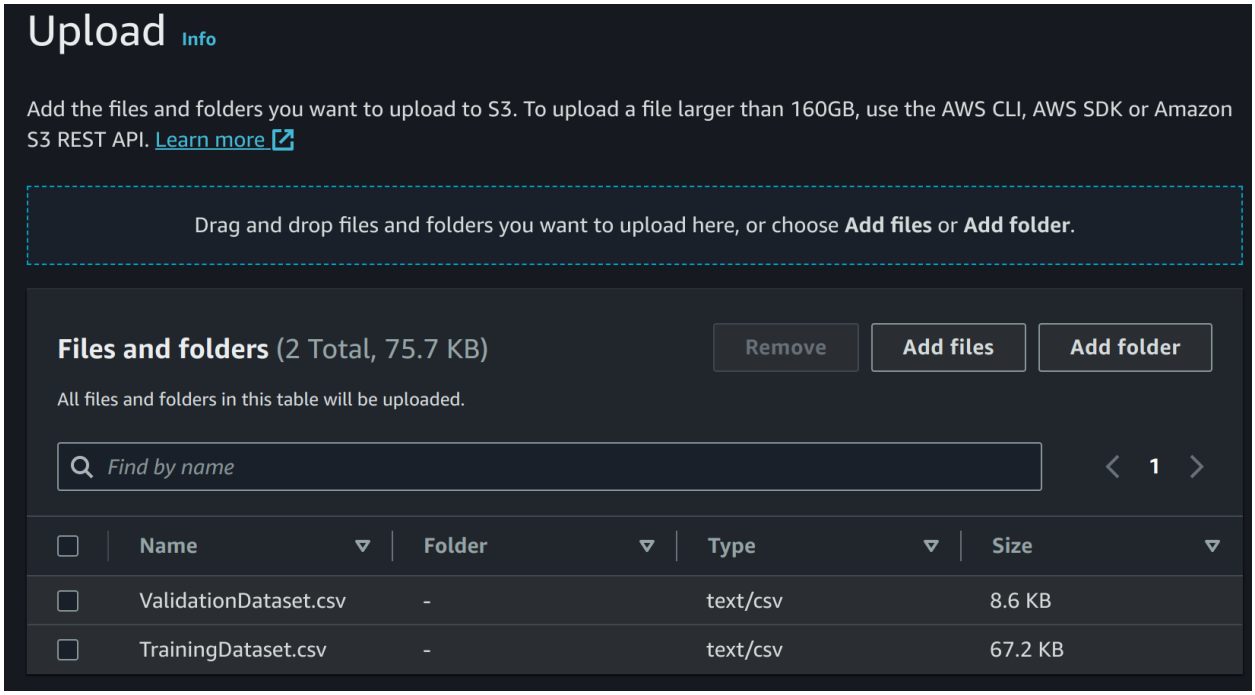
22. Click on the created bucket's name.



23. Click on upload button.



24. Click on add files and select the .csv files for the dataset from the GitHub repository. Now, click on the "Upload button" to upload the dataset in S3 bucket. Now, you have 2 .csv files stored in S3 bucket as ValidationDataset.csv and TrainingDataset.csv.



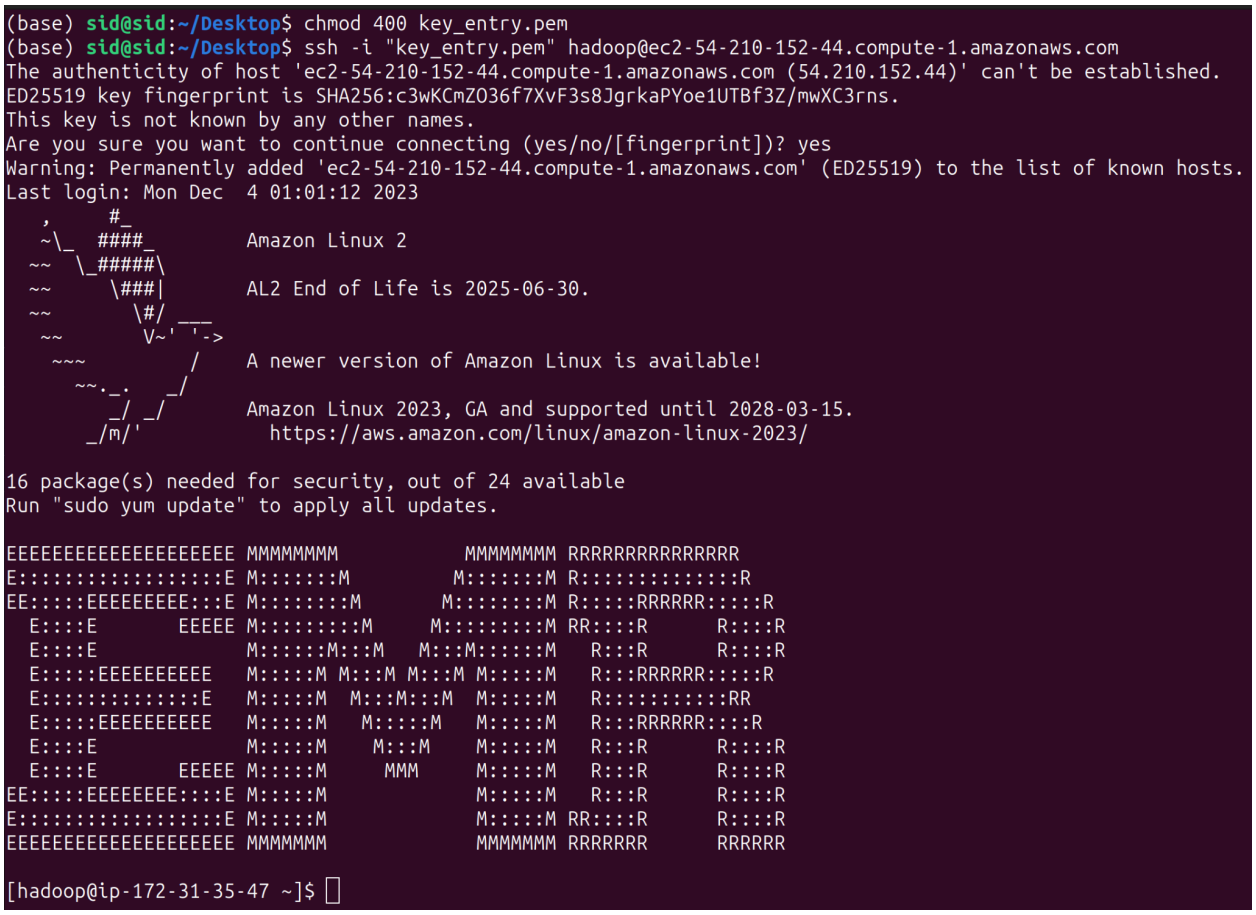
25. Now, open the terminal in that directory where you have saved your .pem file.

26. Run the following command -

chmod 400 key_entry.pem

Here, “key_entry” is the name of the .pem file that I have created. Replace it with your file accordingly.

27. Now, SSH with that .pem file into the master node using “hadoop” as the user as shown below. You can look at your login details on your EC2 instance page for Master Node.



28. You’ll get logged into the master node with “hadoop” as a user.

29. Now, for credentials setup, we need to configure credentials in Master node EC2 instance.

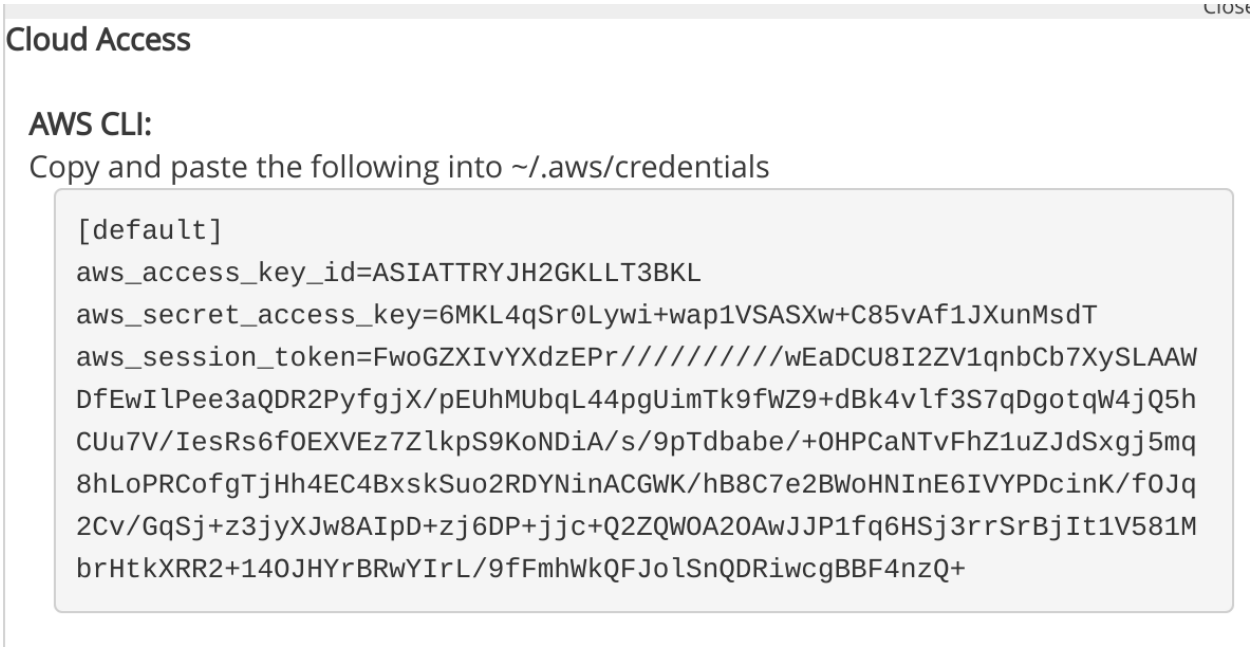
To configure credentials, enter the following commands in your terminal for the Master node:

mkdir .aws

touch .aws/credentials

vi .aws/credentials

30. Copy and paste the credentials which can be found in the AWS Academy page and under AWS details.



31. Now, we need to install the relevant packages in order to run our Spark Application.
Run the following commands:
sudo yum update
sudo yum install git
pip install pyspark findspark boto3 numpy pandas scikit-learn datetime

32. Now, clone the GitHub repository by entering the following command:
#git clone https://github.com/siddharthpradhan20/CS643_Programming_assignment_2.git

33. Now, run the following commands to get started with Spark Application:
spark-submit --master yarn CS643_Programming_assignment_2/WineTraining.py
spark-submit --master yarn CS643_Programming_assignment_2/WineTesting.py > output.txt
The above command will save the output in the output.txt file.

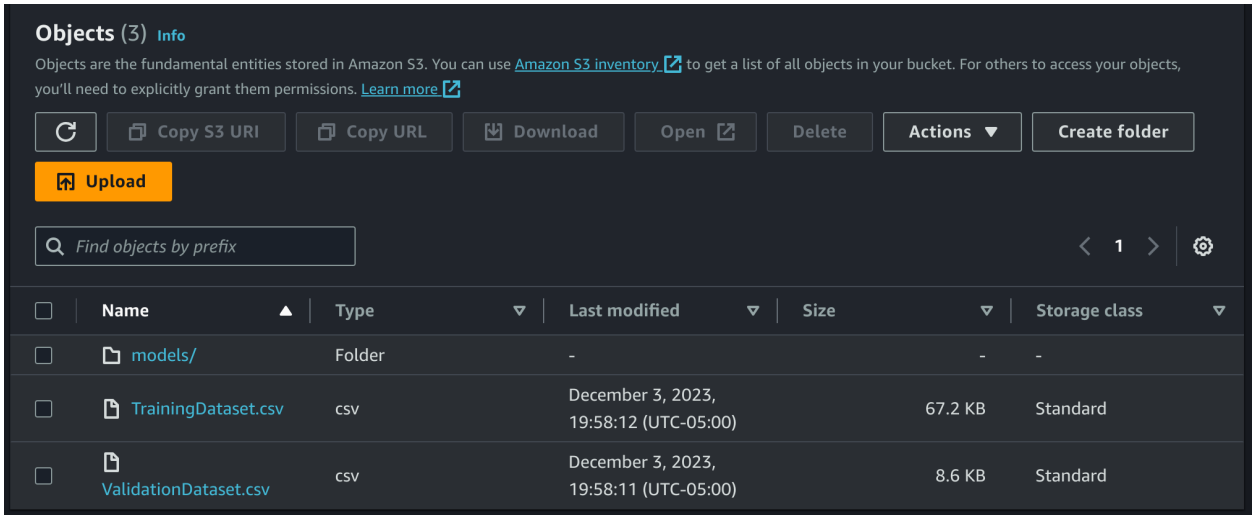
34. Now, to view results, run the following command:

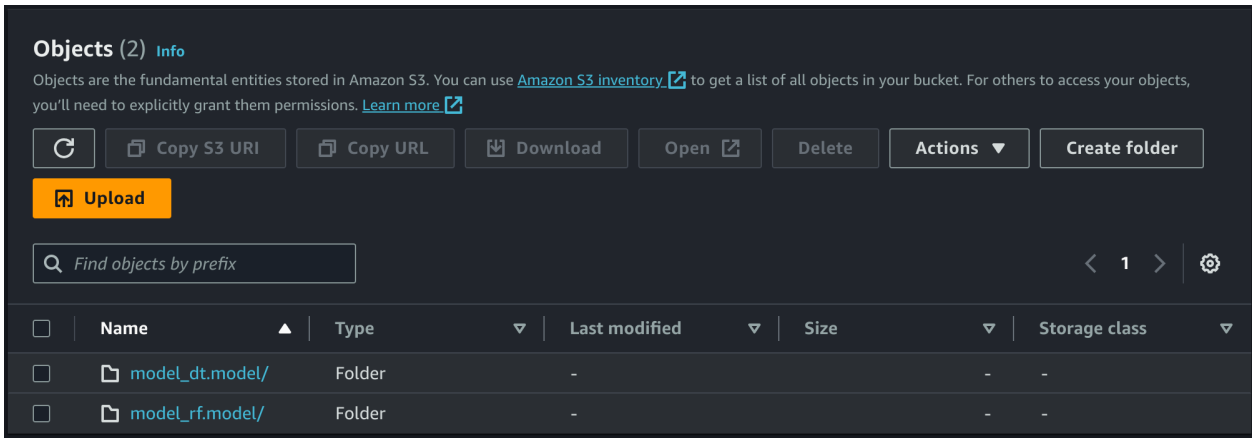
cat output.txt | grep F1

You will get the results as shown below with Accuracy and F1 scores of the applied Machine Learning algorithms.

```
[hadoop@ip-172-31-32-221 CS643-WinePrediction]$ cat output.txt | grep F1
Decision Tree Model - Accuracy: 0.475, F1 Score: 0.47300154672977135
Random Forest Model - Accuracy: 0.5, F1 Score: 0.5126602345173041
```

35. Go to AWS Dashboard and then from S3 services, select the Bucket that you have created. Here, you can see that a new folder called “models/” has been created which contains the trained models as shown below.





DOCKER IMPLEMENTATION -

1. In the Master Node terminal, run the following command:
   # sudo service docker status
   The above command will show the status of the docker application. If it shows status as "inactive", then you need to start the service.

2. To start Docker, run the following command:
   # sudo docker service start

3. To verify the status, run the following command again:
   # sudo service docker status
   Docker service is active now.

```
[hadoop@ip-172-31-32-221 CS643-WinePrediction]$ sudo service docker status
Redirecting to /bin/systemctl status docker.service
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor p
et: disabled)
   Active: active (running) since Mon 2023-12-04 03:30:42 UTC; 6s ago
     Docs: https://docs.docker.com
  Process: 16625 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (c
=exited, status=0/SUCCESS)
  Process: 16562 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=
UCCESS)
 Main PID: 16640 (dockerd)
    Tasks: 9
   Memory: 90.0M
```

4. In the GitHub repo, you will find a file called "Dockerfile".
   To build docker container from Dockerfile, run the following command:
   # sudo docker build -t sidp20/cs643-programming-assignment-2 .
   This command will build the docker docker image stored in your instance.

5. To verify if Docker image is created, run the following command:
   # sudo docker image ls

```
[hadoop@ip-172-31-32-221 CS643-WinePrediction]$ sudo docker image ls
REPOSITORY                              TAG       IMAGE ID       CREATED
  SIZE
sidp20/cs643-programming-assignment-2   latest    f86a747d99e7   12 minutes ago
  2.39GB
```

   Here, you can see that your docker image has been created.

6. To run this docker image, run the following command:
   # sudo docker run -it sidp20/cs643-programming-assignment-2
   Here, instead of using image name, you can also use your image ID
   # sudo docker run -it <IMAGE_ID>

7. This will give the same output with Accuracy and F1 Scores.

8. To push the generated docker image in DockerHub repository, run the following command:
   # sudo docker push sidp20/cs643-programming-assignment-2

9. Now, you can download that docker image from the DockerHub repository and run that image using the instructions given in the DockerHub page.