

**CS643859**  
**Programming Assignment 2**  
**Cloud Computing**

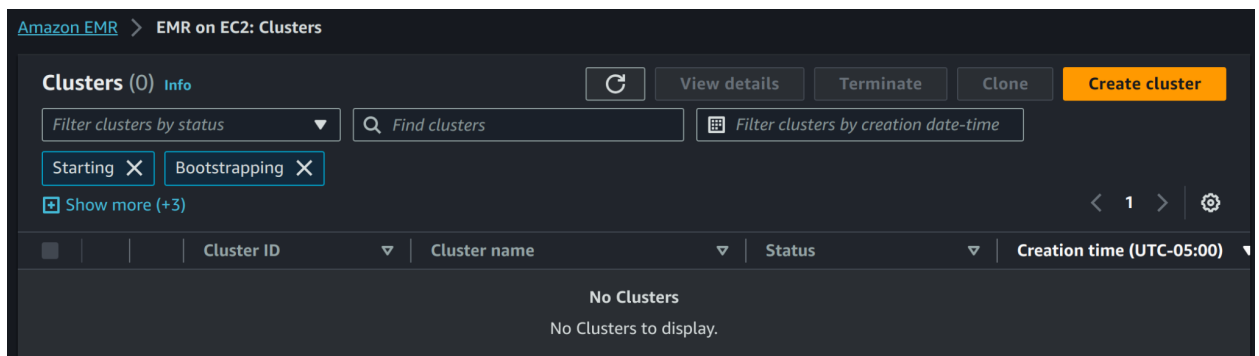
**Name - Siddharth Pradhan**  
**UCID - sp2882**

**GitHub Link -** [https://github.com/siddharthpradhan20/CS643\\_Programming\\_assignment\\_2](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.

## Create cluster [Info](#)

### Name and applications [Info](#)

Name


Programming\_assignment\_2


Amazon EMR release [Info](#)


A release contains a set of applications which can be installed on your cluster.


emr-6.15.0


Application bundle


Spark  
Interactive  



Core  
Hadoop  


Flink  


HBase  


Presto  


Trino  


Custom  


☐ Flink 1.17.1
 ☐ HCatalog 3.1.3
 ☐ Hue 4.11.0
 ☒ Livy 0.7.1
 ☐ Phoenix 5.1.3
 ☒ Spark 3.4.1
 ☐ Tez 0.10.2
 ☐ ZooKeeper 3.5.10
 ☐ Ganglia 3.7.2
 ☒ Hadoop 3.3.6
 ☒ JupyterEnterpriseGateway 2.6.0
 ☐ MXNet 1.9.1
 ☐ Pig 0.17.0
 ☐ Sqoop 1.4.7
 ☐ Trino 426
 ☐ HBase 2.4.17
 ☒ Hive 3.1.3
 ☐ JupyterHub 1.5.0
 ☐ Oozie 5.2.1
 ☐ Presto 0.283
 ☐ TensorFlow 2.11.0
 ☐ Zeppelin 0.10.1

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

### Instance groups

#### Primary

Choose EC2 instance type

c3.xlarge  
4 vCore 7.5 GiB memory 80 GiB storage  
On-Demand price: - Lowest Spot price: -

Actions

☐ Use multiple primary nodes  
To improve cluster availability, use 3 primary nodes with the same configuration and bootstrap actions. You can not use multiple primary nodes with instance fleets.

► Node configuration - optional

#### Core

Choose EC2 instance type

c3.xlarge  
4 vCore 7.5 GiB memory 80 GiB storage  
On-Demand price: - Lowest Spot price: -

Actions

► Node configuration - optional

#### Task 1 of 1

Name

Task - 1

Choose EC2 instance type

c3.xlarge  
4 vCore 7.5 GiB memory 80 GiB storage  
On-Demand price: - Lowest Spot price: -

Actions

Remove instance group

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

### Cluster scaling and provisioning [Info](#)

Set up scaling and provisioning configurations for the core and task node groups for your cluster.

Choose an option

☒ **Set cluster size manually**  
Use this option if you know your workload patterns in advance.

☐ **Use EMR-managed scaling**  
Monitor key workload metrics so that EMR can optimize the cluster size and resource utilization.

☐ **Use custom automatic scaling**  
To programmatically scale core and task nodes, create custom automatic scaling policies.


### Provisioning configuration

Set the size of your core and task instance groups. Amazon EMR attempts to provision this capacity when you launch your cluster.

Name	Instance type	Instance(s) size	Use Spot purchasing option
Core	c3.xlarge	<input type="text" value="1"/>	<input type="checkbox"/>
Task - 1	c3.xlarge	<input type="text" value="3"/>	<input type="checkbox"/>

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

▼ **EC2 security groups (firewall)**

**Change notice**  
We've updated the names of some security groups to use more inclusive language. For example, groups that included terms like "master" and "slave" now use the terms "primary" and "core" instead.

### Primary node

EMR-managed security group  
EMR will automatically update the selected group.

ElasticMapReduce-Primary  
sg-0a07ba8a42fdbaeaa ▼

Additional security groups - *optional*  
Select up to 4 additional security groups.

Choose additional security groups ▼

### Core and task nodes

EMR-managed security group  
EMR will automatically update the selected group.

ElasticMapReduce-Core  
sg-019340623837213ca ▼

Additional security groups - *optional*  
Select up to 4 additional security groups.

Choose additional security groups ▼

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

**Cluster termination** [Info](#)

☒ Manually terminate cluster

☐ Automatically terminate cluster after last step ends

☐ Automatically terminate cluster after idle time (Recommended)

☒ Use termination protection  
Protect your EC2 instances from accidental termination.

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.

**Security configuration and EC2 key pair - optional** [Info](#)

**Security configuration**  
Select your cluster encryption, authentication, authorization, and instance metadata service settings.

**Amazon EC2 key pair for SSH to the cluster** [Info](#)

11. Select the IAM roles as shown below.

**Identity and Access Management (IAM) roles** [Info](#)

Choose or create a service role and instance profile for the EC2 instances in your cluster.

**Amazon EMR service role** [Info](#)  
The service role is an IAM role that Amazon EMR assumes to provision resources and perform service-level actions with other AWS services.

☒ Choose an existing service role  
Select a default service role or a custom role with IAM policies attached so that your cluster can interact with other AWS services.

☐ Create a service role  
Let Amazon EMR create a new service role so that you can grant and restrict access to resources in other AWS services.

**Service role**  
EMR\_DefaultRole

**EC2 instance profile for Amazon EMR**  
The instance profile assigns a role to every EC2 instance in a cluster. The instance profile must specify a role that can access the resources for your steps and bootstrap actions.

☒ Choose an existing instance profile  
Select a default role or a custom instance profile with IAM policies attached so that your cluster can interact with your resources in Amazon S3.

☐ Create an instance profile  
Let Amazon EMR create a new instance profile so that you can specify a custom set of resources for it to access in Amazon S3.

**Instance profile**  
EMR\_EC2\_DefaultRole

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.

The screenshot shows the Amazon EMR console's 'Clusters' page. At the top, there's a breadcrumb 'Amazon EMR > EMR on EC2: Clusters'. Below it, the page title is 'Clusters (1) Info'. There are buttons for 'View details', 'Terminate', 'Clone', and a prominent orange 'Create cluster' button. A search bar 'Find clusters' and a filter 'Filter clusters by status' are present. The status filter is set to 'Starting', but a 'Waiting' filter is also visible. A table lists the clusters:

Cluster ID	Cluster name	Status	Creation time (UTC-05:00)
j-20OERDYPELGB8	Programming_assignment_2	Waiting Ready to run steps	December 02, 2023, 20:42

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.

The screenshot shows the Amazon EC2 console's 'Instances' page. The page title is 'Instances (5) Info'. There are buttons for 'Connect', 'Instance state', 'Actions', and a prominent orange 'Launch instances' button. A search bar 'Find Instance by attribute or tag (case-sensitive)' and a filter 'Instance state = running' are present. A table lists the instances:

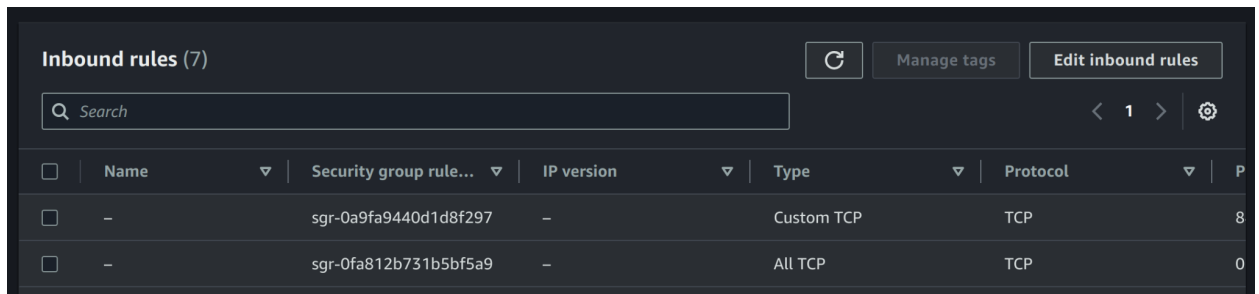
Instance ID	Instance state	Instance type	Status check	Monitoring	Security group name	Key name
i-0507f2f11b4aa3396	Running	c3.xlarge	2/2 checks passed	disabled	ElasticMapReduce-slave	key_entry
i-0c9fe4cf55ecb101e	Running	c3.xlarge	2/2 checks passed	disabled	ElasticMapReduce-slave	key_entry
i-0cc5e8734941b3934	Running	c3.xlarge	2/2 checks passed	disabled	ElasticMapReduce-slave	key_entry
i-0cc9fd8ab0611d9f7	Running	c3.xlarge	2/2 checks passed	disabled	ElasticMapReduce-slave	key_entry
i-0fd410d0407388bd6	Running	c3.xlarge	2/2 checks passed	disabled	ElasticMapReduce-master	key_entry

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

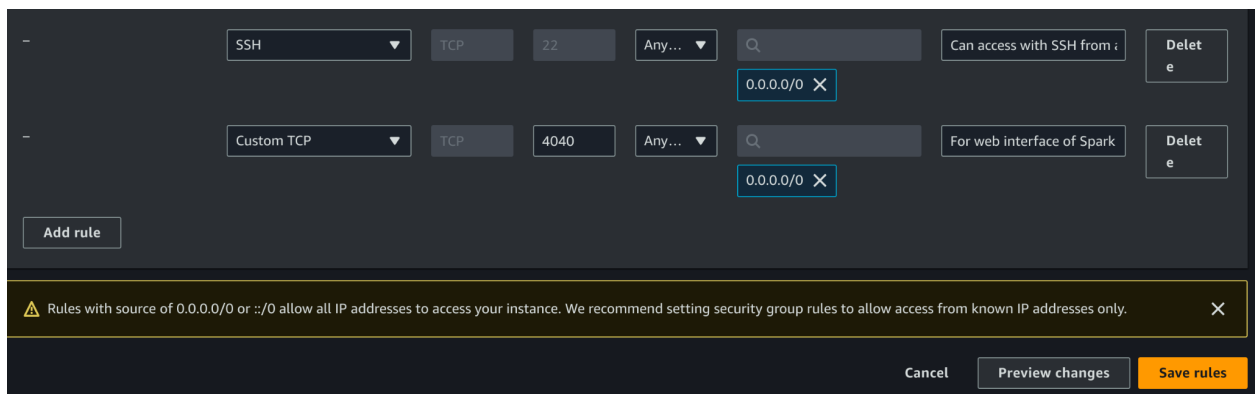
The screenshot shows the Amazon EC2 console's 'Security Groups' page. The page title is 'Security Groups (1) Info'. There are buttons for 'Actions' and 'Export security groups to CSV'. A search bar 'Find resources by attribute or tag' and a filter 'Security group name = ElasticMapReduce-master' are present. A table lists the security groups:

Name	Security group ID	Security group name
-	sg-0a07ba8a42fdbaeaa	ElasticMapReduce-master

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



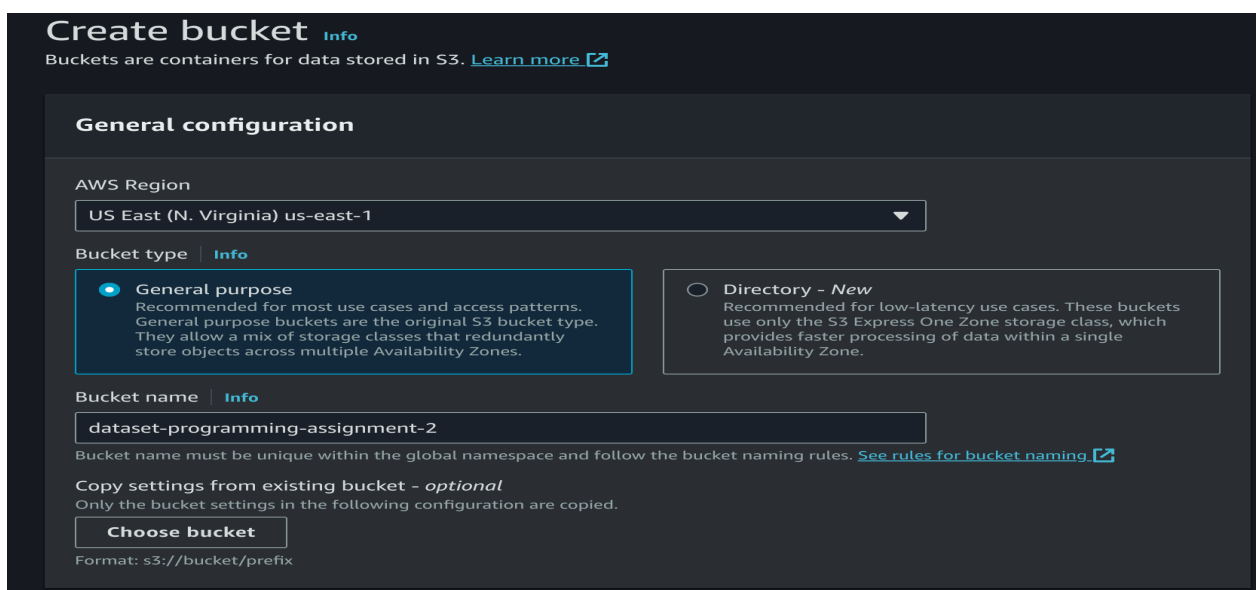
17. Click on Add rule button and add the 2 port numbers, 22 and 4040 with settings shown below and then click on Save rules.



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.

**General purpose buckets (2)** [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)

Search: da 1 match

Name	AWS Region	Access	Creation date
dataset-programming-assignment-2	US East (N. Virginia) us-east-1	Bucket and objects not public	December 2, 2023, 21:09:56 (UTC-05:00)

23. Click on upload button.

**Objects (0)** [Info](#)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Buttons: Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, Upload

Search: Find objects by prefix

**No objects**  
You don't have any objects in this bucket.

Upload

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.

**Upload** [Info](#)

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)

Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.

**Files and folders (2 Total, 75.7 KB)** [Remove](#) [Add files](#) [Add folder](#)

All files and folders in this table will be uploaded.

Search: Find by name

Name	Folder	Type	Size
ValidationDataset.csv	-	text/csv	8.6 KB
TrainingDataset.csv	-	text/csv	67.2 KB

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.

#### Cloud Access

##### AWS CLI:

Copy and paste the following into ~/.aws/credentials

```
[default]
aws_access_key_id=ASIATTRYJH2GKLLT3BKL
aws_secret_access_key=6MKL4qSr0Lywi+wap1VSASXw+C85vAf1JXunMsdT
aws_session_token=FwoGZXIvYXdzEPr//////////wEaDCU8I2ZV1qnbCb7XySLAAW
DfEwILPee3aQDR2PyfgjX/pEUhMUbqL44pgUimTk9fWZ9+dBk4vlf3S7qDgotqW4jQ5h
CUu7V/IesRs6f0EXVEz7ZlkpS9KoNDiA/s/9pTdbabe/+0HPCaNTvFhZ1uZJdSxgj5mq
8hLoPRCofgTjHh4EC4BxskSuo2RDYNinACGWK/hB8C7e2BWoHNInE6IVYpDcinK/f0Jq
2Cv/GqSj+z3jyXJw8AIpD+zj6DP+jjc+Q2ZQW0A20AwJJP1fq6HSj3rrSrBjIt1V581M
brHtkXRR2+140JHYrBRwYIrL/9fFmhWkQFJoLSnQDRiwcgBBF4nzQ+
```

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
[hadoop@ip-172-31-32-221 CS643-WinePrediction]$
```

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.

**Objects (3)** [Info](#)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

[Refresh](#) [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#)

[Upload](#)

| <input type="checkbox"/> | Name                                  | Type   | Last modified                          | Size    | Storage class |
|--------------------------|---------------------------------------|--------|----------------------------------------|---------|---------------|
| <input type="checkbox"/> | <a href="#">models/</a>               | Folder | -                                      | -       | -             |
| <input type="checkbox"/> | <a href="#">TrainingDataset.csv</a>   | csv    | December 3, 2023, 19:58:12 (UTC-05:00) | 67.2 KB | Standard      |
| <input type="checkbox"/> | <a href="#">ValidationDataset.csv</a> | csv    | December 3, 2023, 19:58:11 (UTC-05:00) | 8.6 KB  | Standard      |

**Objects (2)** [Info](#)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

[Refresh](#) [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#)

[Upload](#)

| <input type="checkbox"/> | Name                            | Type   | Last modified | Size | Storage class |
|--------------------------|---------------------------------|--------|---------------|------|---------------|
| <input type="checkbox"/> | <a href="#">model_dt.model/</a> | Folder | -             | -    | -             |
| <input type="checkbox"/> | <a href="#">model_rf.model/</a> | Folder | -             | -    | -             |

# 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.

```
Redirecting to /bin/systemctl start docker.service
[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 preset: 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 (code=exited, status=0/SUCCESS)
   Process: 16562 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
  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 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.