CS-643 Programming Assignment-2 sg2656

Wine Quality Prediction AWS Spark Application:

Pa2Winepred: The creation of a Python application using the PySpark interface is the goal of this project.

An Elastic MapReduce (EMR) cluster from Amazon Web Services (AWS) is used to host the application. The main goal is to use publicly available data to train a machine learning model in parallel on EC2 instances to predict wine quality. The trained algorithm is then used to forecast the wine's quality. The trained machine learning model's container image is made using Docker, which expedites the deployment procedure.

Link for GitHub:

https://github.com/Sanjeev-Prem/Cloud_Assi2

Link for Docker:

https://hub.docker.com/repository/docker/sg2656/predequlity

Steps for the Execution for Wine Quality Prediction AWS Spark Application:

1.Create a Key-pair for the EMR Cluster :go to EC2/Network/Key-pairs

Use the format of .pem and download the keypair

Created key pair as: Prem.pem

2.Create an S3 bucket

Created an S3 bucket in aws: Buicketwine

3.Then go to EMR console and create EMR cluster

4. Creating the spark in the AWS instance by using EMR console:

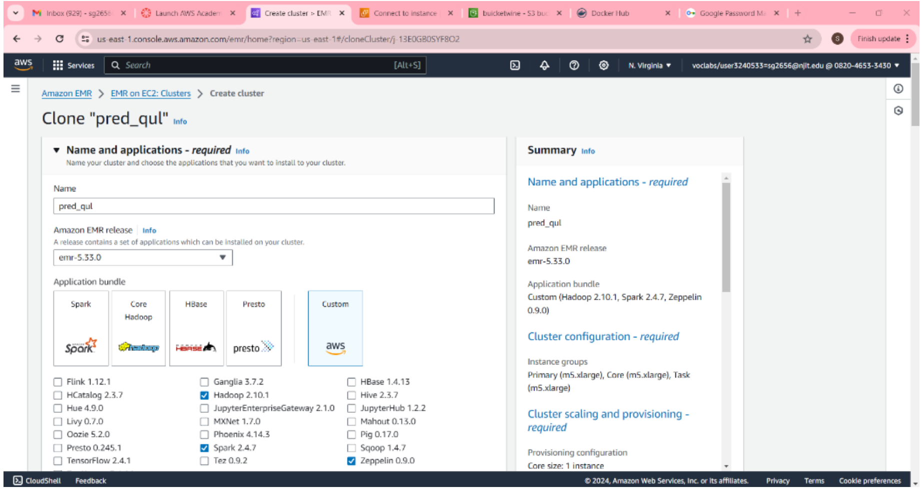
Creating the spark cluster by using the EMR console, and create the 4 instances:

Name and application:

Name:pa2winepqmahi

Amazon EMR release: emr-5.33.0

Application bundle: Hadoop 2.10.1, Spark 2.4.7, Zippeline 0.9.0, and Yarn



Amazon EMR > EMR on EC2: Clusters > Create cluster

Clone "pred_qul" Info

▼ Name and applications - *required* Info

Name your cluster and choose the applications that you want to install to your cluster.

Name

pred_qul

Amazon EMR release | Info

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

emr-5.33.0

Application bundle

Spark | Core Hadoop | HBase | Presto | Custom

☐ Flink 1.12.1
☐ HCatalog 2.3.7
☐ Hue 4.9.0
☐ Livy 0.7.0
☐ Oozie 5.2.0
☐ Presto 0.245.1
☐ TensorFlow 2.4.1
☐ Ganglia 3.7.2
☑ Hadoop 2.10.1
☐ JupyterEnterpriseGateway 2.1.0
☐ MXNet 1.7.0
☐ Phoenix 4.14.3
☑ Spark 2.4.7
☐ Tez 0.9.2
☐ HBase 1.4.13
☐ Hive 2.3.7
☐ JupyterHub 1.2.2
☐ Mahout 0.13.0
☐ Pig 0.17.0
☐ Sqoop 1.4.7
☑ Zeppelin 0.9.0

Summary Info

Name and applications - *required*

Name
pred_qul

Amazon EMR release
emr-5.33.0

Application bundle
Custom (Hadoop 2.10.1, Spark 2.4.7, Zeppelin 0.9.0)

Cluster configuration - *required*

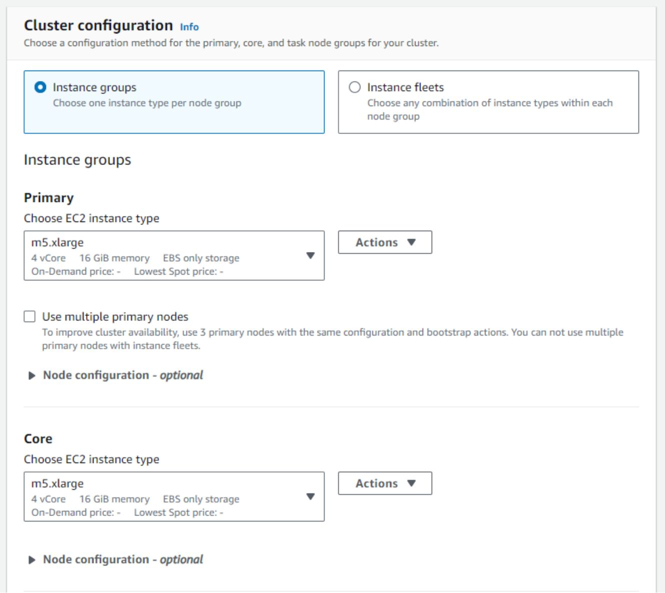
Instance groups
Primary (m5.xlarge), Core (m5.xlarge), Task (m5.xlarge)

Cluster scaling and provisioning - *required*

Provisioning configuration
Core size: 1 instance

Note: Here it says Clone "pred_qul" as I have cloned the previous configuration instead of creating from scratch to save time.

Cluster Configuration:



Cluster configuration Info

Choose a configuration method for the primary, core, and task node groups for your cluster.

● Instance groups
Choose one instance type per node group

○ Instance fleets
Choose any combination of instance types within each node group

Instance groups

Primary

Choose EC2 instance type

m5.xlarge
4 vCore 16 GiB memory EBS only storage
On-Demand price: - Lowest Spot price: -

☐ 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*

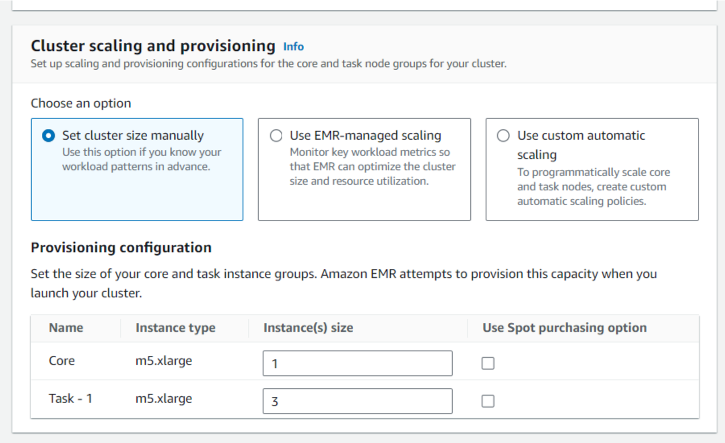
Core

Choose EC2 instance type

m5.xlarge
4 vCore 16 GiB memory EBS only storage
On-Demand price: - Lowest Spot price: -

▶ Node configuration - *optional*

Cluster Scaling and provisioning:



Cluster scaling and provisioning

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.

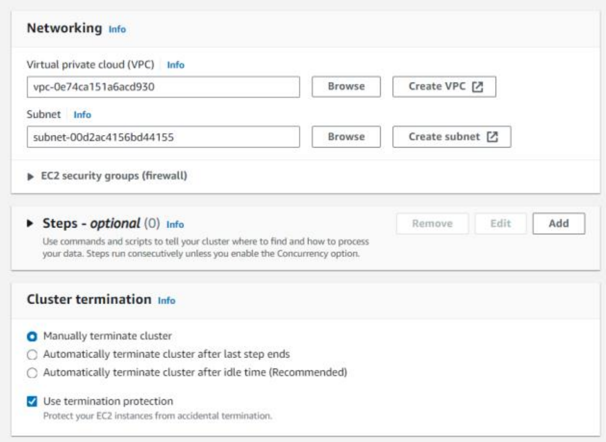
○ Use custom automatic scaling
To programmatically scale core and task nodes, create custom automatic scaling policies.

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	m5.xlarge	1	□
Task - 1	m5.xlarge	3	□

Networking & Cluster Termination:



Networking

Virtual private cloud (VPC)

vpc-0e74ca151a6acd930

Subnet

subnet-00d2ac4156bd44155

▶ EC2 security groups (firewall)

▶ Steps - *optional* (0)

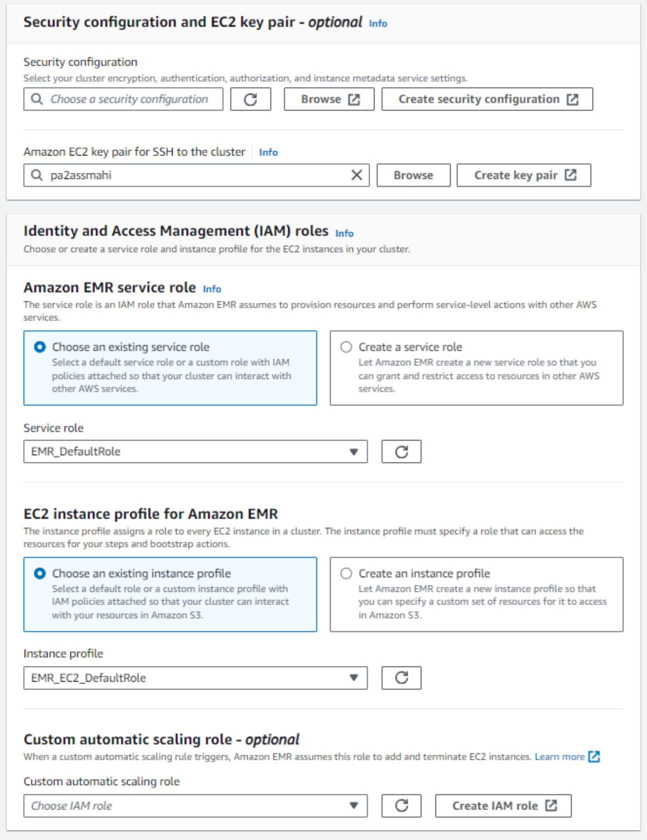
Use commands and scripts to tell your cluster where to find and how to process your data. Steps run consecutively unless you enable the Concurrency option.

Cluster termination

● 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.

Security Configuration and EC2 Key pair & Identity and access management(IAM) roles:



Security configuration and EC2 key pair - *optional* Info

Security configuration
Select your cluster encryption, authentication, authorization, and instance metadata service settings.
Choose a security configuration

Amazon EC2 key pair for SSH to the cluster Info
pa2assmahi

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.

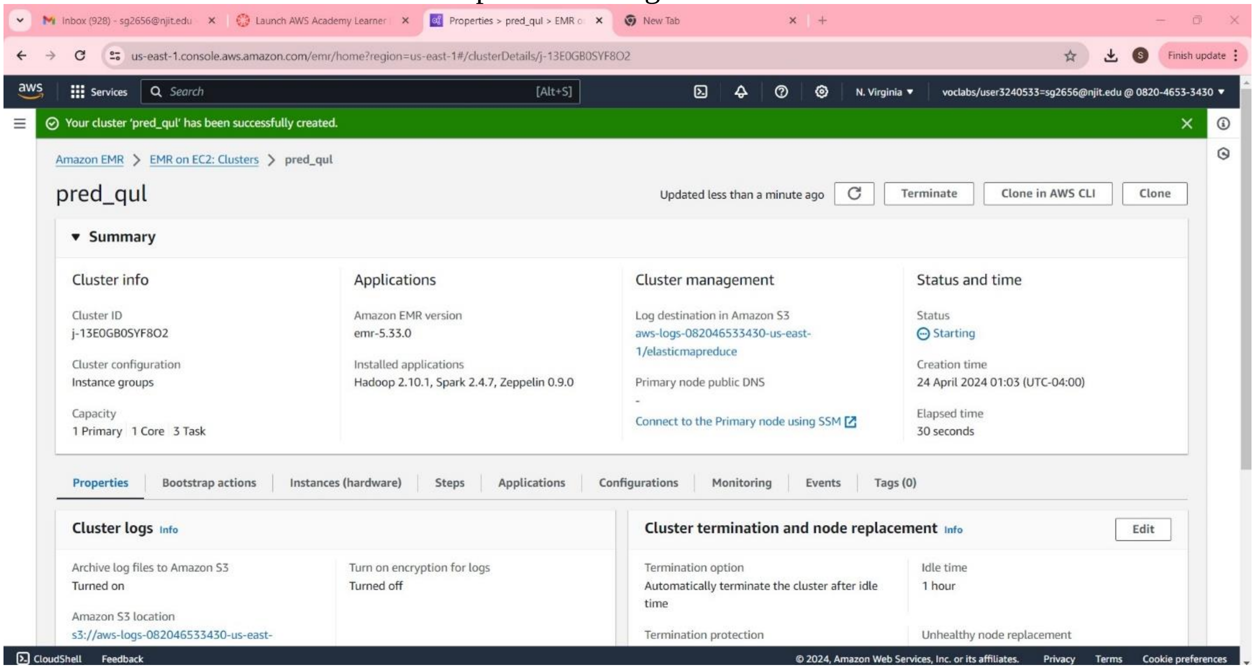
○ Create an instance profile
Let Amazon EMR create a new instance profile so that you can specify a custom set of resources for it to access in Amazon S3.

Instance profile
EMR_EC2_DefaultRole

Custom automatic scaling role - *optional*
When a custom automatic scaling rule triggers, Amazon EMR assumes this role to add and terminate EC2 instances. Learn more

Custom automatic scaling role
Choose IAM role

We can follow above steps for creating the EMR cluster for the instances



us-east-1.console.aws.amazon.com/emr/home?region=us-east-1#/clusterDetails/j-13E0GB0SYF8O2

Your cluster 'pred_qul' has been successfully created.

Amazon EMR > EMR on EC2: Clusters > pred_qul

pred_qul

Updated less than a minute ago

Summary

Cluster info

Cluster ID
j-13E0GB0SYF8O2

Cluster configuration
Instance groups

Capacity
1 Primary | 1 Core | 3 Task

Applications

Amazon EMR version
emr-5.33.0

Installed applications
Hadoop 2.10.1, Spark 2.4.7, Zeppelin 0.9.0

Cluster management

Log destination in Amazon S3
aws-logs-082046533430-us-east-1/elasticmapreduce

Primary node public DNS
-

Connect to the Primary node using SSM

Status and time

Status
Starting

Creation time
24 April 2024 01:03 (UTC-04:00)

Elapsed time
30 seconds

Cluster logs Info

Archive log files to Amazon S3
Turned on

Turn on encryption for logs
Turned off

Amazon S3 location
s3://aws-logs-082046533430-us-east-

Cluster termination and node replacement Info

Termination option
Automatically terminate the cluster after idle time

Idle time
1 hour

Termination protection

Unhealthy node replacement

5. We are now parallelly training an ML model on an EC2 instance in a spark cluster without Docker:

1.Now the cluster will accept the tasks to run the ML model

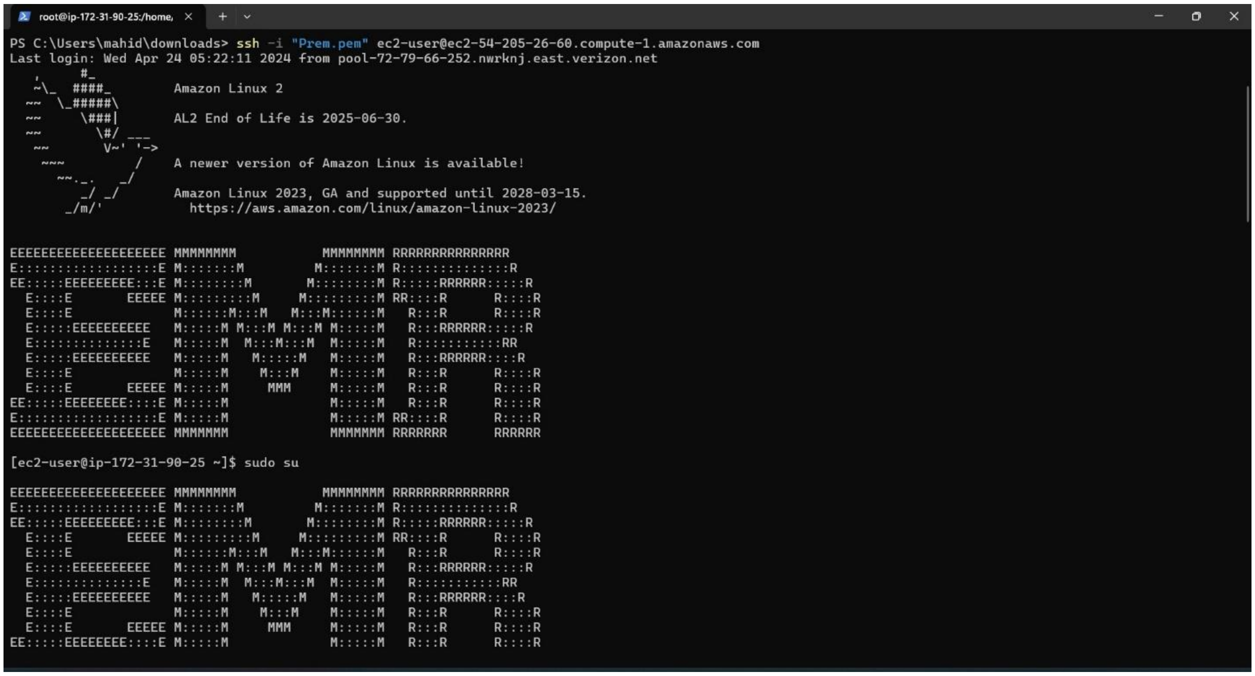
Need to connect the Master instance in the Terminal:

ssh -i "Prem.pem" ec2-user@ec2-54-205-26-60.compute-1.amazonaws.com

and it is successfully login.

2.After the login of Master instance then change the root by using

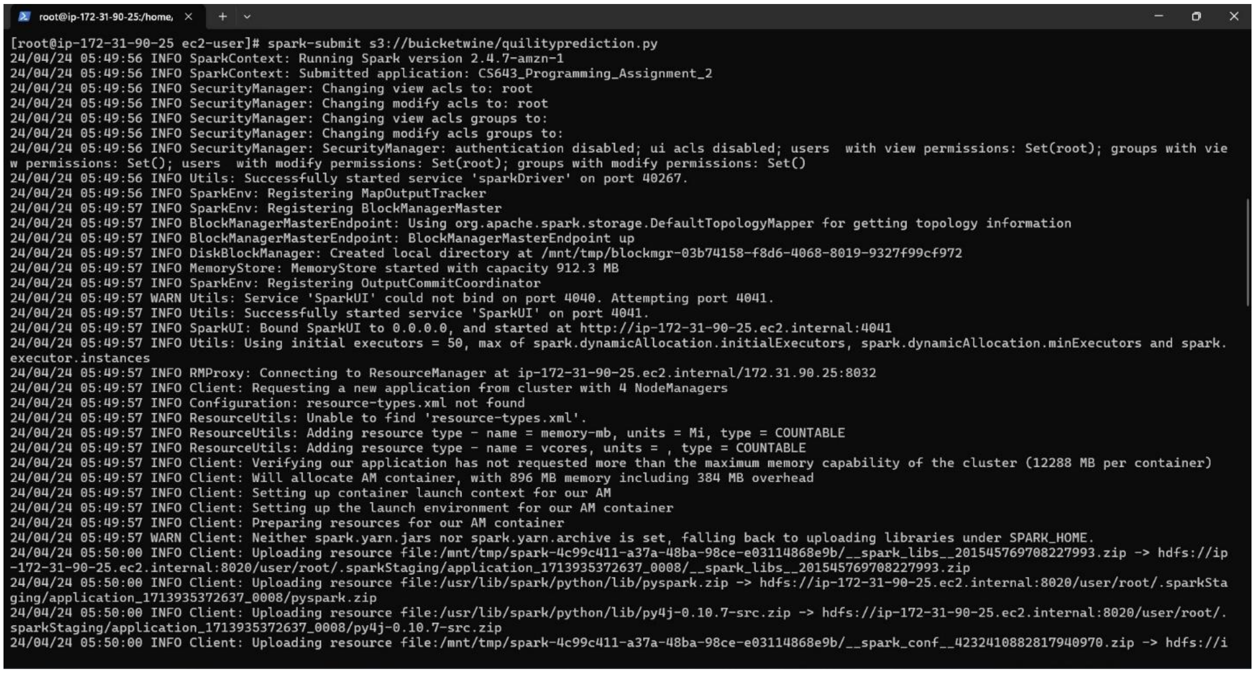
Sudo su



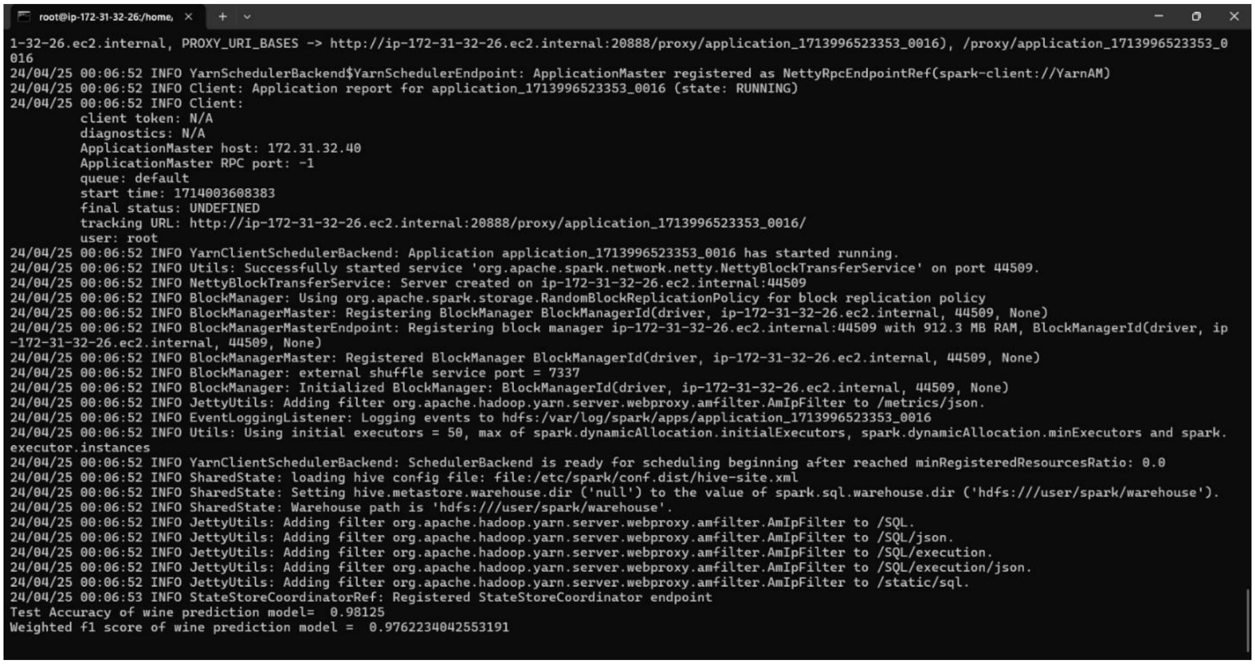
```
PS C:\Users\mahid\downloads> ssh -i "Prem.pem" ec2-user@ec2-54-205-26-60.compute-1.amazonaws.com
Last login: Wed Apr 24 05:22:11 2024 from pool-72-79-66-252.nwrknj.east.verizon.net
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2025-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/


EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::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 M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  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     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[ec2-user@ip-172-31-90-25 ~]$ sudo su

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::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 M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  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     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
```

3. Submit the task by the command:

spark-submit s3://buicketwine/ quilityprediction.py

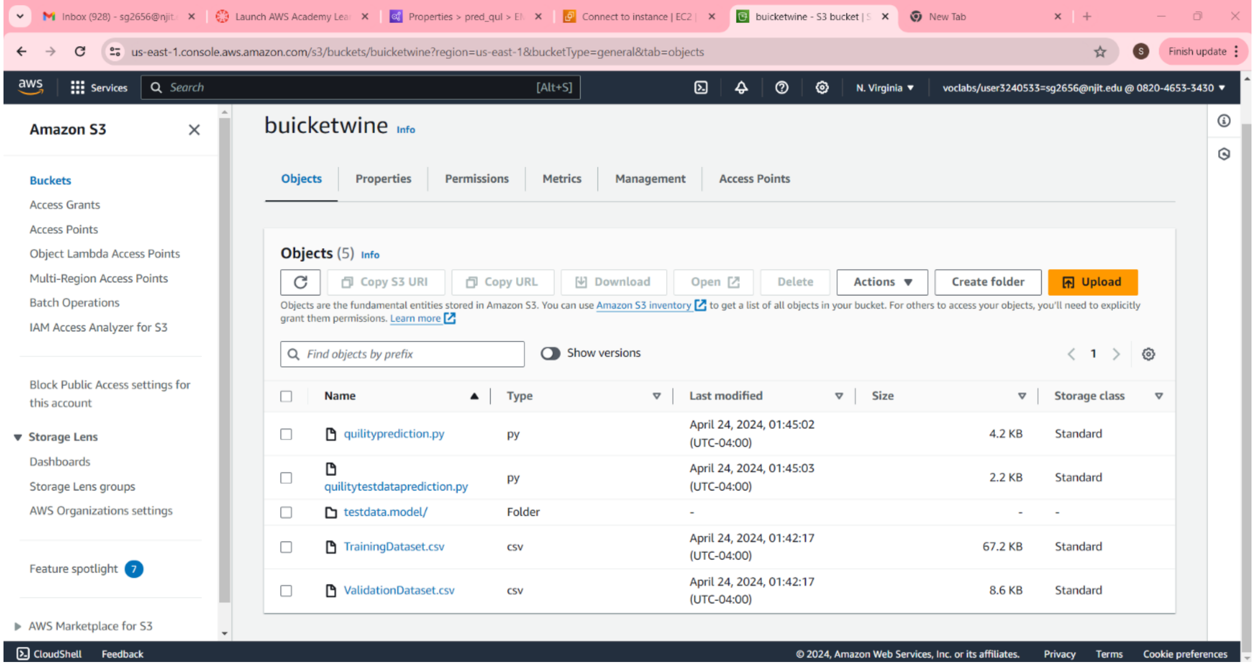


```
[root@ip-172-31-90-25 ec2-user]# spark-submit s3://buicketwine/quilityprediction.py
24/04/24 05:49:56 INFO SparkContext: Running Spark version 2.4.7-amzn-1
24/04/24 05:49:56 INFO SparkContext: Submitted application: CS643_Programming_Assignment_2
24/04/24 05:49:56 INFO SecurityManager: Changing view acls to: root
24/04/24 05:49:56 INFO SecurityManager: Changing modify acls to: root
24/04/24 05:49:56 INFO SecurityManager: Changing view acls groups to:
24/04/24 05:49:56 INFO SecurityManager: Changing modify acls groups to:
24/04/24 05:49:56 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users  with view permissions: Set(root); groups with vie
w permissions: Set(); users  with modify permissions: Set(root); groups with modify permissions: Set()
24/04/24 05:49:56 INFO Utils: Successfully started service 'sparkDriver' on port 40267.
24/04/24 05:49:56 INFO SparkEnv: Registering MapOutputTracker
24/04/24 05:49:57 INFO SparkEnv: Registering BlockManagerMaster
24/04/24 05:49:57 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper for getting topology information
24/04/24 05:49:57 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
24/04/24 05:49:57 INFO DiskBlockManager: Created local directory at /mnt/tmp/blockmgr-03b74158-f8d6-4068-8019-9327f99cf972
24/04/24 05:49:57 INFO MemoryStore: MemoryStore started with capacity 912.3 MB
24/04/24 05:49:57 INFO SparkEnv: Registering OutputCommitCoordinator
24/04/24 05:49:57 WARN Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
24/04/24 05:49:57 INFO Utils: Successfully started service 'SparkUI' on port 4041.
24/04/24 05:49:57 INFO SparkUI: Bound SparkUI to 0.0.0.0, and started at http://ip-172-31-90-25.ec2.internal:4041
24/04/24 05:49:57 INFO Utils: Using initial executors = 50, max of spark.dynamicAllocation.initialExecutors, spark.dynamicAllocation.minExecutors and spark.
executor.instances
24/04/24 05:49:57 INFO RMProxy: Connecting to ResourceManager at ip-172-31-90-25.ec2.internal/172.31.90.25:8032
24/04/24 05:49:57 INFO Client: Requesting a new application from cluster with 4 NodeManagers
24/04/24 05:49:57 INFO Configuration: resource-types.xml not found
24/04/24 05:49:57 INFO ResourceUtils: Unable to find 'resource-types.xml'.
24/04/24 05:49:57 INFO ResourceUtils: Adding resource type - name = memory-mb, units = Mi, type = COUNTABLE
24/04/24 05:49:57 INFO ResourceUtils: Adding resource type - name = vcores, units = , type = COUNTABLE
24/04/24 05:49:57 INFO Client: Verifying our application has not requested more than the maximum memory capability of the cluster (12288 MB per container)
24/04/24 05:49:57 INFO Client: Will allocate AM container, with 896 MB memory including 384 MB overhead
24/04/24 05:49:57 INFO Client: Setting up container launch context for our AM
24/04/24 05:49:57 INFO Client: Setting up the launch environment for our AM container
24/04/24 05:49:57 INFO Client: Preparing resources for our AM container
24/04/24 05:49:57 WARN Client: Neither spark.yarn.jars nor spark.yarn.archive is set, falling back to uploading libraries under SPARK_HOME.
24/04/24 05:50:00 INFO Client: Uploading resource file:/mnt/tmp/spark-4c99c411-a37a-48ba-98ce-e03114868e9b/__spark_libs__201545769708227993.zip -> hdfs://ip
-172-31-90-25.ec2.internal:8020/user/root/.sparkStaging/application_1713935372637_0008/__spark_libs__201545769708227993.zip
24/04/24 05:50:00 INFO Client: Uploading resource file:/usr/lib/spark/python/lib/pyspark.zip -> hdfs://ip-172-31-90-25.ec2.internal:8020/user/root/.sparkSta
ging/application_1713935372637_0008/pyspark.zip
24/04/24 05:50:00 INFO Client: Uploading resource file:/usr/lib/spark/python/lib/py4j-0.10.7-src.zip -> hdfs://ip-172-31-90-25.ec2.internal:8020/user/root/.
sparkStaging/application_1713935372637_0008/py4j-0.10.7-src.zip
24/04/24 05:50:00 INFO Client: Uploading resource file:/mnt/tmp/spark-4c99c411-a37a-48ba-98ce-e03114868e9b/__spark_conf__4232410882817940970.zip -> hdfs://i
```



```
1-32-26.ec2.internal, PROXY_URI_BASES -> http://ip-172-31-32-26.ec2.internal:20888/proxy/application_1713996523353_0016), /proxy/application_1713996523353_0
016
24/04/25 00:06:52 INFO YarnSchedulerBackend$YarnSchedulerEndpoint: ApplicationMaster registered as NettyRpcEndpointRef(spark-client://YarnAM)
24/04/25 00:06:52 INFO Client: Application report for application_1713996523353_0016 (state: RUNNING)
24/04/25 00:06:52 INFO Client:
         client token: N/A
         diagnostics: N/A
         ApplicationMaster host: 172.31.32.40
         ApplicationMaster RPC port: -1
         queue: default
         start time: 1714003608383
         final status: UNDEFINED
         tracking URL: http://ip-172-31-32-26.ec2.internal:20888/proxy/application_1713996523353_0016/
         user: root
24/04/25 00:06:52 INFO YarnClientSchedulerBackend: Application application_1713996523353_0016 has started running.
24/04/25 00:06:52 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 44509.
24/04/25 00:06:52 INFO NettyBlockTransferService: Server created on ip-172-31-32-26.ec2.internal:44509
24/04/25 00:06:52 INFO BlockManager: Using org.apache.spark.storage.RandomBlockReplicationPolicy for block replication policy
24/04/25 00:06:52 INFO BlockManagerMaster: Registering BlockManager BlockManagerId(driver, ip-172-31-32-26.ec2.internal, 44509, None)
24/04/25 00:06:52 INFO BlockManagerMasterEndpoint: Registering block manager ip-172-31-32-26.ec2.internal:44509 with 912.3 MB RAM, BlockManagerId(driver, ip
-172-31-32-26.ec2.internal, 44509, None)
24/04/25 00:06:52 INFO BlockManagerMaster: Registered BlockManager BlockManagerId(driver, ip-172-31-32-26.ec2.internal, 44509, None)
24/04/25 00:06:52 INFO BlockManager: external shuffle service port = 7337
24/04/25 00:06:52 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver, ip-172-31-32-26.ec2.internal, 44509, None)
24/04/25 00:06:52 INFO JettyUtils: Adding filter org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter to /metrics/json.
24/04/25 00:06:52 INFO EventLoggingListener: Logging events to hdfs:/var/log/spark/apps/application_1713996523353_0016
24/04/25 00:06:52 INFO Utils: Using initial executors = 50, max of spark.dynamicAllocation.initialExecutors, spark.dynamicAllocation.minExecutors and spark.
executor.instances
24/04/25 00:06:52 INFO YarnClientSchedulerBackend: SchedulerBackend is ready for scheduling beginning after reached minRegisteredResourcesRatio: 0.0
24/04/25 00:06:52 INFO SharedState: loading hive config file: file:/etc/spark/conf.dist/hive-site.xml
24/04/25 00:06:52 INFO SharedState: Setting hive.metastore.warehouse.dir ('null') to the value of spark.sql.warehouse.dir ('hdfs:///user/spark/warehouse').
24/04/25 00:06:52 INFO SharedState: Warehouse path is 'hdfs:///user/spark/warehouse'.
24/04/25 00:06:52 INFO JettyUtils: Adding filter org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter to /SQL.
24/04/25 00:06:52 INFO JettyUtils: Adding filter org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter to /SQL/json.
24/04/25 00:06:52 INFO JettyUtils: Adding filter org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter to /SQL/execution.
24/04/25 00:06:52 INFO JettyUtils: Adding filter org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter to /SQL/execution/json.
24/04/25 00:06:52 INFO JettyUtils: Adding filter org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter to /static/sql.
24/04/25 00:06:53 INFO StateStoreCoordinatorRef: Registered StateStoreCoordinator endpoint
Test Accuracy of wine prediction model=  0.98125
Weighted f1 score of wine prediction model =  0.9762234042553191
```

4.Then you can find the trace status for the above tasks, The status is succeed then there is a creation of test.model in the s3 bucket s3://buicketwine


Inbox (928) - sg2656@njit
Launch AWS Academy Lea
Properties > pred_qul > El
Connect to instance | EC2 |
buicketwine - S3 bucket | S
New Tab
us-east-1.console.aws.amazon.com/s3/buckets/buicketwine?region=us-east-1&bucketType=general&tab=objects
Finish update
Services
Search
[Alt+S]
N. Virginia
voclabs/user3240533=sg2656@njit.edu @ 0820-4653-3430
Amazon S3
Buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 7
AWS Marketplace for S3
buicketwine Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Objects (5) Info
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Find objects by prefix
Show versions
1
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

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

☐	Name	Type	Last modified	Size	Storage class
☐	quilityprediction.py	py	April 24, 2024, 01:45:02 (UTC-04:00)	4.2 KB	Standard
☐	quilitytestdataprediction.py	py	April 24, 2024, 01:45:03 (UTC-04:00)	2.2 KB	Standard
☐	testdata.model/	Folder	-	-	-
☐	TrainingDataset.csv	csv	April 24, 2024, 01:42:17 (UTC-04:00)	67.2 KB	Standard
☐	ValidationDataset.csv	csv	April 24, 2024, 01:42:17 (UTC-04:00)	8.6 KB	Standard

6.Now we are running ML model using the Docker:

1.Create an docker account and sign up.

2. After the successful login then download and setup the docker in your local system

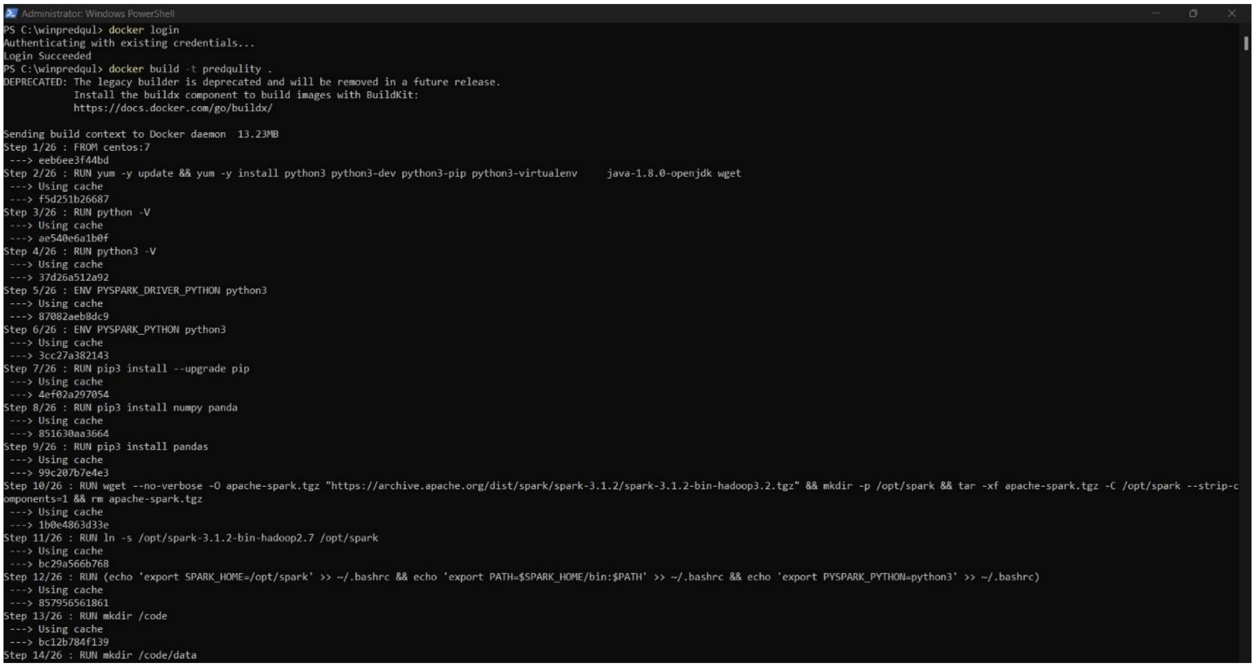
3.Install the docker

4. Login the docker in the power shell by the command

```
docker login
Pwd
```

5.After login you need to build the image:

```
docker build -t predquilty .
```



Administrator: Windows PowerShell

```
PS C:\winpredqul> docker login
Authenticating with existing credentials...
Login Succeeded
PS C:\winpredqul> docker build -t predqulity .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  13.23MB
Step 1/26 : FROM centos:7
 ---> eeb6ee3f44bd
Step 2/26 : RUN yum -y update && yum -y install python3 python3-dev python3-pip python3-virtualenv     java-1.8.0-openjdk wget
 ---> Using cache
 ---> f5d251b26687
Step 3/26 : RUN python -V
 ---> Using cache
 ---> ae540e6a1b0f
Step 4/26 : RUN python3 -V
 ---> Using cache
 ---> 37d26a512a92
Step 5/26 : ENV PYSPARK_DRIVER_PYTHON python3
 ---> Using cache
 ---> 87082aeb8dc9
Step 6/26 : ENV PYSPARK_PYTHON python3
 ---> Using cache
 ---> 3cc27a382143
Step 7/26 : RUN pip3 install --upgrade pip
 ---> Using cache
 ---> 4ef02a297054
Step 8/26 : RUN pip3 install numpy panda
 ---> Using cache
 ---> 851630aa3664
Step 9/26 : RUN pip3 install pandas
 ---> Using cache
 ---> 99c207b7e4e3
Step 10/26 : RUN wget --no-verbose -O apache-spark.tgz "https://archive.apache.org/dist/spark/spark-3.1.2/spark-3.1.2-bin-hadoop3.2.tgz" && mkdir -p /opt/spark && tar -xf apache-spark.tgz -C /opt/spark --strip-c
omponents=1 && rm apache-spark.tgz
 ---> Using cache
 ---> 1b0e4863d33e
Step 11/26 : RUN ln -s /opt/spark-3.1.2-bin-hadoop2.7 /opt/spark
 ---> Using cache
 ---> bc29a566b768
Step 12/26 : RUN (echo 'export SPARK_HOME=/opt/spark' >> ~/.bashrc && echo 'export PATH=$SPARK_HOME/bin:$PATH' >> ~/.bashrc && echo 'export PYSPARK_PYTHON=python3' >> ~/.bashrc)
 ---> Using cache
 ---> 857956561861
Step 13/26 : RUN mkdir /code
 ---> Using cache
 ---> bc12b784f139
Step 14/26 : RUN mkdir /code/data
```

6. The push and pull into the docker hub repository:

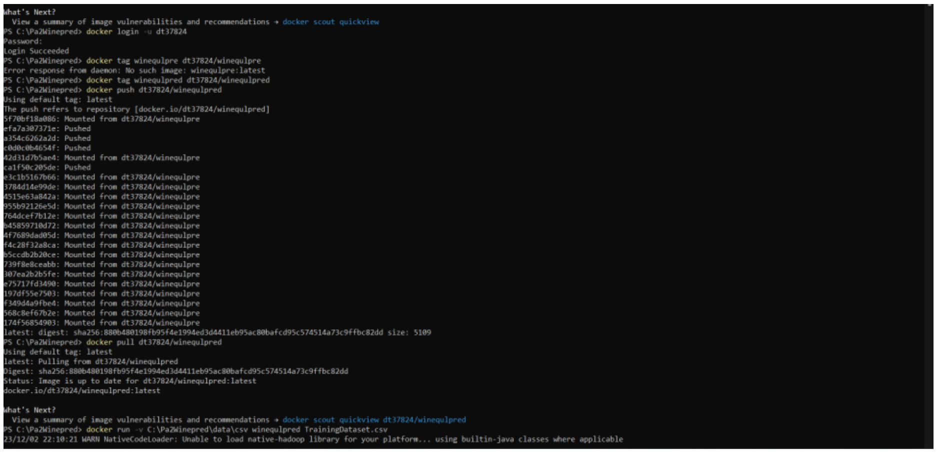
PUSH:

docker tag predquility sg2656/ predquility

docker push sg2656/predquility

PULL:

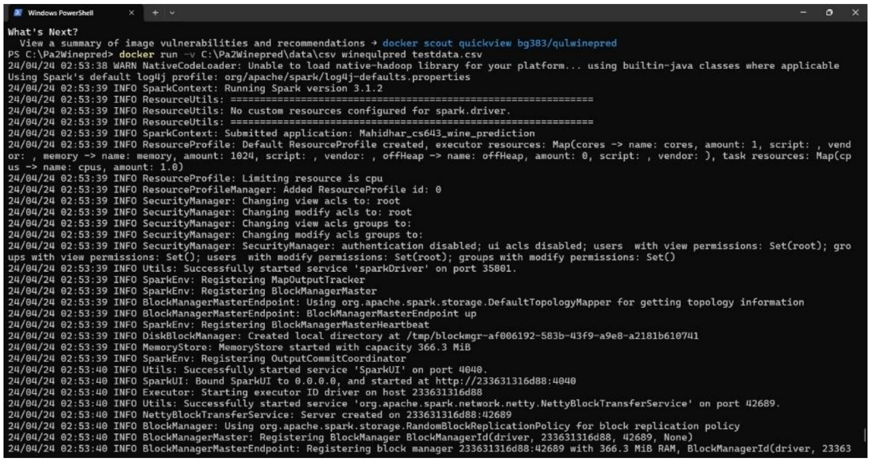
docker pull sg2656/predquility



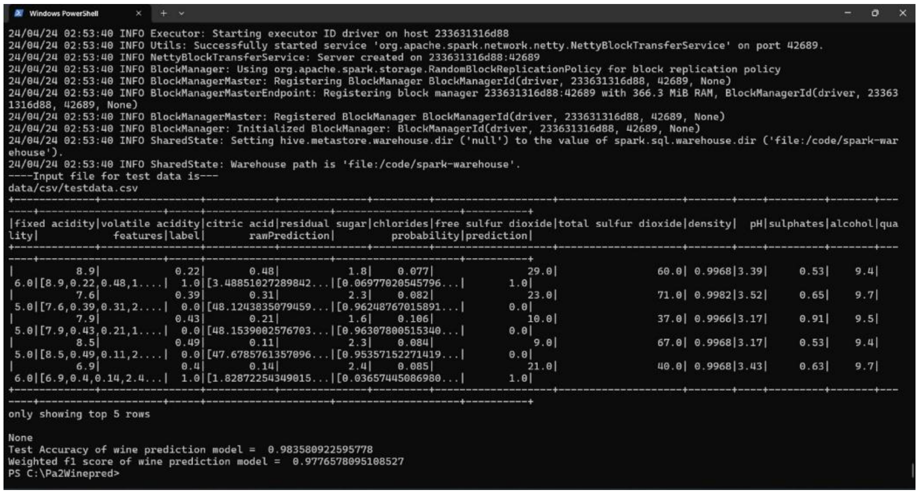
```
What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview
PS C:\Pa2Winepred> docker login -u dt37824
Password:
Login Succeeded
PS C:\Pa2Winepred> docker tag winequlpre dt37824/winequlpre
Error response from daemon: No such image: winequlpre:latest
PS C:\Pa2Winepred> docker tag winequlpred dt37824/winequlpred
PS C:\Pa2Winepred> docker push dt37824/winequlpred
Using default tag: latest
The push refers to repository [docker.io/dt37824/winequlpred]
5f70bf18a086: Mounted from dt37824/winequlpre
efa7a307371e: Pushed
a354c6262a2d: Pushed
c0d0c0b4654f: Pushed
42d31d7b5ae4: Mounted from dt37824/winequlpre
ca1f50c205de: Pushed
e3c1b5167b66: Mounted from dt37824/winequlpre
3784d14e99de: Mounted from dt37824/winequlpre
4515e63a842a: Mounted from dt37824/winequlpre
955b92126e5d: Mounted from dt37824/winequlpre
764dcef7b12e: Mounted from dt37824/winequlpre
b45859710d72: Mounted from dt37824/winequlpre
4f7689dad05d: Mounted from dt37824/winequlpre
f4c28f32a8ca: Mounted from dt37824/winequlpre
b5ccdb2b20ce: Mounted from dt37824/winequlpre
739f8e8ceabb: Mounted from dt37824/winequlpre
307ea2b2b5fe: Mounted from dt37824/winequlpre
e75717fd3490: Mounted from dt37824/winequlpre
197df55e7503: Mounted from dt37824/winequlpre
f349d4a9fbe4: Mounted from dt37824/winequlpre
568c8ef67b2e: Mounted from dt37824/winequlpre
174f56854903: Mounted from dt37824/winequlpre
latest: digest: sha256:880b480198fb95f4e1994ed3d4411eb95ac80bafcd95c574514a73c9ffbc82dd size: 5109
PS C:\Pa2Winepred> docker pull dt37824/winequlpred
Using default tag: latest
latest: Pulling from dt37824/winequlpred
Digest: sha256:880b480198fb95f4e1994ed3d4411eb95ac80bafcd95c574514a73c9ffbc82dd
Status: Image is up to date for dt37824/winequlpred:latest
docker.io/dt37824/winequlpred:latest

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview dt37824/winequlpred
PS C:\Pa2Winepred> docker run -v C:\Pa2Winepred\data\csv winequlpred TrainingDataset.csv
23/12/02 22:10:21 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
```

7. Now Your test data file should be kept in a special folder called "dir." Use the Docker container to mount this directory, then run the container by issuing the following command..

docker run -v C:\Pa2 predquility\data\csv Predquility testdata.csv



```
What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview bg383/qulwinepred
PS C:\Pa2Winepred> docker run -v C:\Pa2Winepred\data\csv winequlpred testdata.csv
24/04/24 02:53:38 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
24/04/24 02:53:39 INFO SparkContext: Running Spark version 3.1.2
24/04/24 02:53:39 INFO ResourceUtils: ==============================================================
24/04/24 02:53:39 INFO ResourceUtils: No custom resources configured for spark.driver.
24/04/24 02:53:39 INFO ResourceUtils: ==============================================================
24/04/24 02:53:39 INFO SparkContext: Submitted application: Mahidhar_cs643_wine_prediction
24/04/24 02:53:39 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(cores -> name: cores, amount: 1, script: , vend
or: , memory -> name: memory, amount: 1024, script: , vendor: , offHeap -> name: offHeap, amount: 0, script: , vendor: ), task resources: Map(cp
us -> name: cpus, amount: 1.0)
24/04/24 02:53:39 INFO ResourceProfile: Limiting resource is cpu
24/04/24 02:53:39 INFO ResourceProfileManager: Added ResourceProfile id: 0
24/04/24 02:53:39 INFO SecurityManager: Changing view acls to: root
24/04/24 02:53:39 INFO SecurityManager: Changing modify acls to: root
24/04/24 02:53:39 INFO SecurityManager: Changing view acls groups to:
24/04/24 02:53:39 INFO SecurityManager: Changing modify acls groups to:
24/04/24 02:53:39 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users  with view permissions: Set(root); gro
ups with view permissions: Set(); users  with modify permissions: Set(root); groups with modify permissions: Set()
24/04/24 02:53:39 INFO Utils: Successfully started service 'sparkDriver' on port 35801.
24/04/24 02:53:39 INFO SparkEnv: Registering MapOutputTracker
24/04/24 02:53:39 INFO SparkEnv: Registering BlockManagerMaster
24/04/24 02:53:39 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper for getting topology information
24/04/24 02:53:39 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
24/04/24 02:53:39 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
24/04/24 02:53:39 INFO DiskBlockManager: Created local directory at /tmp/blockmgr-af006192-583b-43f9-a9e8-a2181b610741
24/04/24 02:53:39 INFO MemoryStore: MemoryStore started with capacity 366.3 MiB
24/04/24 02:53:39 INFO SparkEnv: Registering OutputCommitCoordinator
24/04/24 02:53:40 INFO Utils: Successfully started service 'SparkUI' on port 4040.
24/04/24 02:53:40 INFO SparkUI: Bound SparkUI to 0.0.0.0, and started at http://233631316d88:4040
24/04/24 02:53:40 INFO Executor: Starting executor ID driver on host 233631316d88
24/04/24 02:53:40 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 42689.
24/04/24 02:53:40 INFO NettyBlockTransferService: Server created on 233631316d88:42689
24/04/24 02:53:40 INFO BlockManager: Using org.apache.spark.storage.RandomBlockReplicationPolicy for block replication policy
24/04/24 02:53:40 INFO BlockManagerMaster: Registering BlockManager BlockManagerId(driver, 233631316d88, 42689, None)
24/04/24 02:53:40 INFO BlockManagerMasterEndpoint: Registering block manager 233631316d88:42689 with 366.3 MiB RAM, BlockManagerId(driver, 23363
```



```
24/04/24 02:53:40 INFO Executor: Starting executor ID driver on host 233631316d88
24/04/24 02:53:40 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 42689.
24/04/24 02:53:40 INFO NettyBlockTransferService: Server created on 233631316d88:42689
24/04/24 02:53:40 INFO BlockManager: Using org.apache.spark.storage.RandomBlockReplicationPolicy for block replication policy
24/04/24 02:53:40 INFO BlockManagerMaster: Registering BlockManager BlockManagerId(driver, 233631316d88, 42689, None)
24/04/24 02:53:40 INFO BlockManagerMasterEndpoint: Registering block manager 233631316d88:42689 with 366.3 MiB RAM, BlockManagerId(driver, 23363
1316d88, 42689, None)
24/04/24 02:53:40 INFO BlockManagerMaster: Registered BlockManager BlockManagerId(driver, 233631316d88, 42689, None)
24/04/24 02:53:40 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver, 233631316d88, 42689, None)
24/04/24 02:53:40 INFO SharedState: Setting hive.metastore.warehouse.dir ('null') to the value of spark.sql.warehouse.dir ('file:/code/spark-war
ehouse').
24/04/24 02:53:40 INFO SharedState: Warehouse path is 'file:/code/spark-warehouse'.
----Input file for test data is---
data/csv/testdata.csv
+-------------+----------------+-----------+--------------+---------+-------------------+--------------------+-------+----+---------+-------+---
----+--------------------+-----+--------------------+--------------------+----------+
|fixed acidity|volatile acidity|citric acid|residual sugar|chlorides|free sulfur dioxide|total sulfur dioxide|density|  pH|sulphates|alcohol|qua
lity|            features|label|       rawPrediction|         probability|prediction|
+-------------+----------------+-----------+--------------+---------+-------------------+--------------------+-------+----+---------+-------+---
----+--------------------+-----+--------------------+--------------------+----------+
|          8.9|            0.22|       0.48|           1.8|    0.077|               29.0|                60.0| 0.9968|3.39|     0.53|    9.4|
 6.0|[8.9,0.22,0.48,1....|  1.0|[3.48851027289842...|[0.06977020545796...|       1.0|
|          7.6|            0.39|       0.31|           2.3|    0.082|               23.0|                71.0| 0.9982|3.52|     0.65|    9.7|
 5.0|[7.6,0.39,0.31,2....|  0.0|[48.1243835079459...|[0.96248767015891...|       0.0|
|          7.9|            0.43|       0.21|           1.6|    0.106|               10.0|                37.0| 0.9966|3.17|     0.91|    9.5|
 5.0|[7.9,0.43,0.21,1....|  0.0|[48.1539002576703...|[0.96307800515340...|       0.0|
|          8.5|            0.49|       0.11|           2.3|    0.084|                9.0|                67.0| 0.9968|3.17|     0.53|    9.4|
 5.0|[8.5,0.49,0.11,2....|  0.0|[47.6785761357096...|[0.95357152271419...|       0.0|
|          6.9|             0.4|       0.14|           2.4|    0.085|               21.0|                40.0| 0.9968|3.43|     0.63|    9.7|
 6.0|[6.9,0.4,0.14,2.4...|  1.0|[1.82872254349015...|[0.03657445086980...|       1.0|
+-------------+----------------+-----------+--------------+---------+-------------------+--------------------+-------+----+---------+-------+---
----+--------------------+-----+--------------------+--------------------+----------+
only showing top 5 rows

None
Test Accuracy of wine prediction model =  0.983580922595778
Weighted f1 score of wine prediction model =  0.9776578095108527
PS C:\Pa2Winepred>
```

Conclusion: As can be seen in the accompanying graphic, while estimating the wine quality, the accuracy was around 98%.