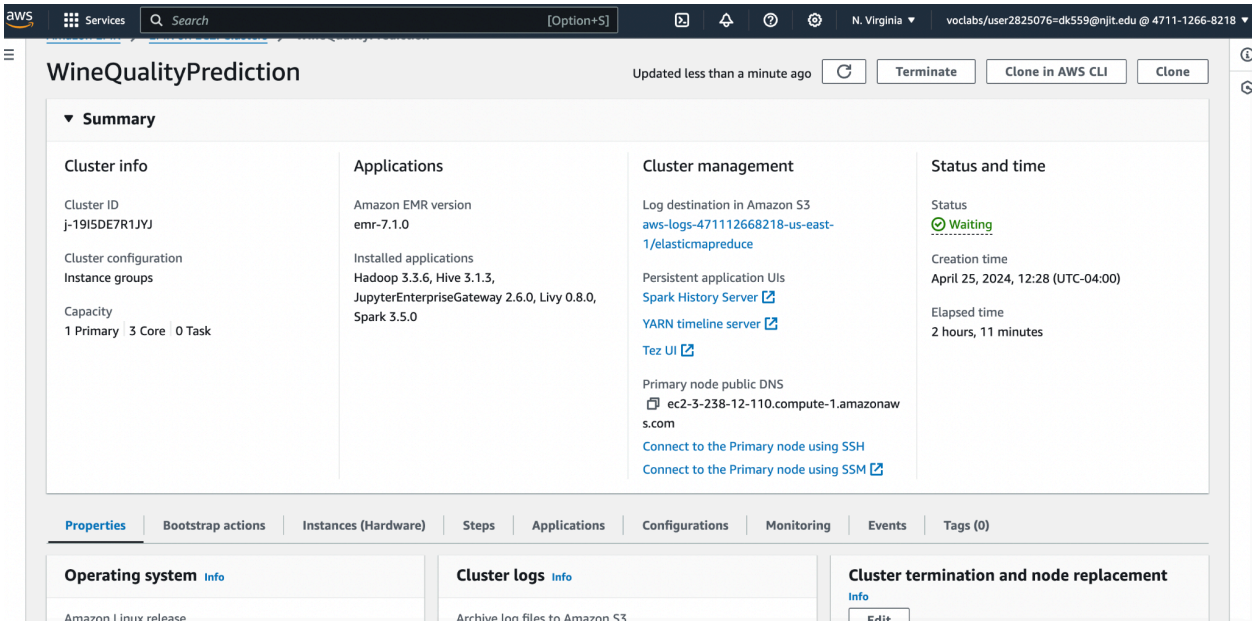
Programming Assignment 2:

In this assignment, EMR cluster with 3 nodes is created.
ec2-3-238-12-110.compute-1.amazonaws.com is the master node and the security group of master node has ssh permissions.

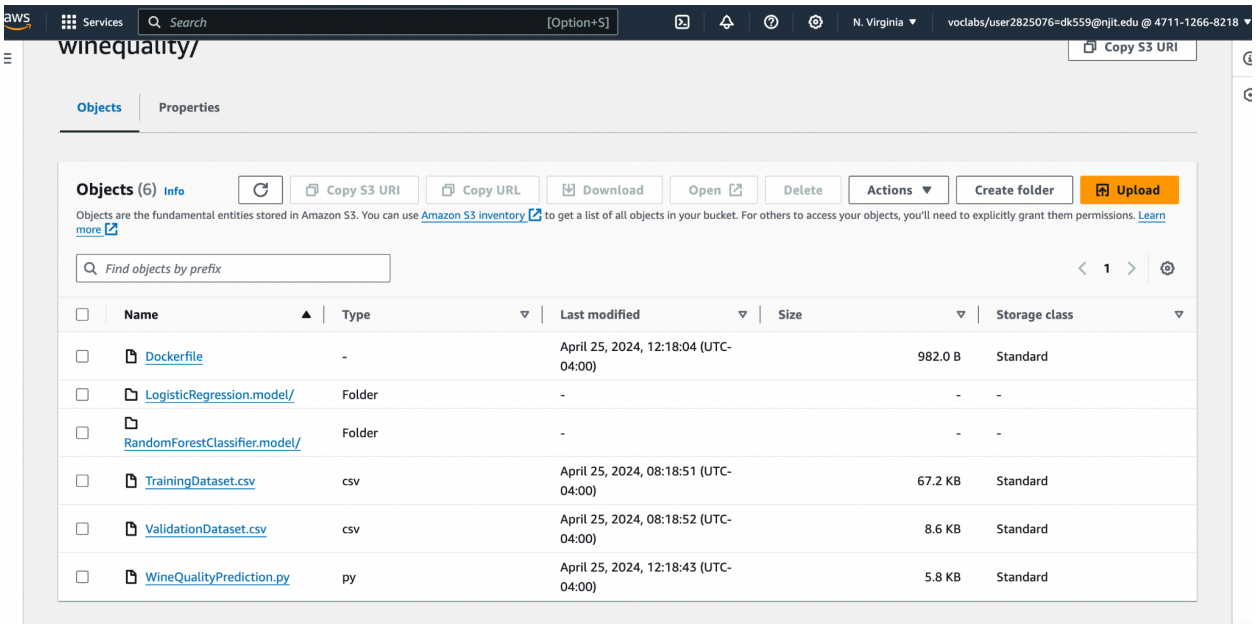


Connect to EC2 Instance - Master node:

1. Open terminal and go to the folder where the key is saved. Change the permission using chmod command.
   chmod 400 prediction.pem

2. Later on, connect the ec2 instance with the following command:
   ssh -i prediction.pem hadoop@ec2-3-238-12-110.compute-1.amazonaws.com
   ec2-3-238-12-110.compute-1.amazonaws.com is the DNS of ec2 instance master node.

Uploaded required files to S3: s3://aws-logs-471112668218-us-east-1/winequality/



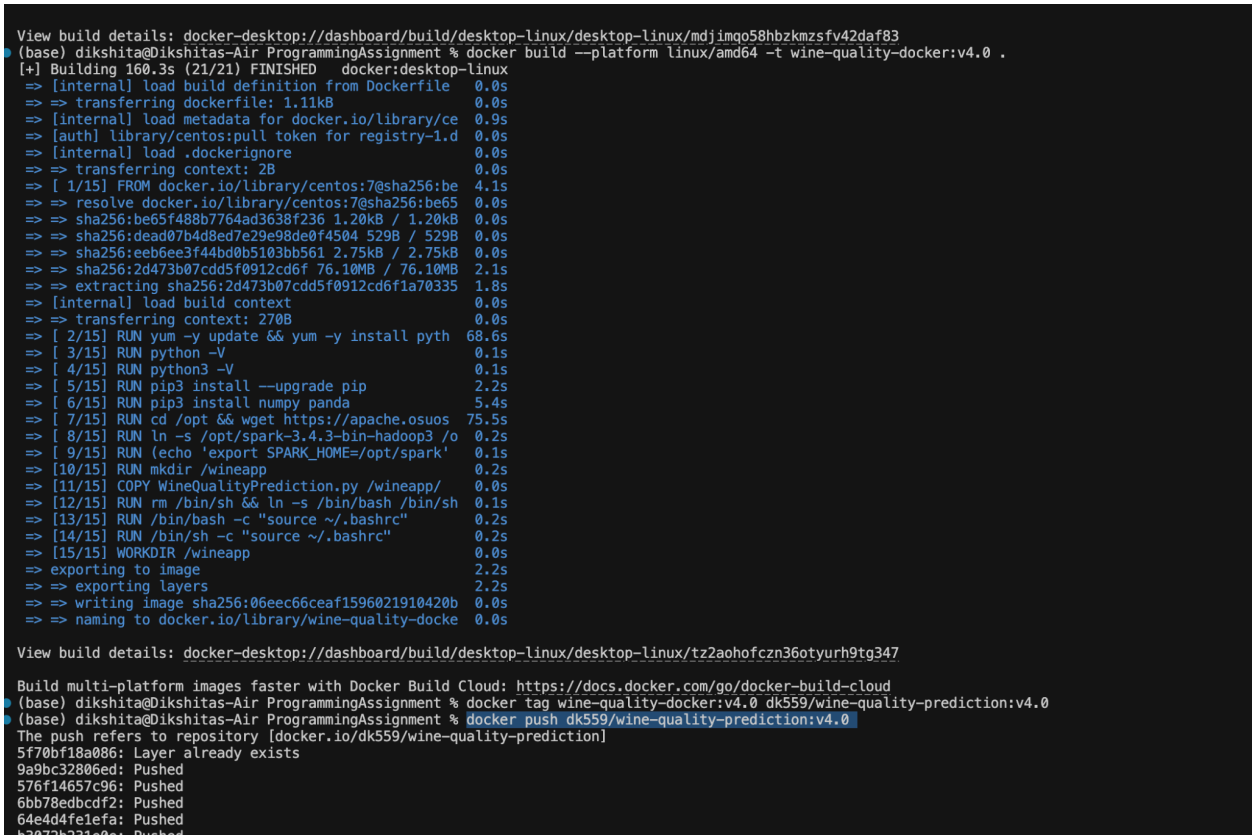
Docker image:

Installed docker on local computer and created docker image and pushed the docker image on
> docker hub with the following command.

> docker build --platform linux/amd64 -t wine-quality-docker:v5.0 .

> docker tag wine-quality-docker:v5.0 dk559/wine-quality-prediction:v5.0

> docker push dk559/wine-quality-prediction:v4.0



Run the Docker image on EC2 instance:

> sudo yum update -y
> sudo yum install docker -y
> sudo service docker start
> sudo usermod -a -G docker ec2-user
> docker run dk559/wine-quality-prediction:v5.0

I learned about AWS EMR, Dockerfile and Apache Spark by watching YouTube videos

I had used ChatGPT incase I faced any errors by pasting the errors.
Example: I was facing issue with the Docker run command on EC2 instance after creating the docker image. ChatGPT suggested me to pass the platform parameter while creating the image.

You

The requested image's platform (linux/arm64/v8) does not match the detected host platform (linux/amd64/v3) and no specific platform was requested

ChatGPT

The error message you're encountering indicates a mismatch between the platform of the Docker image you're trying to run and the platform of the host system where Docker is running.

It seems that the Docker image you're trying to run is built for the `linux/arm64/v8` platform, which is for ARM-based systems, while the host system where Docker is running is `linux/amd64`, which is for x86-based systems.

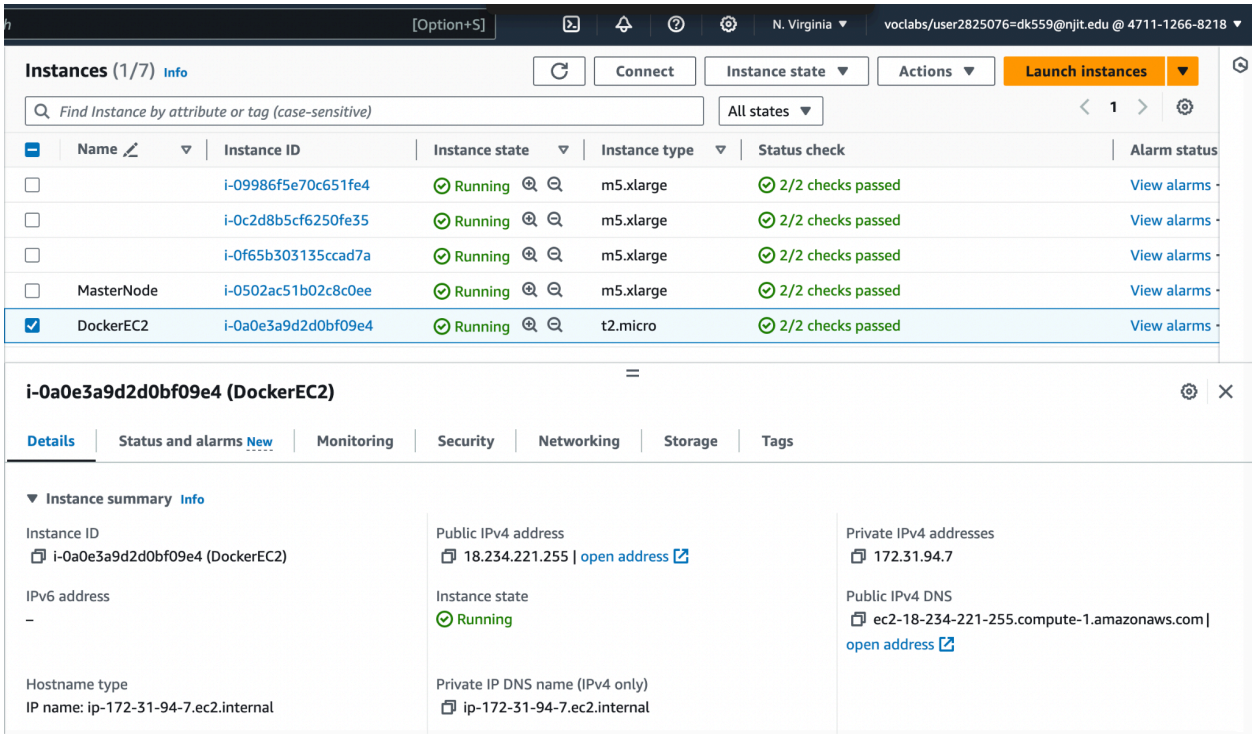
To resolve this issue, you have a few options:

1. **Build Docker Image for the Host Architecture**: If you have access to the Dockerfile used to build the image, you can rebuild the Docker image for the `linux/amd64` platform. You can do this by specifying the `--platform` option when building the image:

bash Copy code

```
docker build --platform linux/amd64 -t your-image-name .
```

2. **Pull Correct Image for Host Architecture**: If you're pulling the image from a registry, ensure


[Option+S]
N. Virginia
voclabs/user2825076=dk559@njit.edu @ 4711-1266-8218
Instances (1/7) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
1
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
i-09986f5e70c651fe4 Running m5.xlarge 2/2 checks passed View alarms
i-0c2d8b5cf6250fe35 Running m5.xlarge 2/2 checks passed View alarms
i-0f65b303135ccad7a Running m5.xlarge 2/2 checks passed View alarms
MasterNode i-0502ac51b02c8c0ee Running m5.xlarge 2/2 checks passed View alarms
DockerEC2 i-0a0e3a9d2d0bf09e4 Running t2.micro 2/2 checks passed View alarms
i-0a0e3a9d2d0bf09e4 (DockerEC2)
Details
Status and alarms New
Monitoring
Security
Networking
Storage
Tags
Instance summary Info
Instance ID
i-0a0e3a9d2d0bf09e4 (DockerEC2)
Public IPv4 address
18.234.221.255 | open address
Private IPv4 addresses
172.31.94.7
IPv6 address
–
Instance state
Running
Public IPv4 DNS
ec2-18-234-221-255.compute-1.amazonaws.com | open address
Hostname type
IP name: ip-172-31-94-7.ec2.internal
Private IP DNS name (IPv4 only)
ip-172-31-94-7.ec2.internal