**CI/CD Deployment for Springboot Application.**

Course : PG FSD Testing in a DevOps Lifecycle part of PG FSD Caltech program

Programmer Details: Remya Pillai

<https://github.com/RemyaPillai99/RemyaPillai99-Phase5Assessment.git>

Git Hub Link: <https://github.com/RemyaPillai99/RemyaPillai99-Phase5Assessment>

## Assignment - 1: Deploying a project on Amazon EC2

The following are the steps that are involved in the assignment -

1. Create EC2 Instance on AWS
2. Connect to EC2 Instance using SSH
3. Cloning the Repo. on EC2
4. Building the App. Image using Docker
5. Running the Image using Docker
6. Access the Application on Browser using public IPv4 address

### Creating EC2 Instance on AWS

1. Launch AWS Console from the LMS Portal

Graphical user interface, text, application

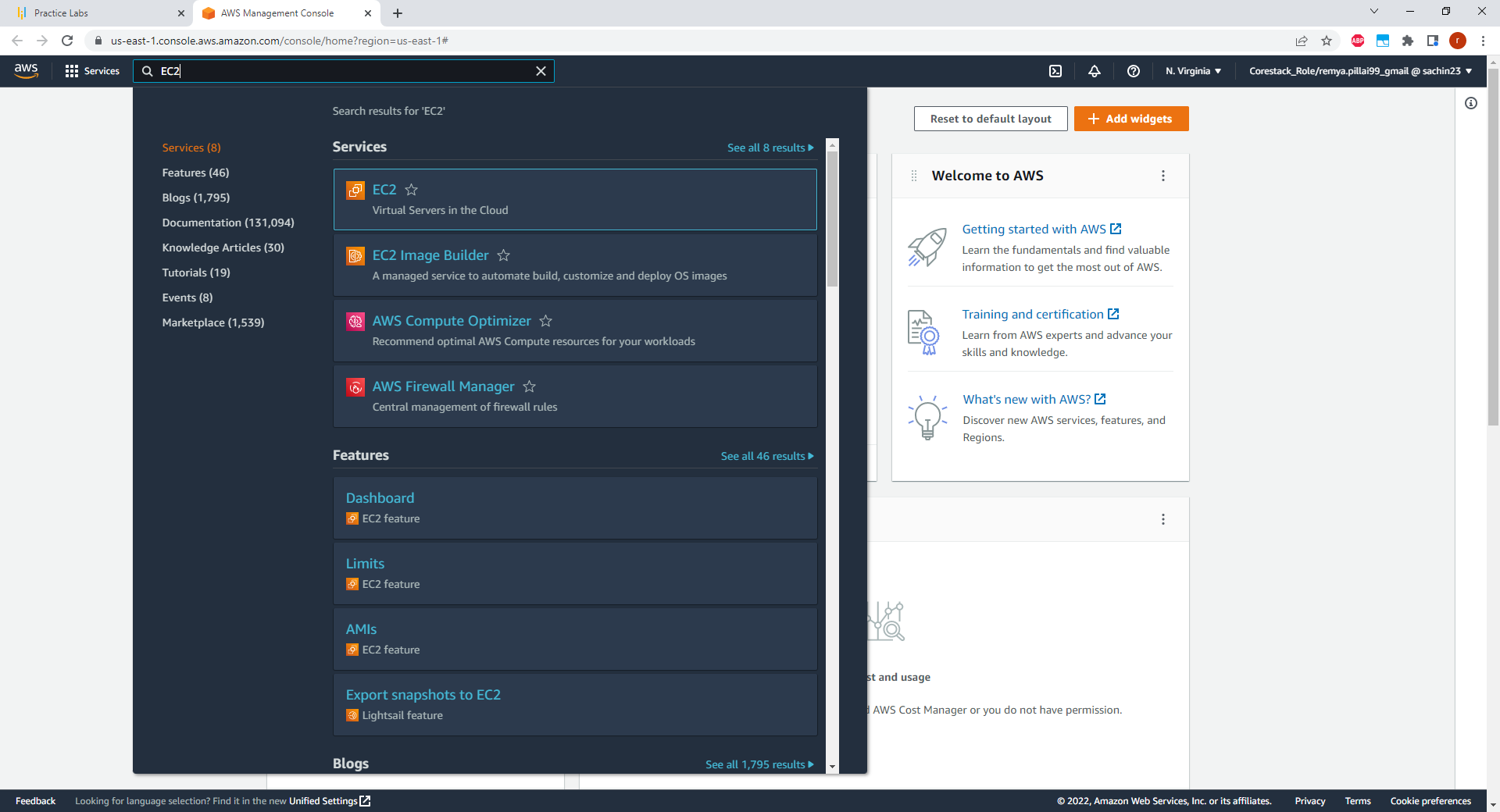
Description automatically generated

1. AWS Web Console – AWS Management Console Page

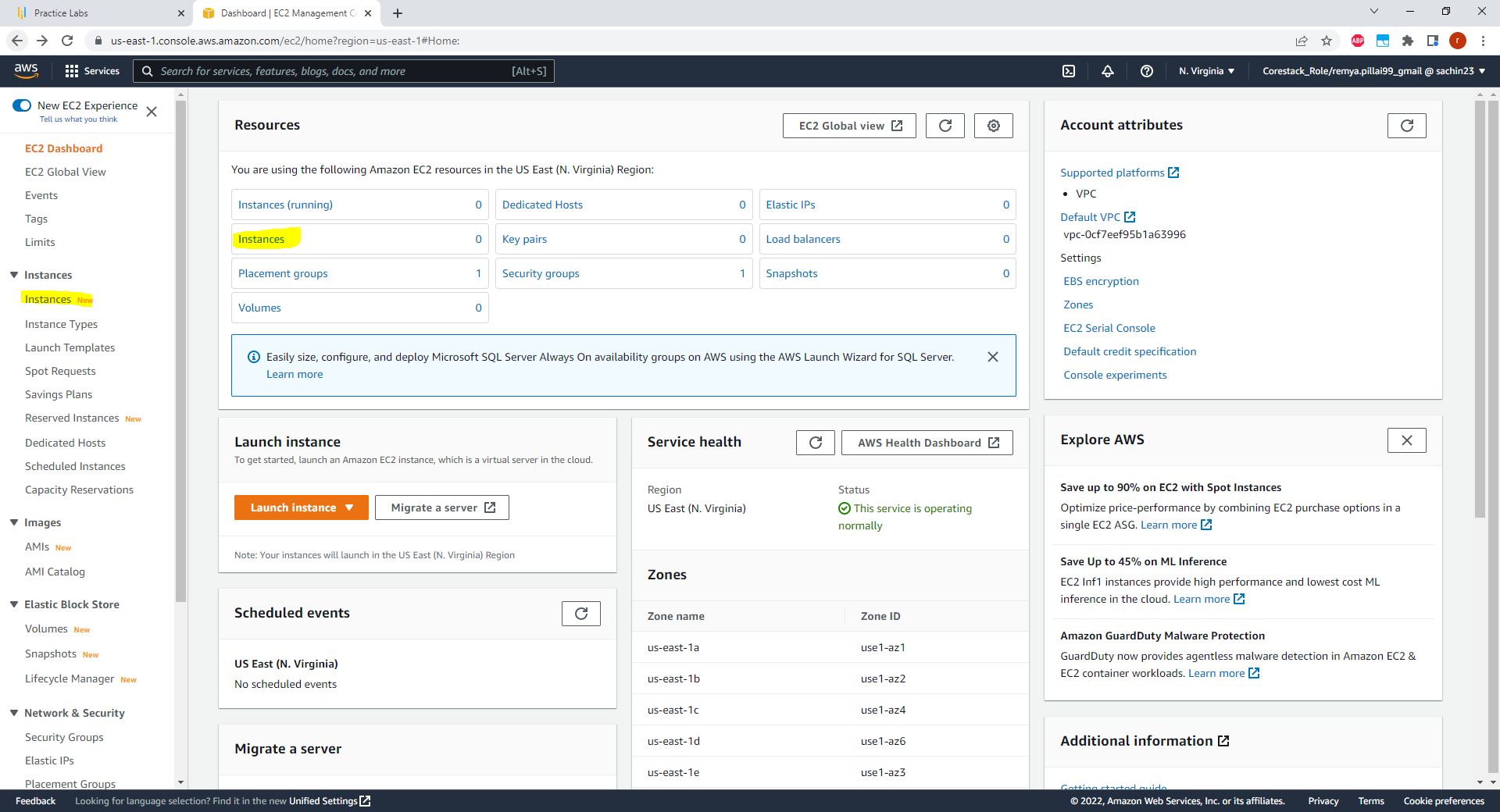
Graphical user interface, application

Description automatically generated

1. Search for EC2



1. Click on “Instances”



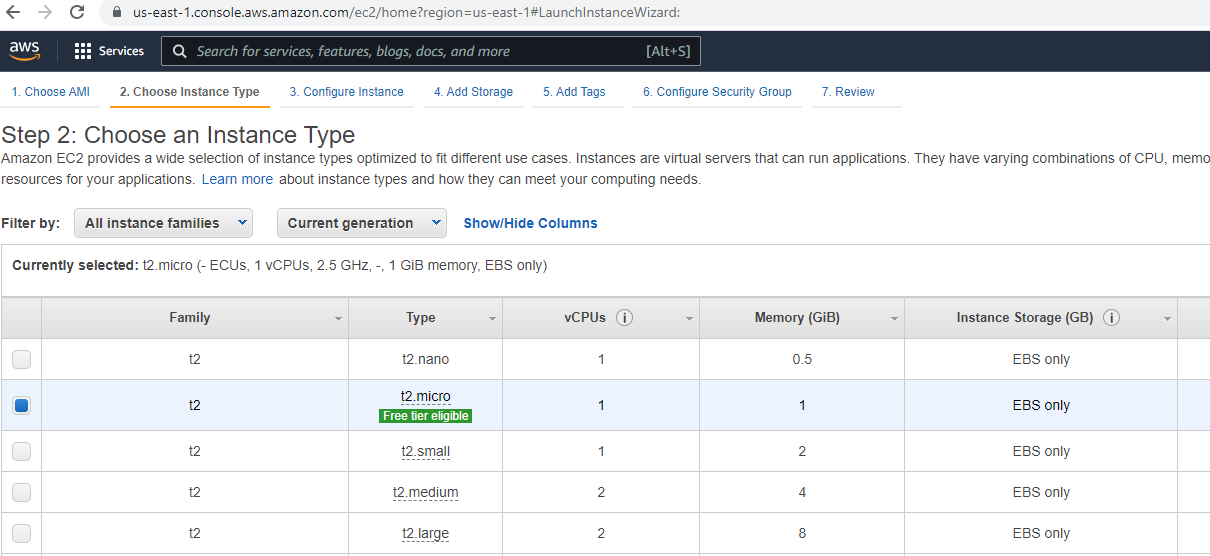
NOTE: We will be creating a New Instance as a part of this exercise

1. Click on Launch Instance - Choose AMI : Select the first option

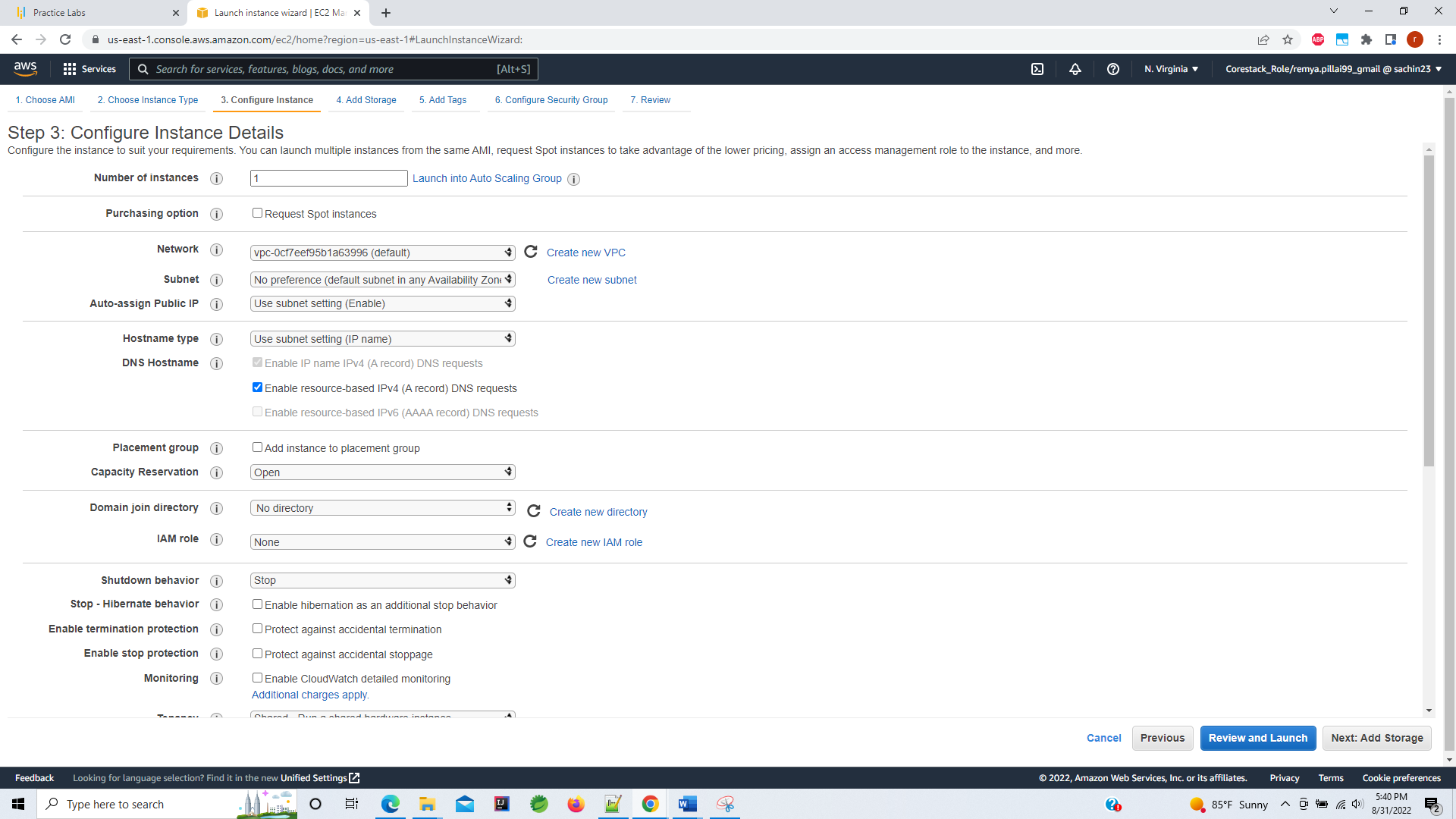
Graphical user interface, text, application, email

Description automatically generated

Select Instance type



Configure Instance Details: keep the defaults – Click next

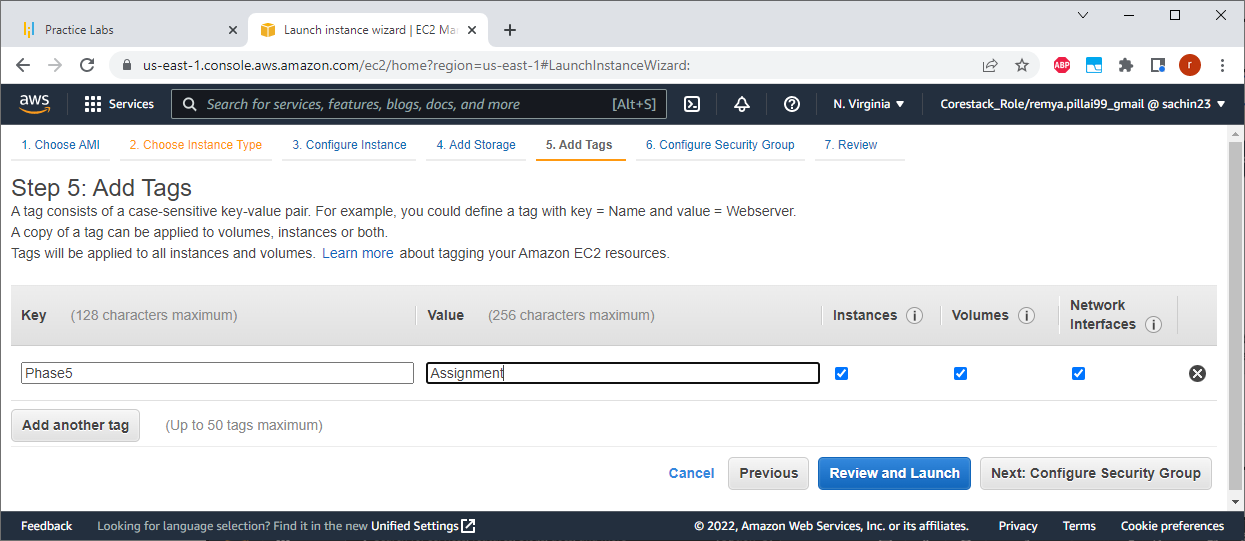


Add Storage – default selection

Graphical user interface, text, application, email

Description automatically generated

Add tag – Key value pair

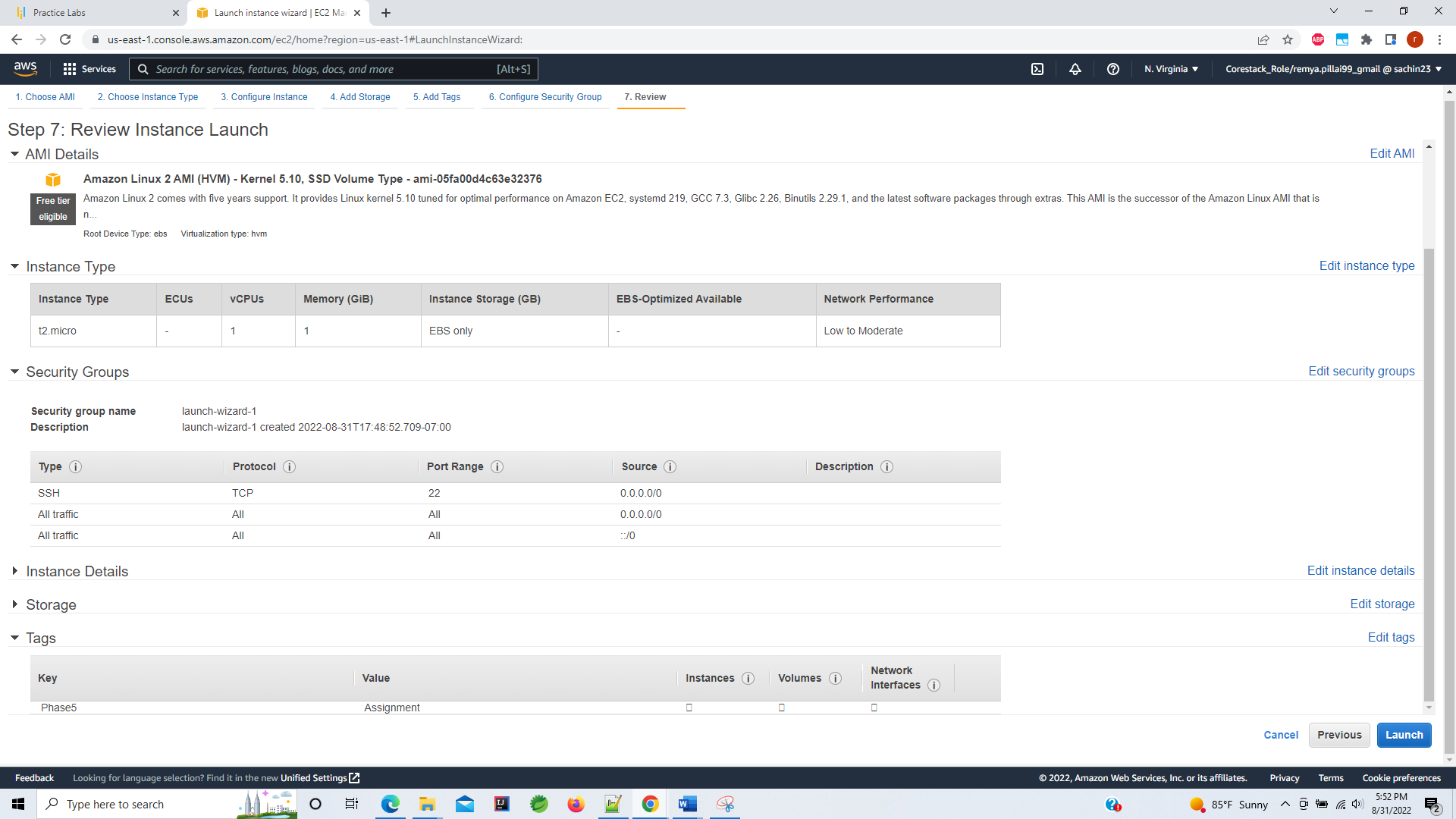


Configure Security Group: Add Type All traffic and Source – Anywhere. Click Review and Launch

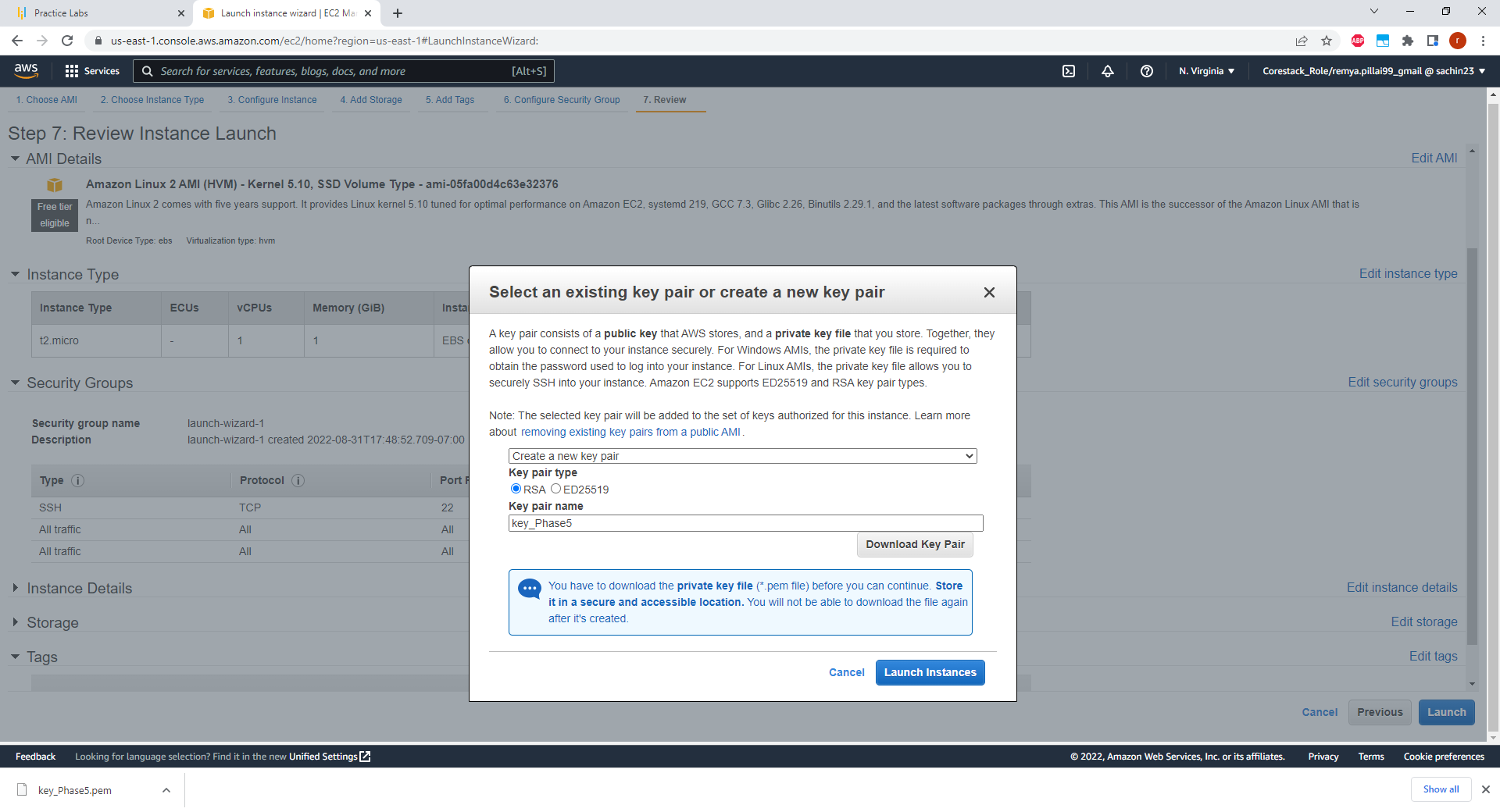
Graphical user interface, text, application, email

Description automatically generated

Review the details and click on Launch



Create a new key pair



Download the private key and store on your system

Graphical user interface, text, application, email

Description automatically generated

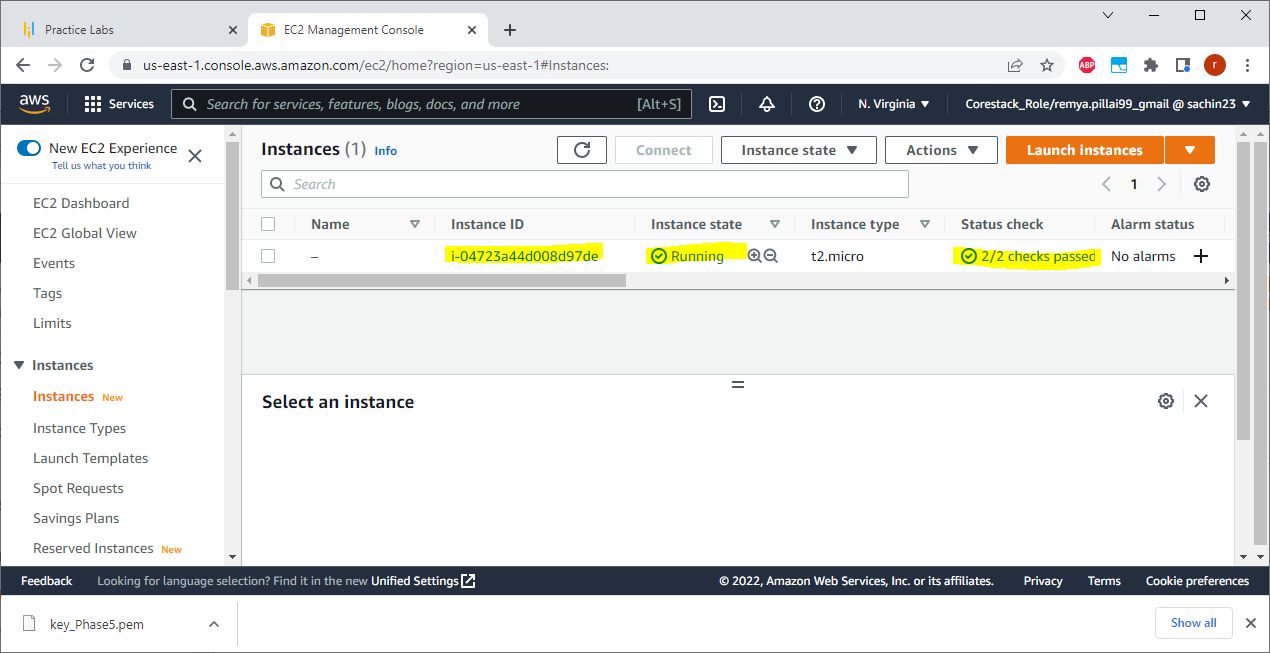
Launch instance – Click on View instance to check the status

Graphical user interface, text, application, email

Description automatically generated

A screenshot of a computer

Description automatically generated



1. Select the instance : Click on Connect

Graphical user interface, text

Description automatically generated

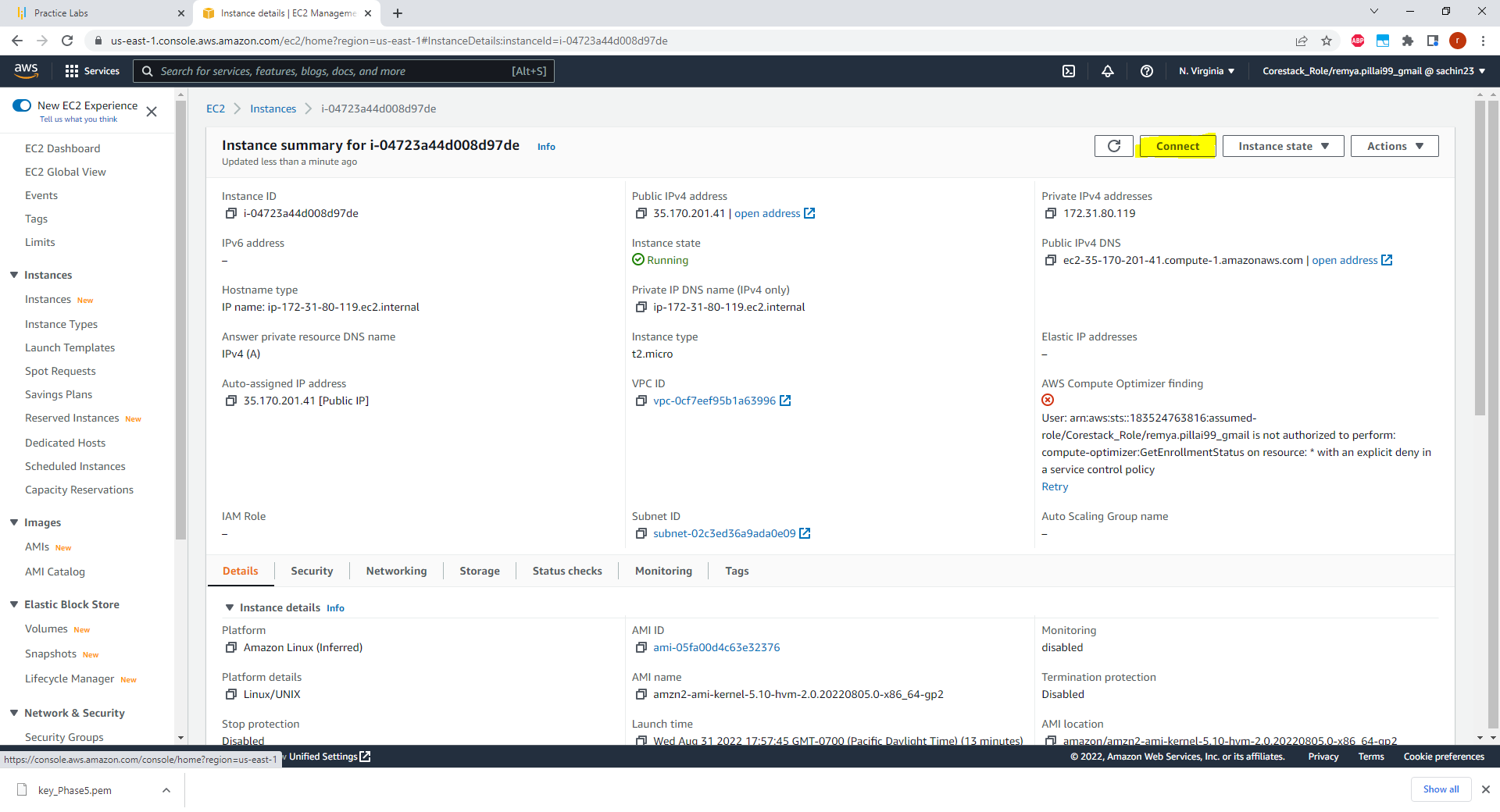
### Connect to EC2 Instance using SSH

1. Connecting to the AWS EC2 Instance from Terminal: Open Command Prompt from the folder where the keypair is stored

Text

Description automatically generated

1. Click on connect



1. Connect using SSH client Copy the highlighted cmd

Graphical user interface, text

Description automatically generated

1. Enter the below command   
    ssh -i "key\_Phase5.pem" [ec2-user@ec2-35-170-201-41.compute-1.amazonaws.com](mailto:ec2-user@ec2-35-170-201-41.compute-1.amazonaws.com)

Text

Description automatically generated

1. Install all updates : sudo yum update -y

A screenshot of a computer

Description automatically generated with medium confidence

1. Install git : sudo yum install git -y

A picture containing graphical user interface

Description automatically generated

1. Install docker : sudo yum install docker -y

Graphical user interface

Description automatically generated with medium confidence

1. Start docker service : sudo service docker start

Text

Description automatically generated

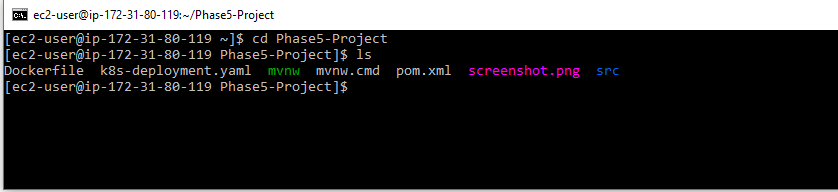
### Cloning git repository on EC2

1. Clone : git clone <https://github.com/nicks204/Phase5-Project.git>

Text

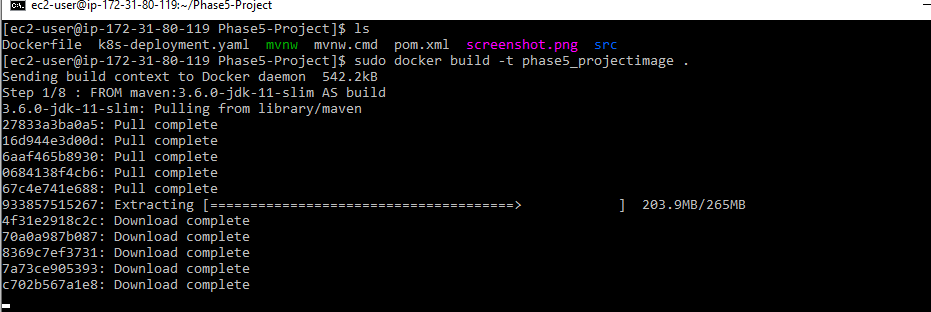
Description automatically generated

1. Go to repository folder : See the required files present



### Building the App Image

1. Run Docker build: sudo docker build -t phase5\_projectimage .



Text

Description automatically generated

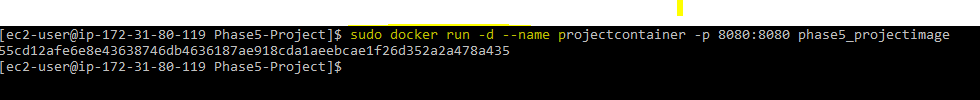
1. List docker images created: sudo docker images – You see the phase5\_projectimage

Text

Description automatically generated

### Running the Image using docker run

1. Run the image: sudo docker run -d --name projectcontainer -p 8080:8080 phase5\_projectimage



### Access the application using IpV4 address

1. See active process: sudo docker ps -a

Graphical user interface, text

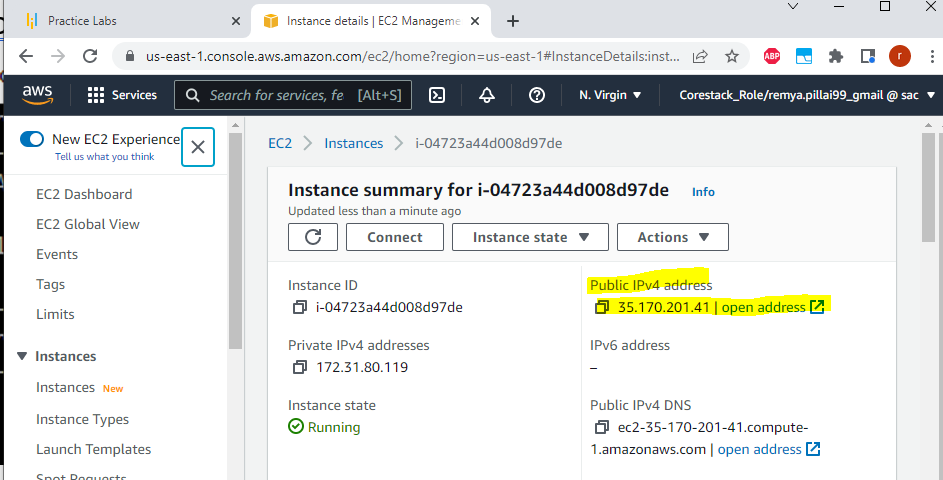
Description automatically generated

1. See logs: sudo docker attach <containerId> 55cd12afe6e8

Text

Description automatically generated

1. Access the application -Copy the public IPv4 Address



1. Add port number 8080mto address : <http://35.170.201.41:8080/> Application is launched

Graphical user interface, application, PowerPoint

Description automatically generated

1. See logs Application was deployed successfully using EC2

Text

Description automatically generatedGraphical user interface, application

Description automatically generated