Phase 5 :

Testing :

Docker

CI and CD tool Using Jenkin

Grunt done

AWS : S3 EC2 instance

Phase 4 project

Cloud Computing

Cloud is refer to network. Cloud computing providing infrastructure base upon our requirement. They are ready to do customize and we need to pay depending upon our uses.

Cloud computing provide use software ie system software as well as application software, tools, server, databases etc.

There are certain services and model working behind the scene making the cloud computing feasible and accessible for end user (developer, programmer or manager etc).

Deployment model : this models are divided into 4 types.

Public cloud : public cloud any end user can create login and use those modules base upon their requirements.

AWS or Azure etc.

Private cloud : those cloud maintain by organization.

Hybrid cloud : it is a combination of public and private.

Community cloud : for this type of cloud lot of organization support.

Service model : this service model mainly divided into 3 types

1. IaaS : Infrastructure as a Service : It provides to fundamental resources such as physical machine, virtual machine, data bases or storage etc.
2. PaaS : Platform as a Service : provide the runtime environment for application to deploy that application.

<http://localhost:4200> <http://123.56.78.1:4200>

<http://localhost:3000> (Express Js )

1. SaaS : Software as Service :this model allow us to use software application as a service to end users.

Google Form

Sales forces

Open Office

Google excel

Top most cloud vendor

AWS

Azure

Google cloud

Oracle cloud

A to Z service

AWS : AWS allow use to create 1 year student logic with few free tier modules.

2 Rs.

AWS S3 (Simple Storage Service)

It is a type of service provided by AWS which help share any type of data

Java project, angular project, war, ear file, .zip folder.

EC2 Instance : AWS Elastic Compute cloud : EC2 instance provide virtual lab or server with unique IP Address. First we need to create EC2 instance with Type of os and RAM and external memory.

Then we need to connect EC2 instance using SSH Client or putty client and install required software

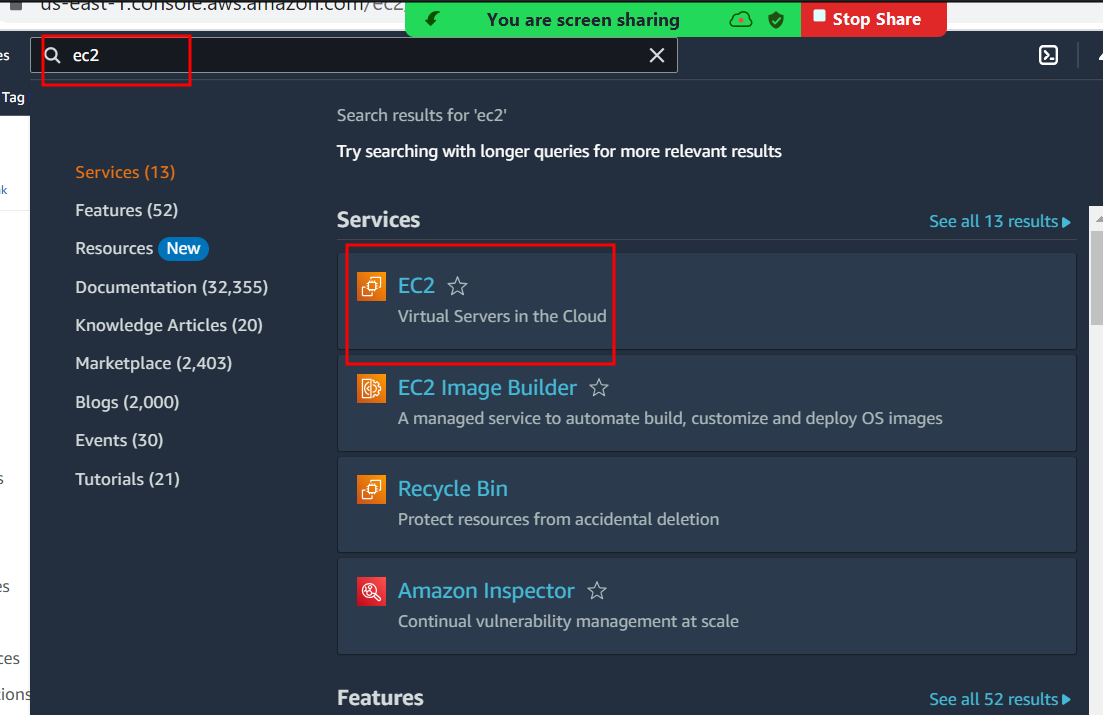
And then deploy the application once the application running in EC2 instance we can access that application using Unique IP Address provided by EC2 instance.

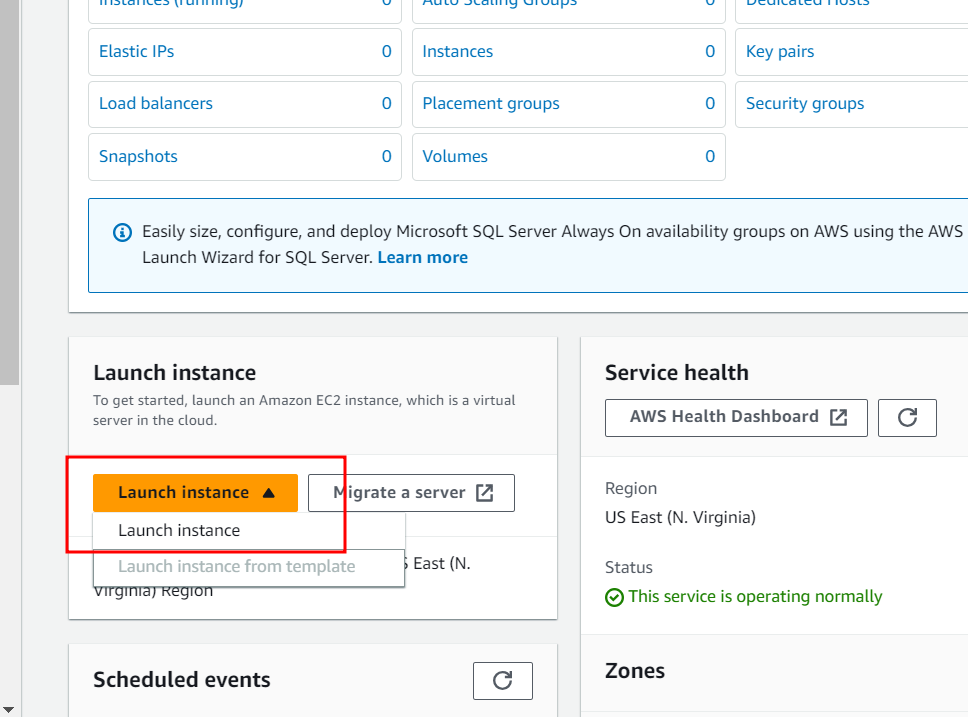
<http://localhost:4200>

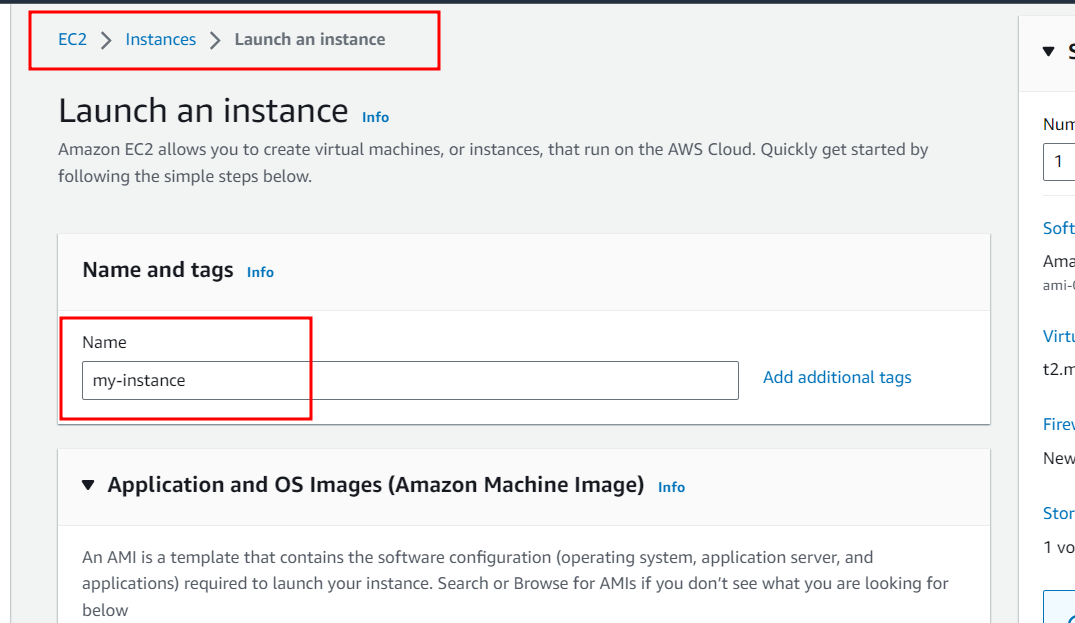
127.0.0.1

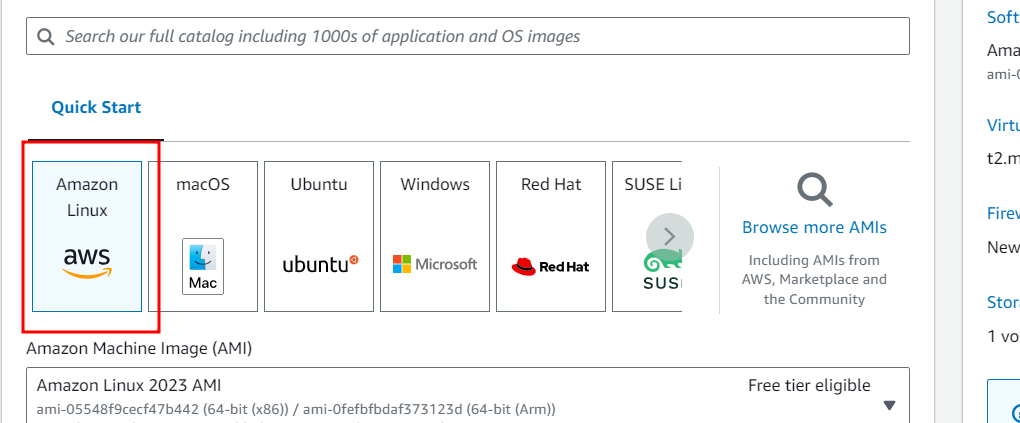
Default ip address for every machine when machine is offline.

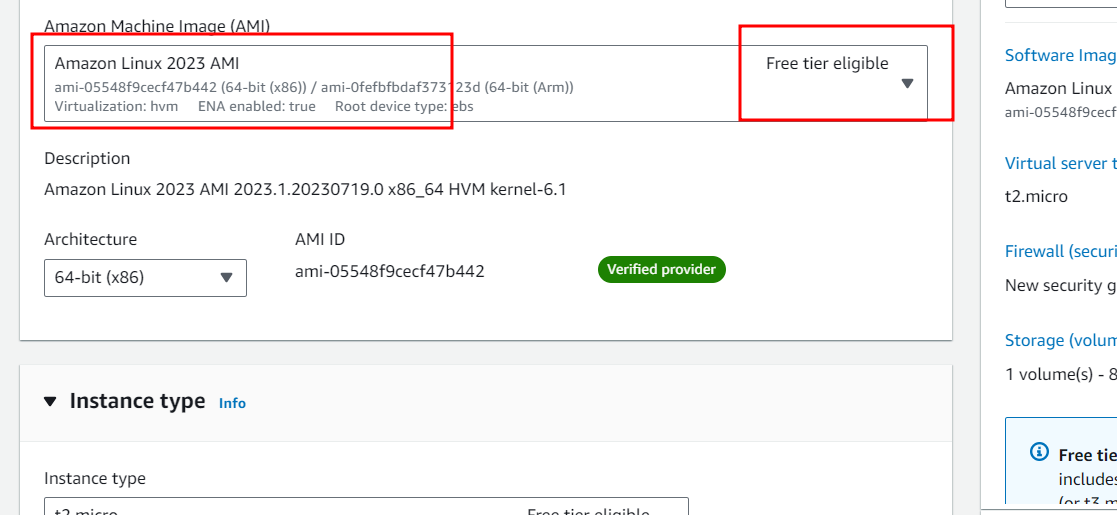
Steps to create EC2 instance

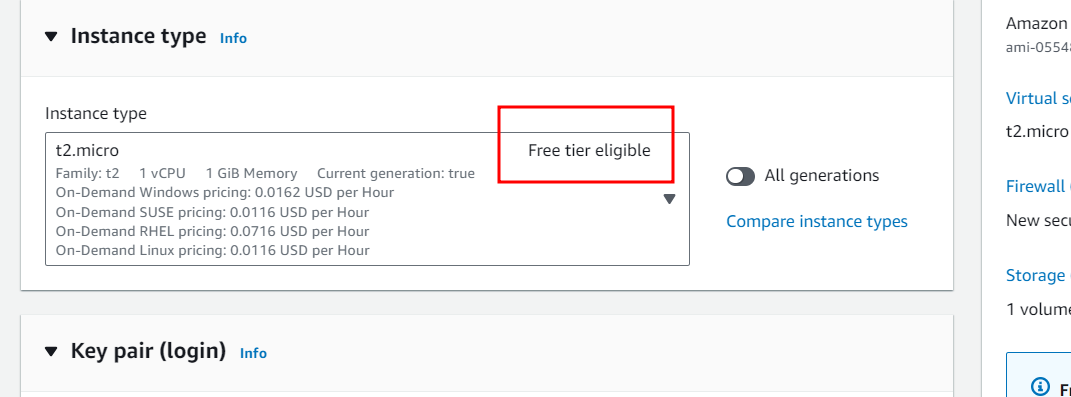


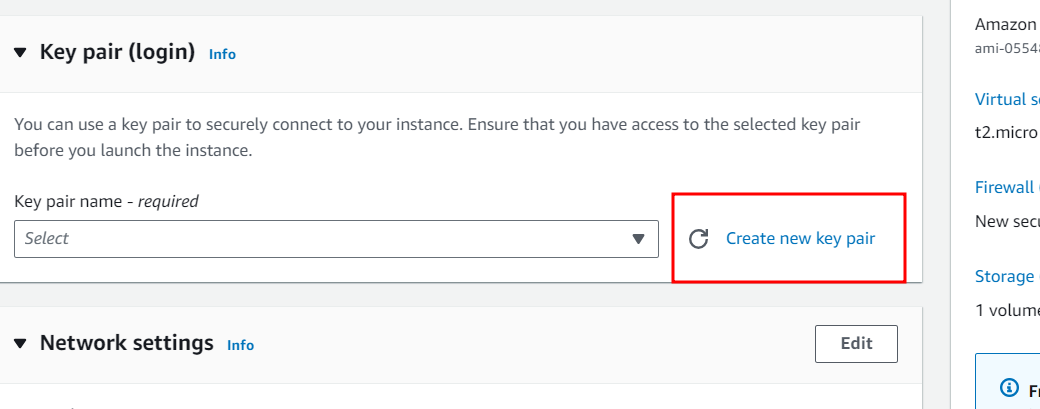


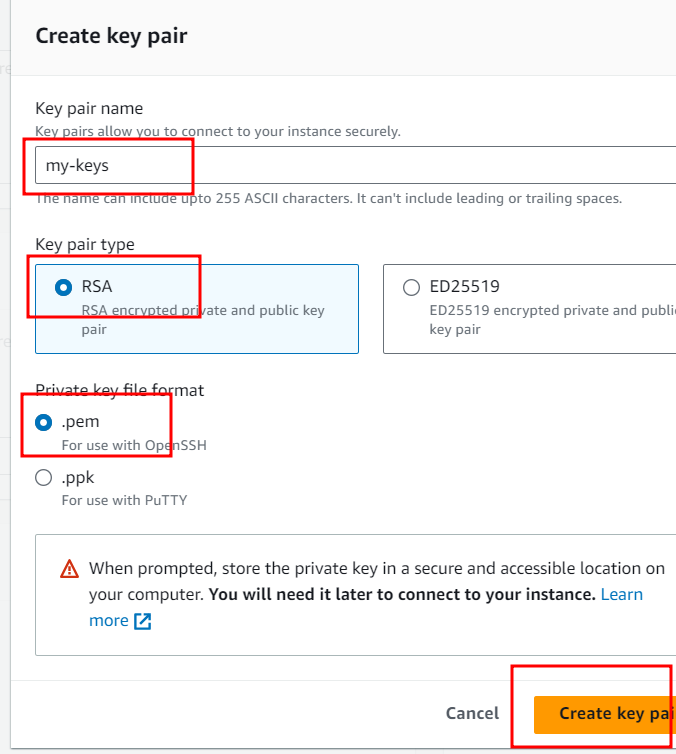












Create angular project in your local machine or virtual lab with simple your message in the project.

Then build the project using command as ng bcuild

In Docker hub you pushed your angular project.

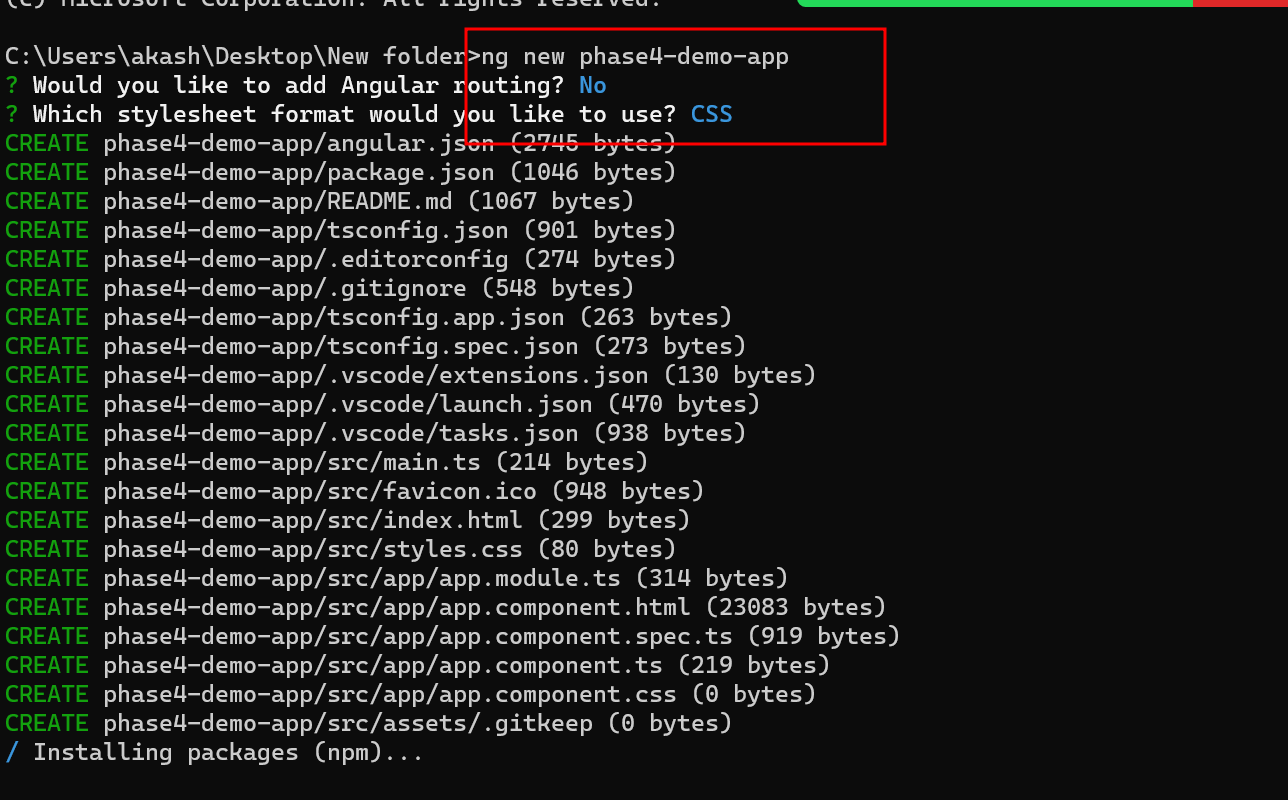
Then you need to create EC2 instance

Connect to EC2 instance using SSH

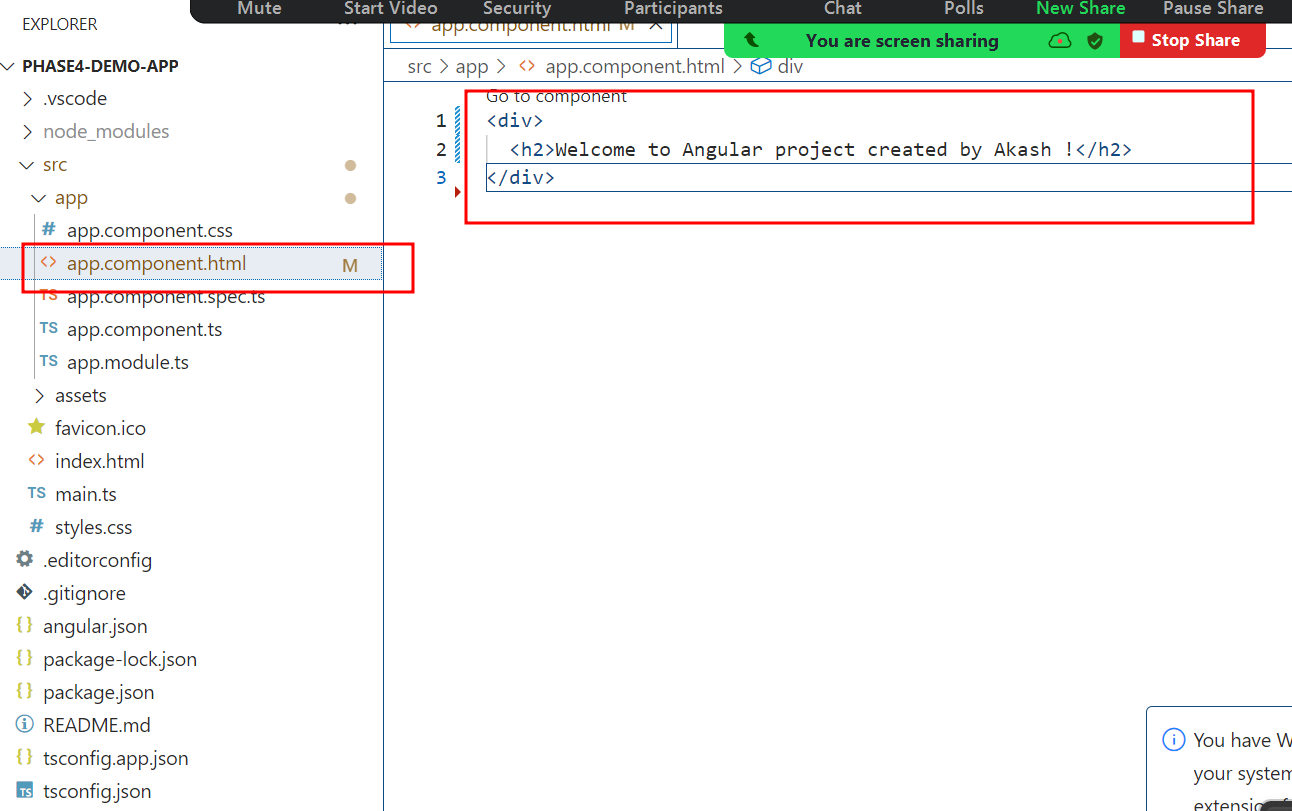
In EC2 instance install docker and pull the image which you pushed in Docker hub.

And run this image in EC2 instance so EC2 instance provide you ip address so you can view your project with IPAddress and port number

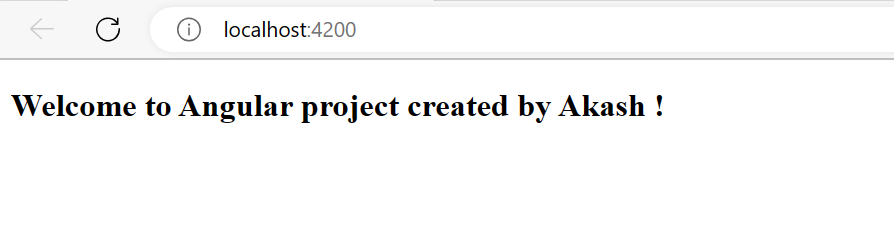
Create the angular project using ng new dphase4-demo-app



Edit in App.component.html

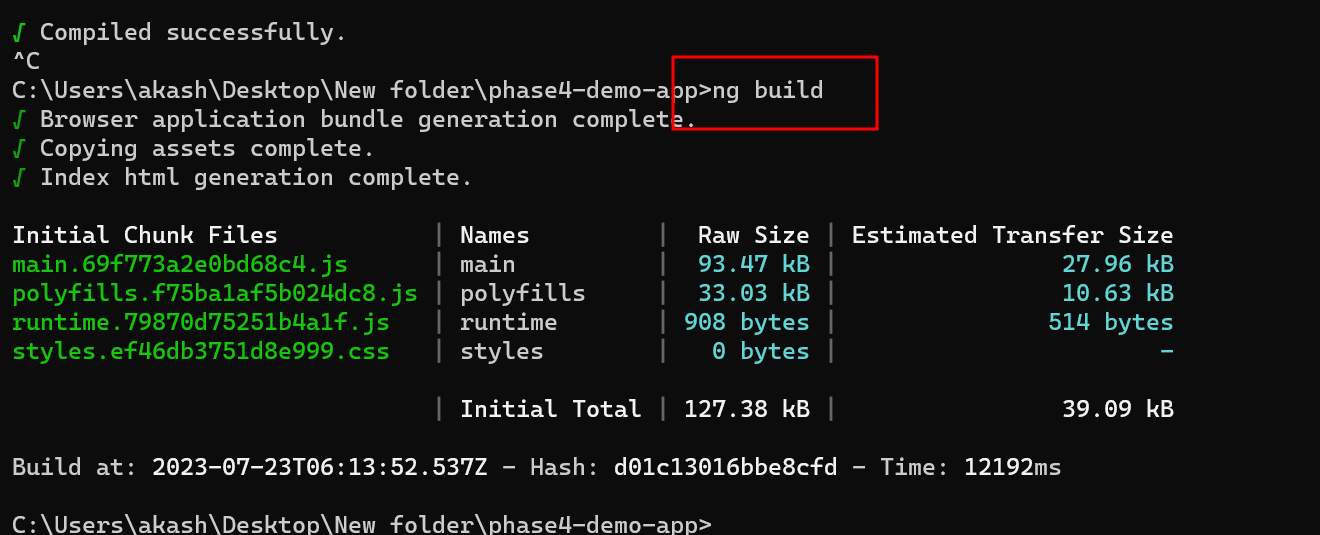


Using ng serve -o

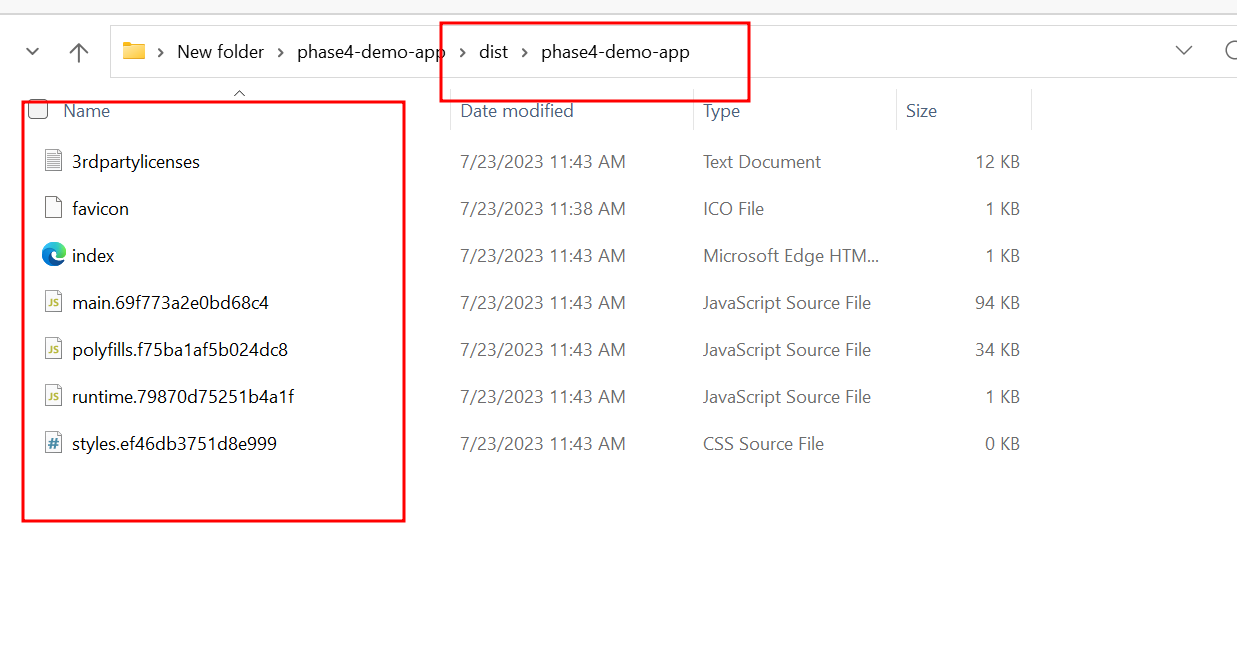


Angular provided default web server which run on default port number 4200 in development mode.

After project creation we need to build the project using command as ng build

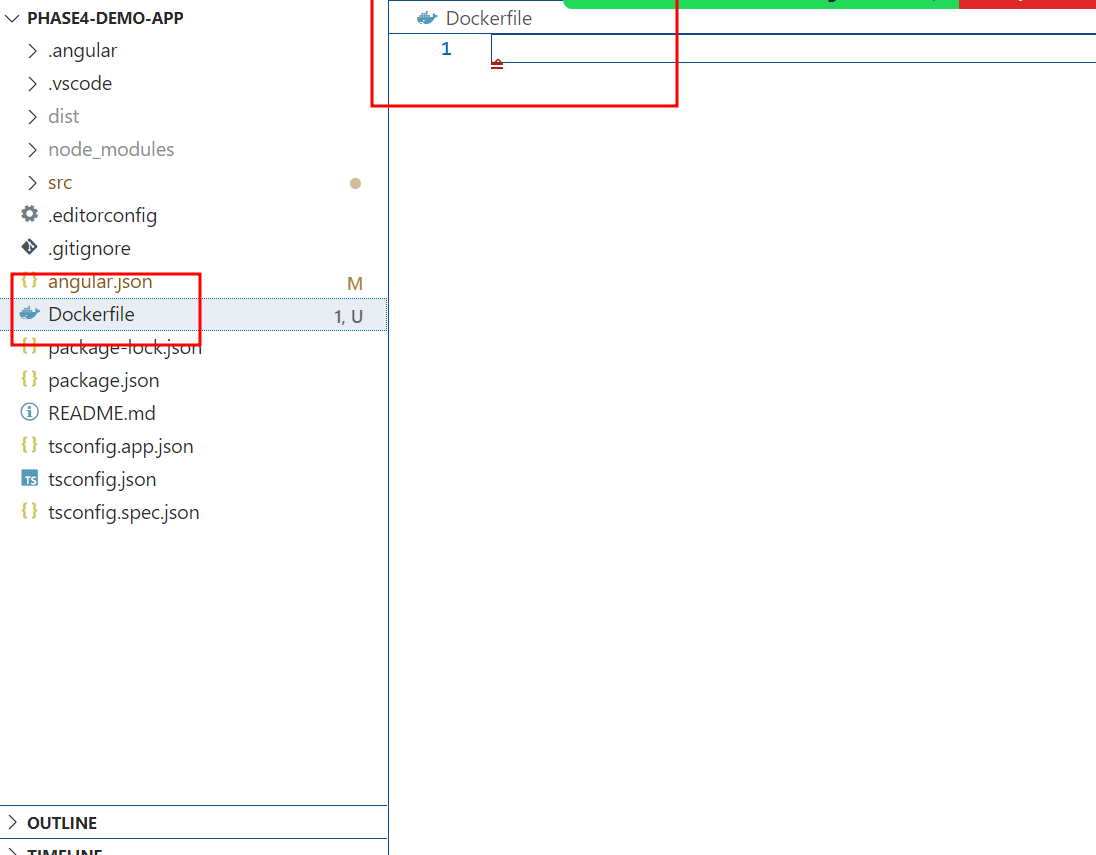


Once you build it will create dist folder which contains all build files.



We need to create the image using docker.

Now we will create the image for our angular application using nginx server images provided by docker hub.



Dockerfile

FROM nginx

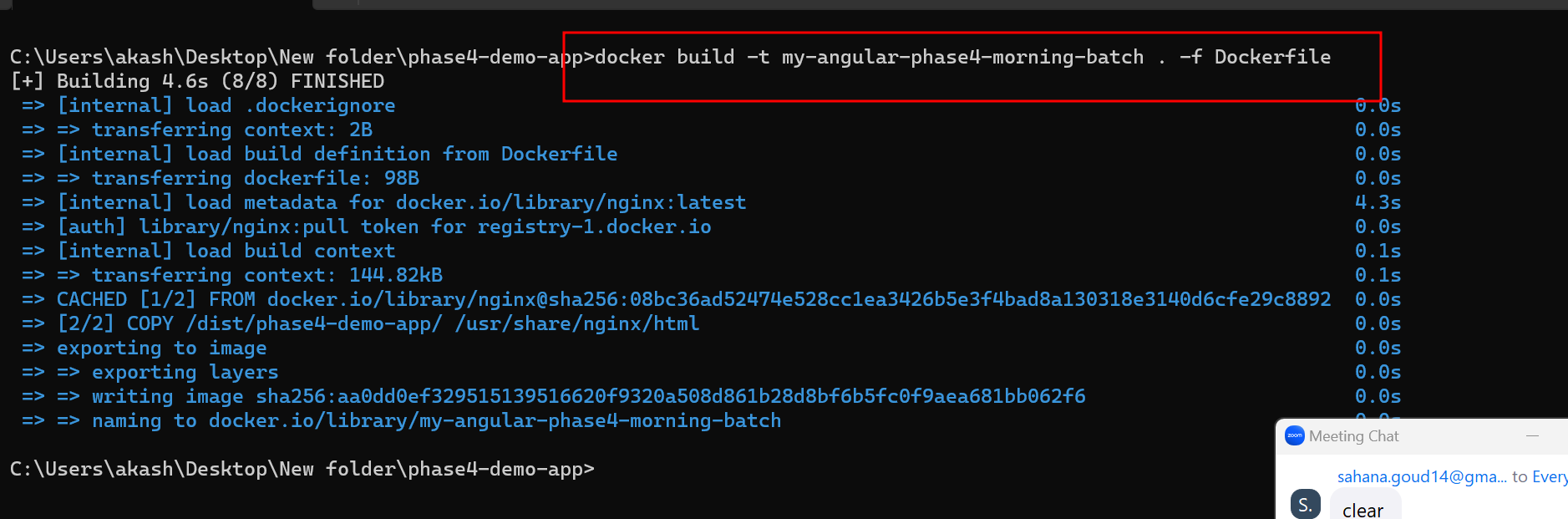
COPY /dist/phase4-demo-app/ /usr/share/nginx/html

To create the image

docker build -t my-angular-phase4-morning-batch . -f Dockerfile

In virtual lab please execute same command to create image with sudo prefix

sudo docker build -t my-angular-phase4-morning-batch . -f Dockerfile

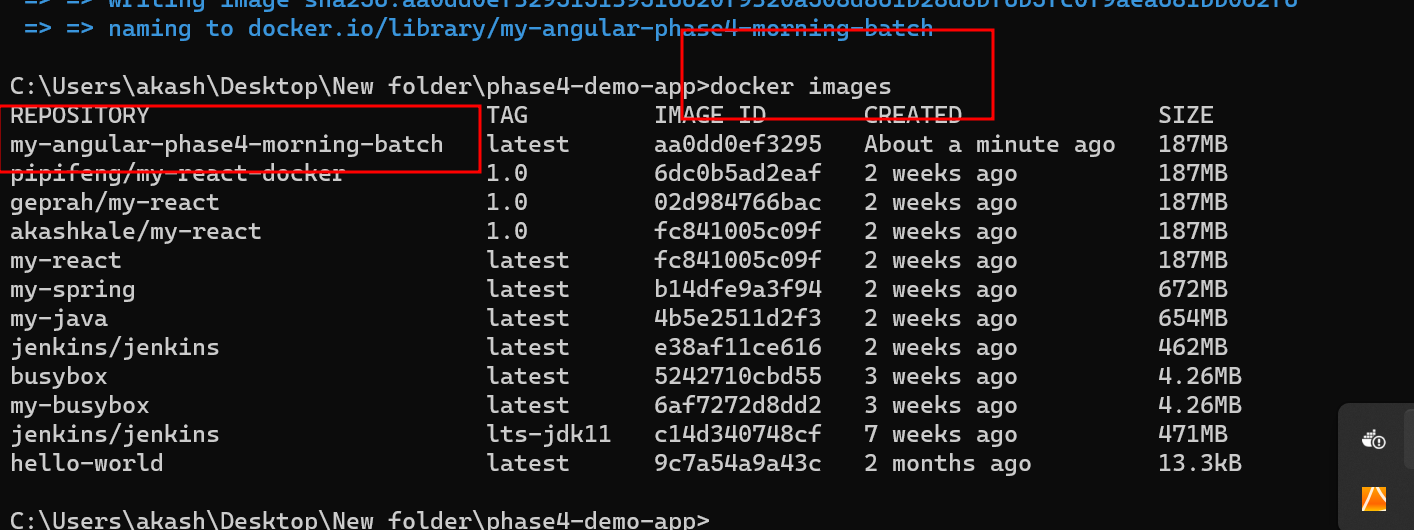


Then check image created or not.

**docker images**

Or

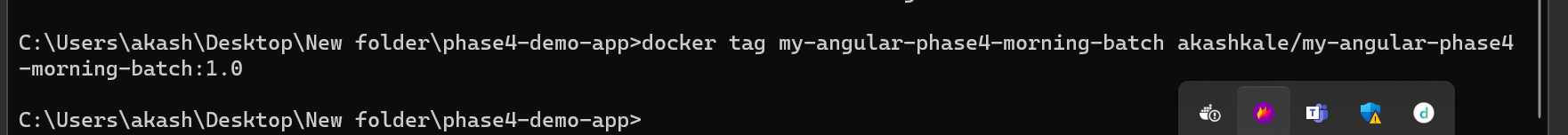
**sudo docker images**



Before publish we need to create the tag for that images

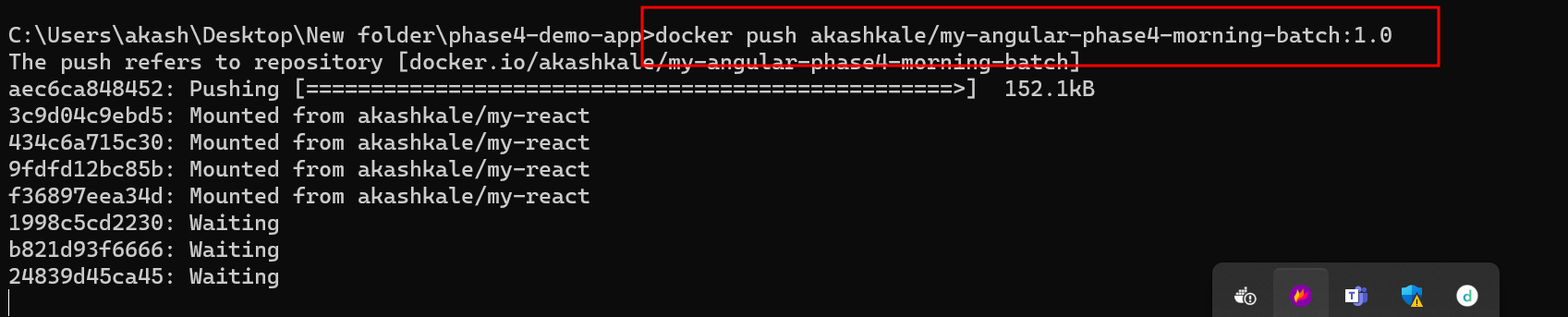
**sudo docker tag imageName dockerHubAccountName/imageName:version**

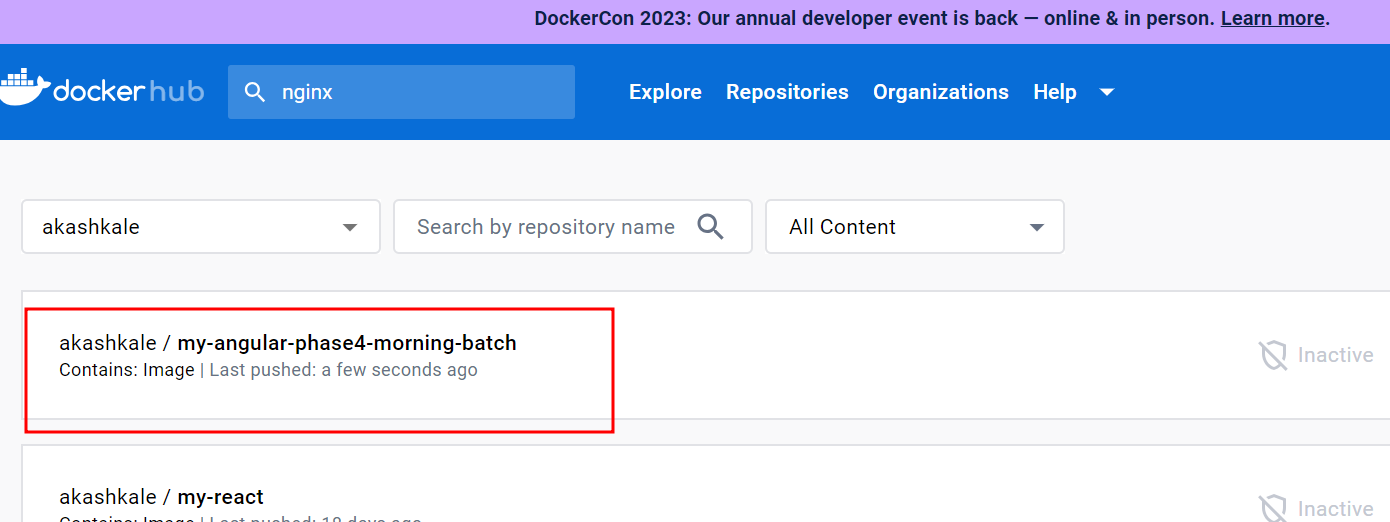
**docker tag my-angular-phase4-morning-batch akashkale/my-angular-phase4-morning-batch:1.0**



Then after created tag we need to push this image to Docker hub account.

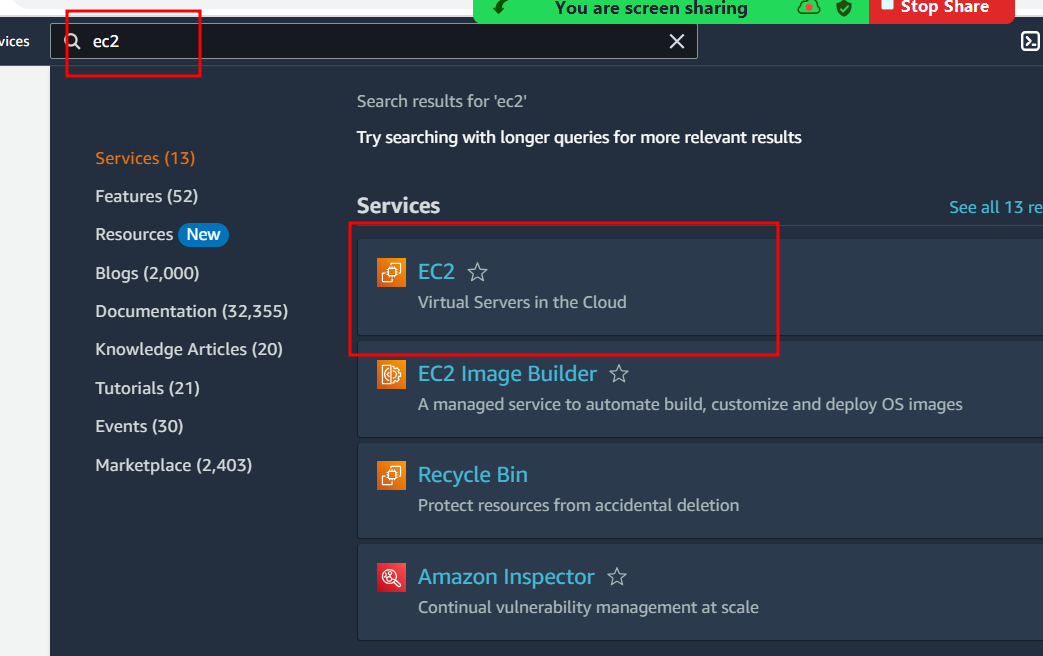
**sudo docker push dockerhuaccountid/imageName:version**



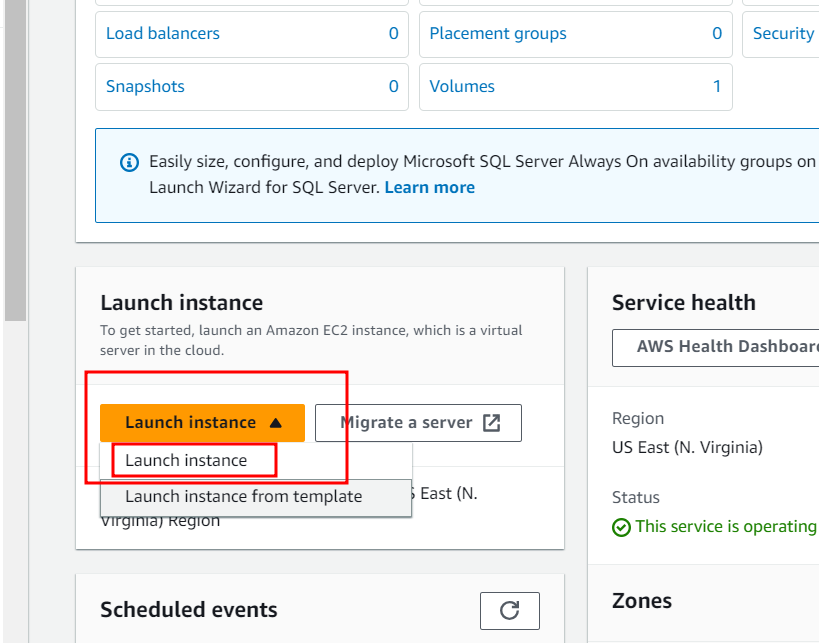


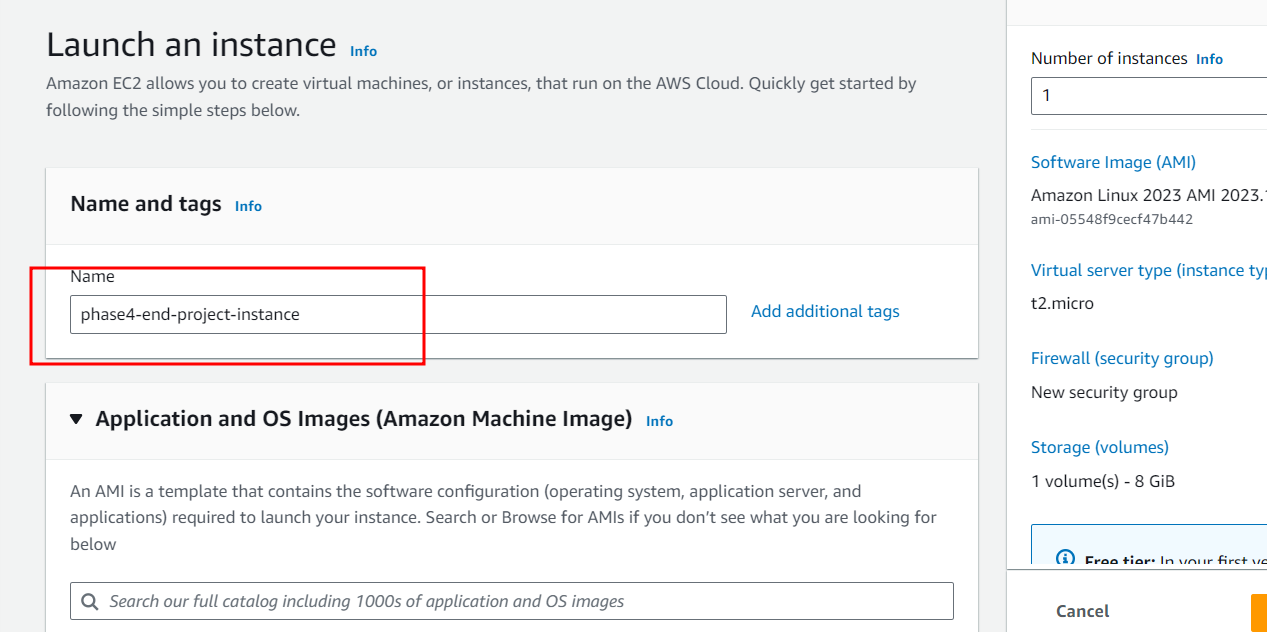
Now you need to login for AWS account

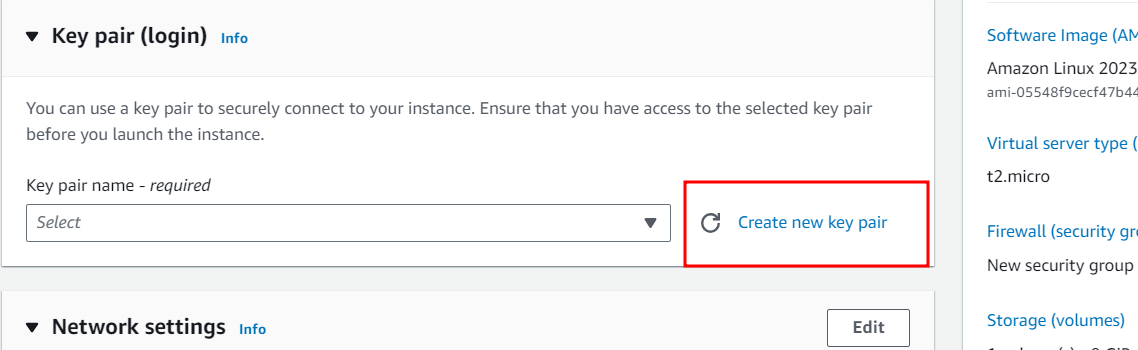
And search EC2 instance

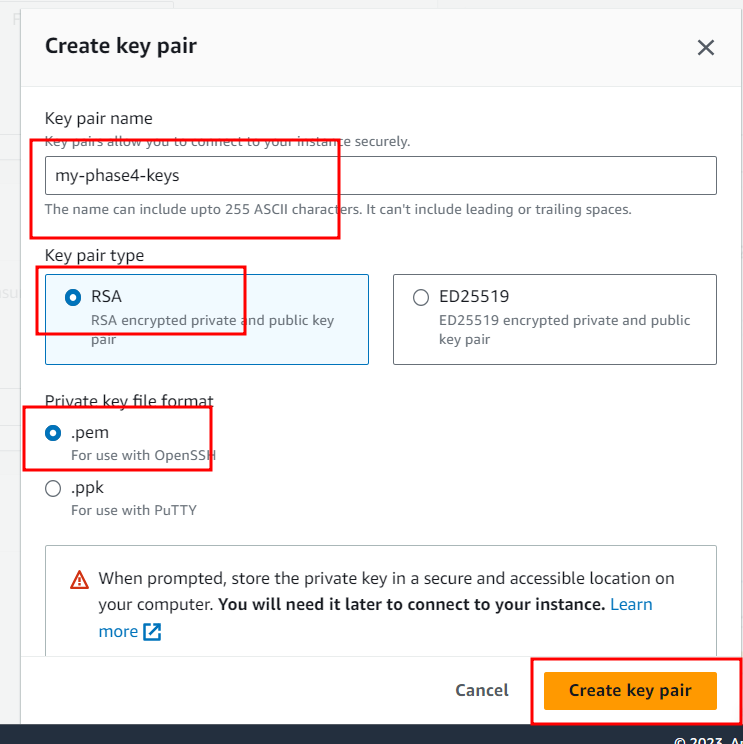


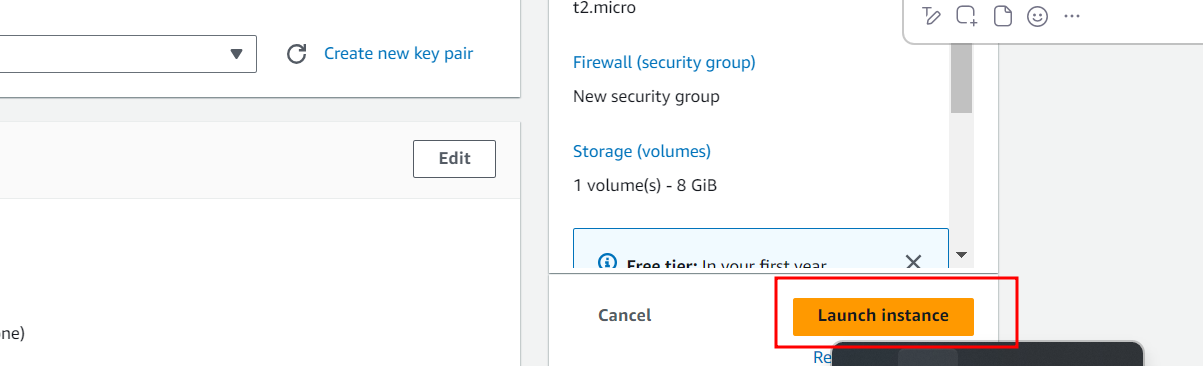
Now create EC2 instance

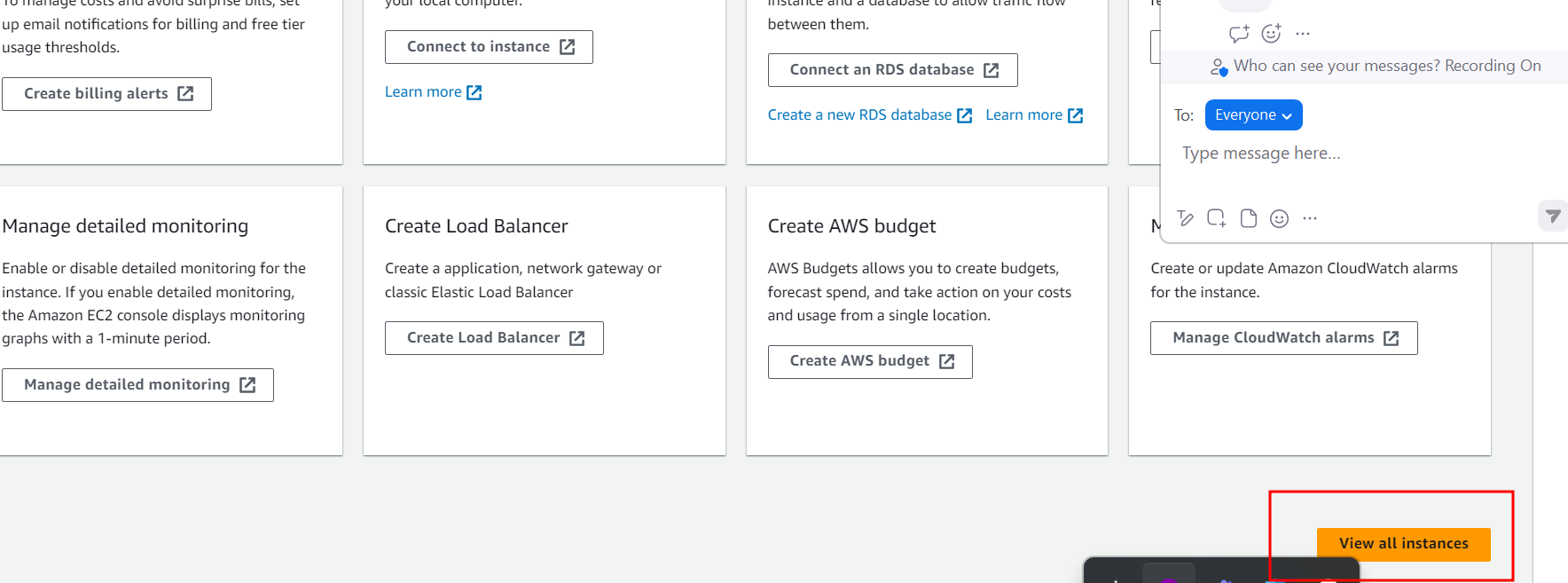


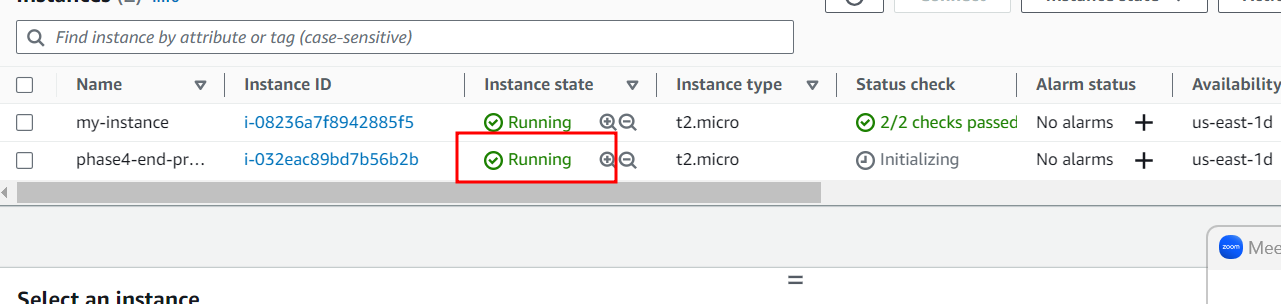




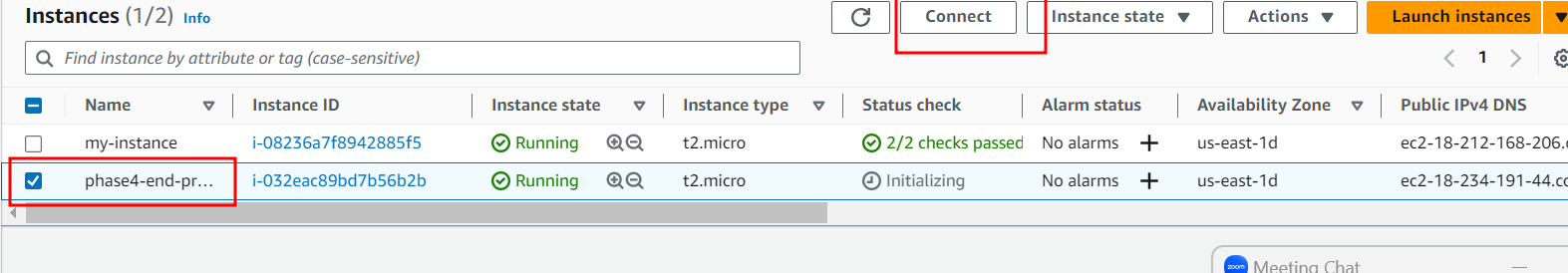


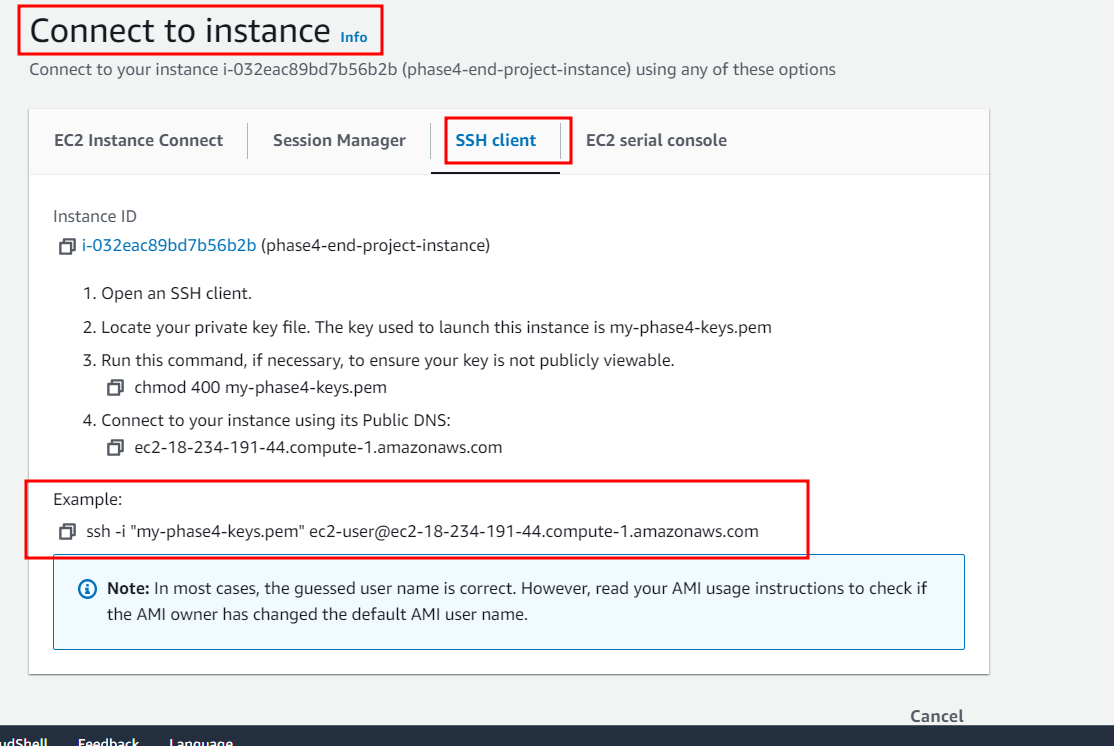






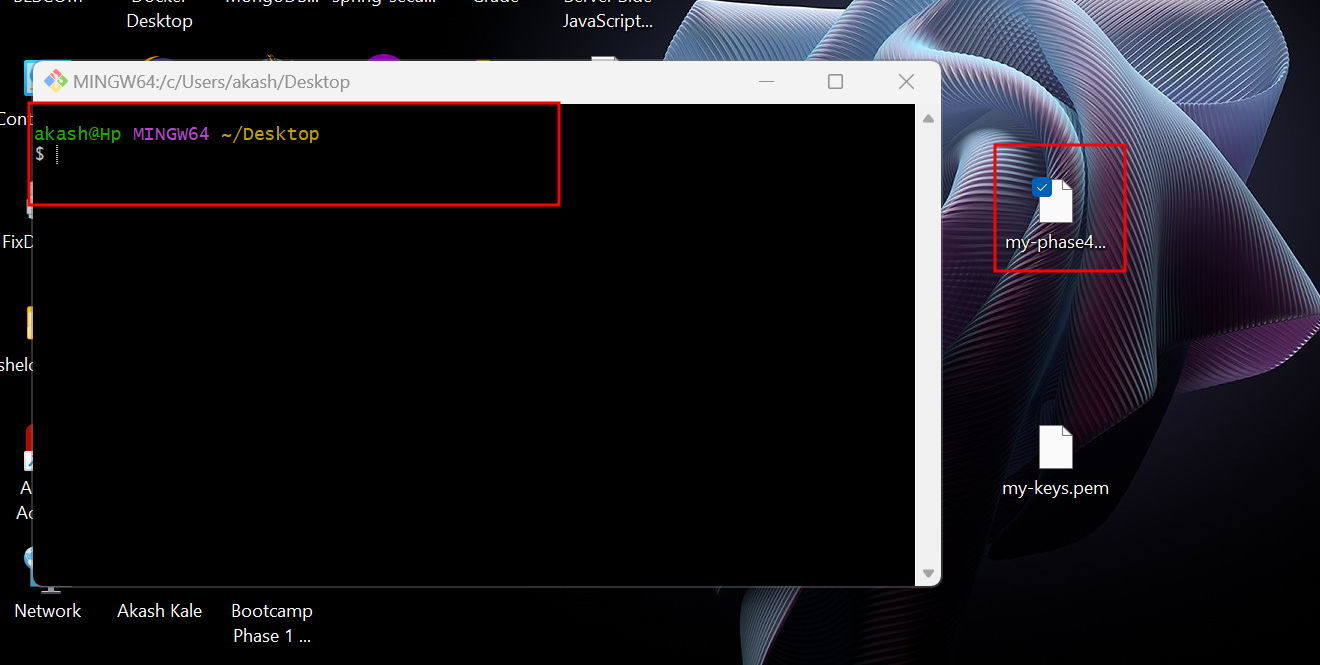
To connect this instance using SSH client please select check box of instance and click connect button.





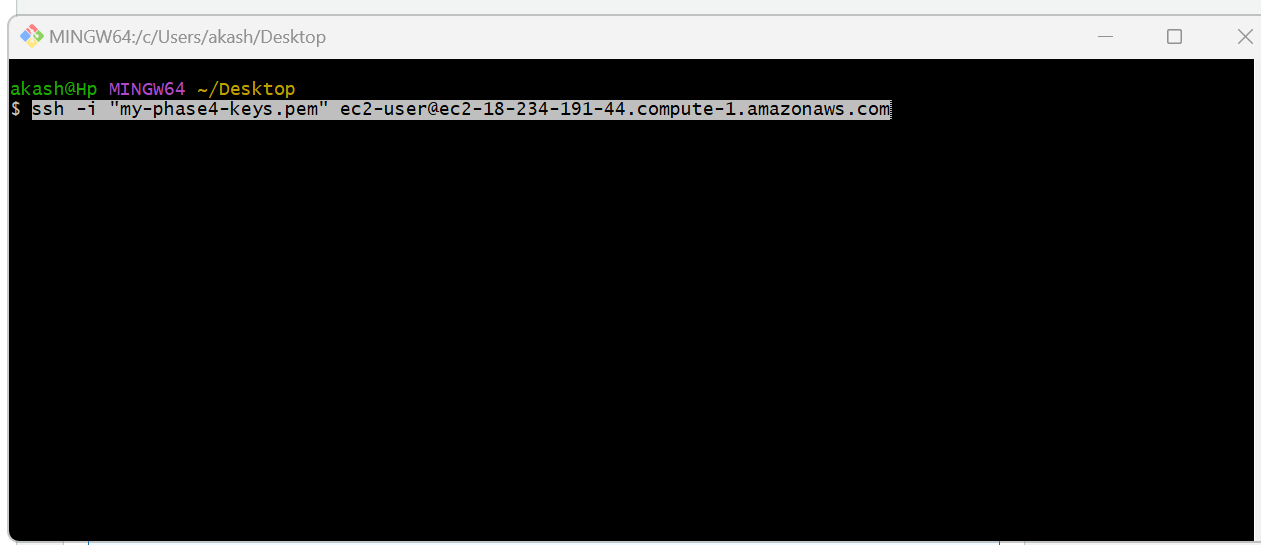
Copy the example command and open terminal in virtual machine if .pem key present in virtual machine.

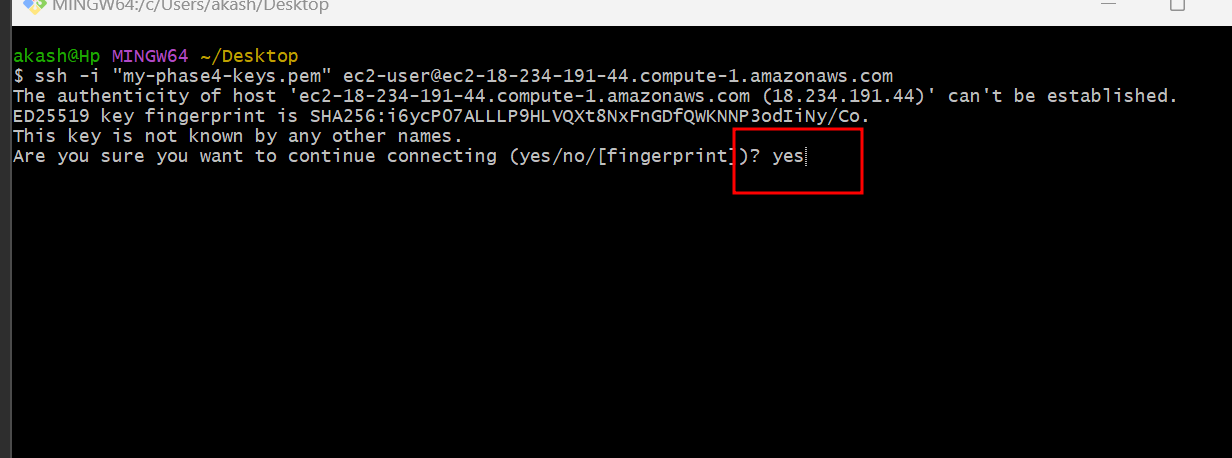
If key present in local machine please open git bash command.

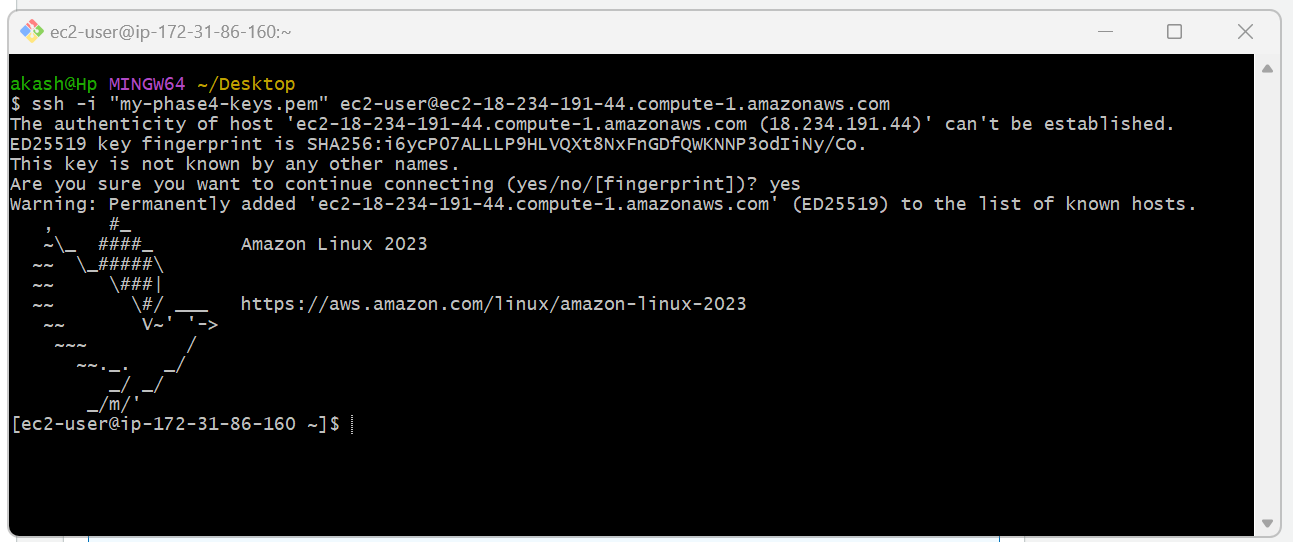


Then paste the ssh command

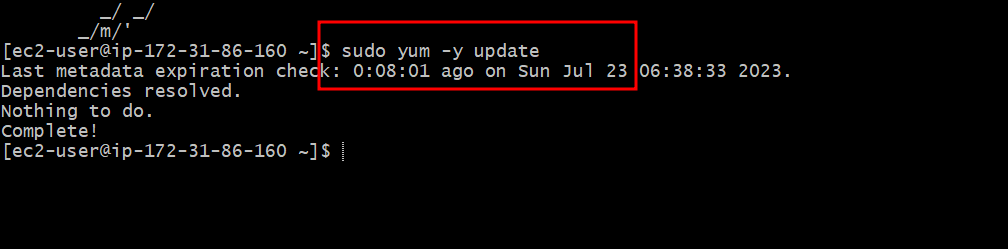
in Git terminal if window user or unix or virtual lab open normal terminal





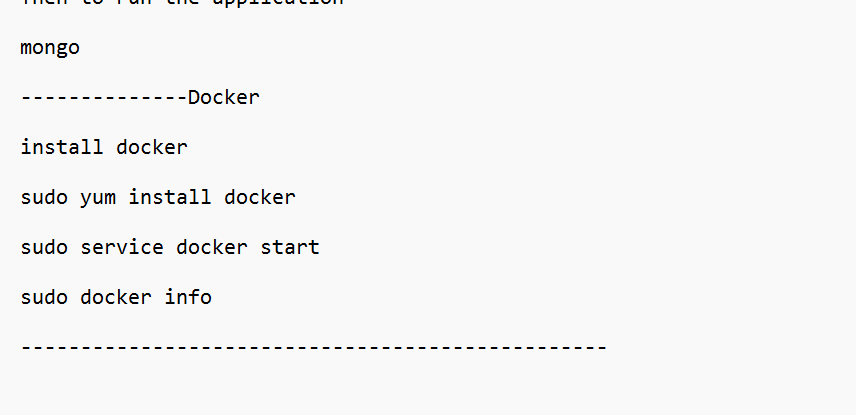


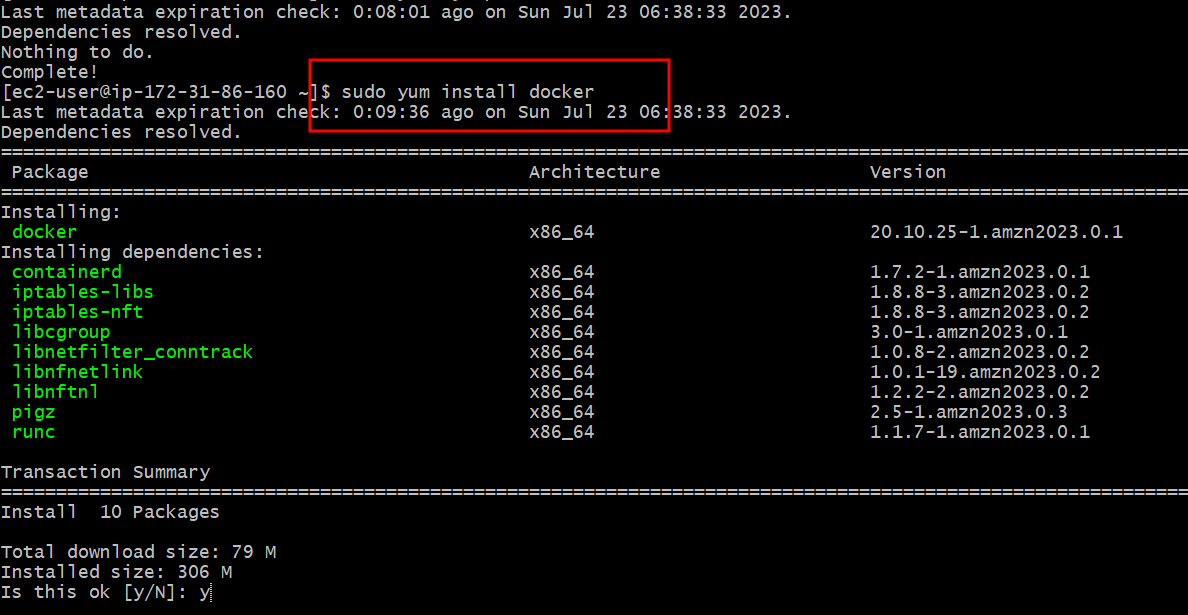
This is virtual lab terminal.



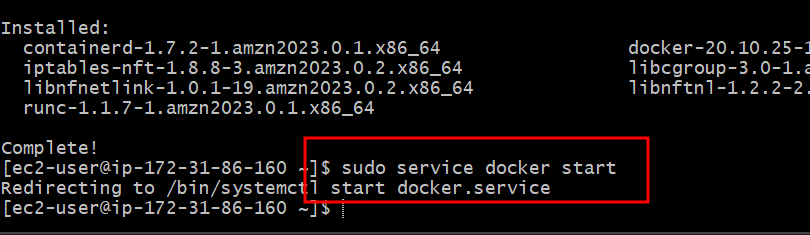
Top command is use to update EC2 instance.

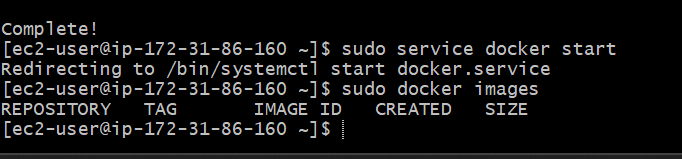
Now we will install docker in EC2 instance using command as





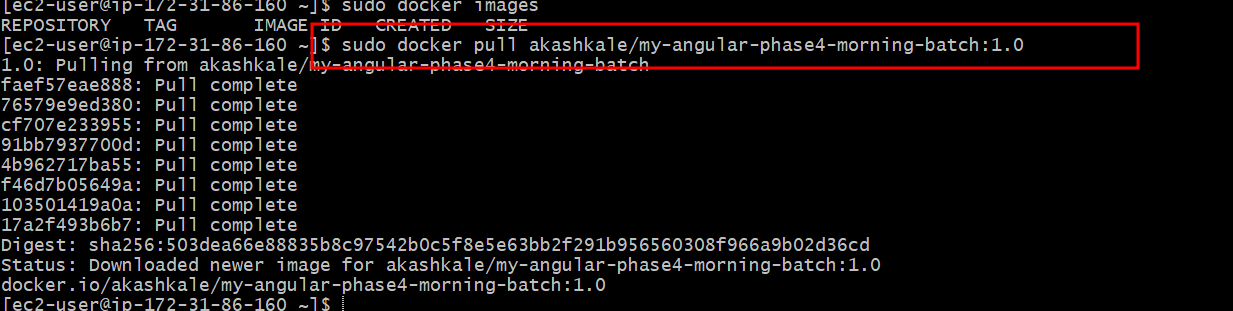
Now we need to start the docker



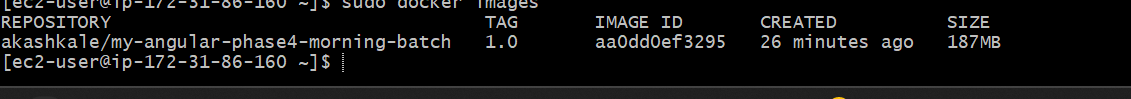


Now please pull the images from docker hub account which we already published.

**docker pull dockerhubaccount/imagename:version**

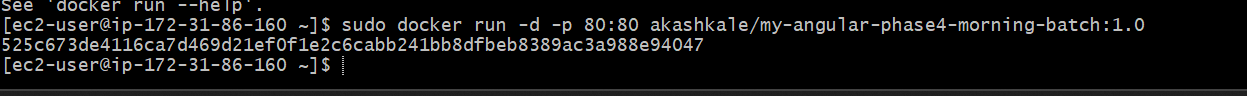


Please check this images

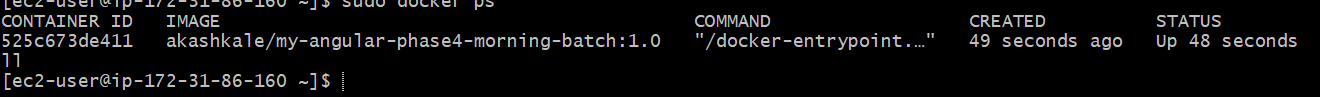


Now we need to run this images in EC2 instance (note this image running using nginx server and nginx server default port number is 80).

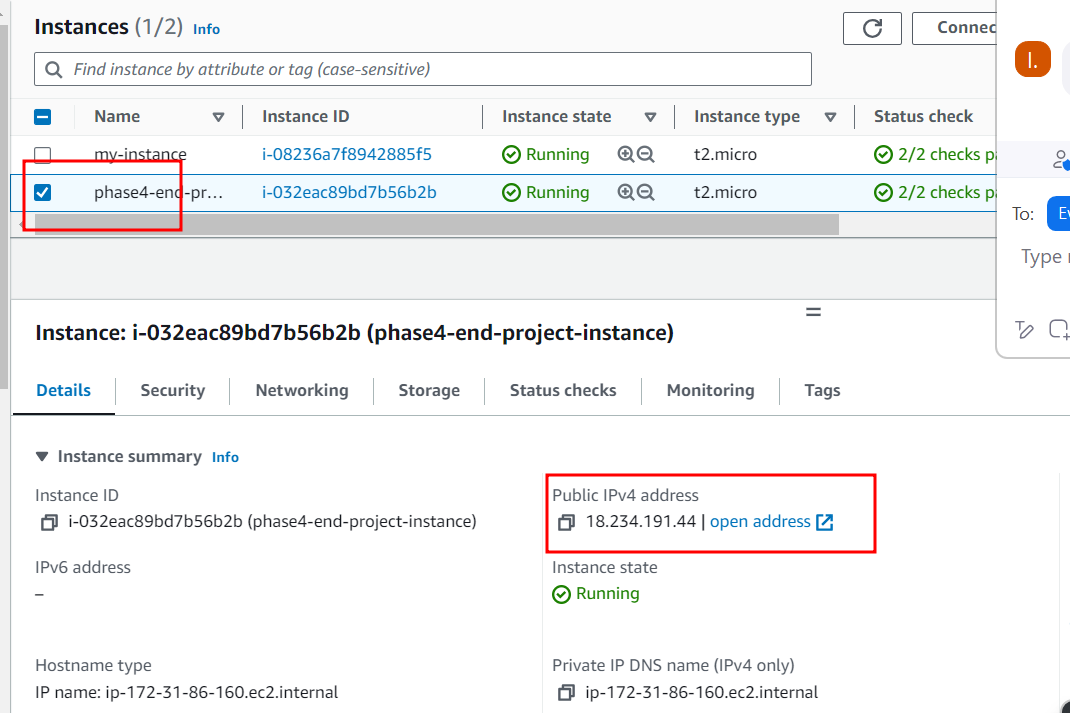
**docker run -d -p 80:80 imagename:version**

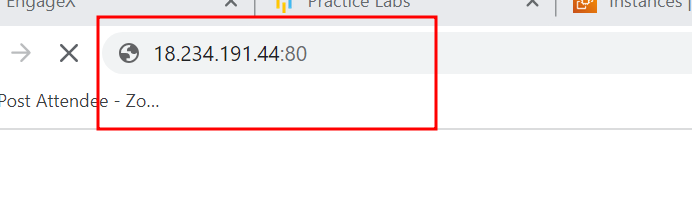


To check images running or not use docker ps



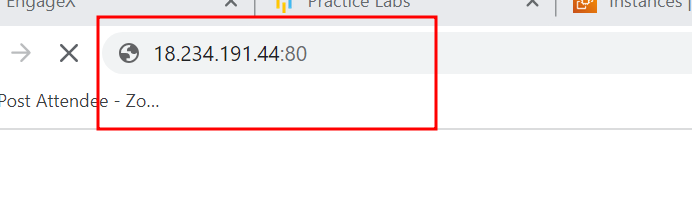
To view this application we need to find IpAdderss of our ec2 instance.



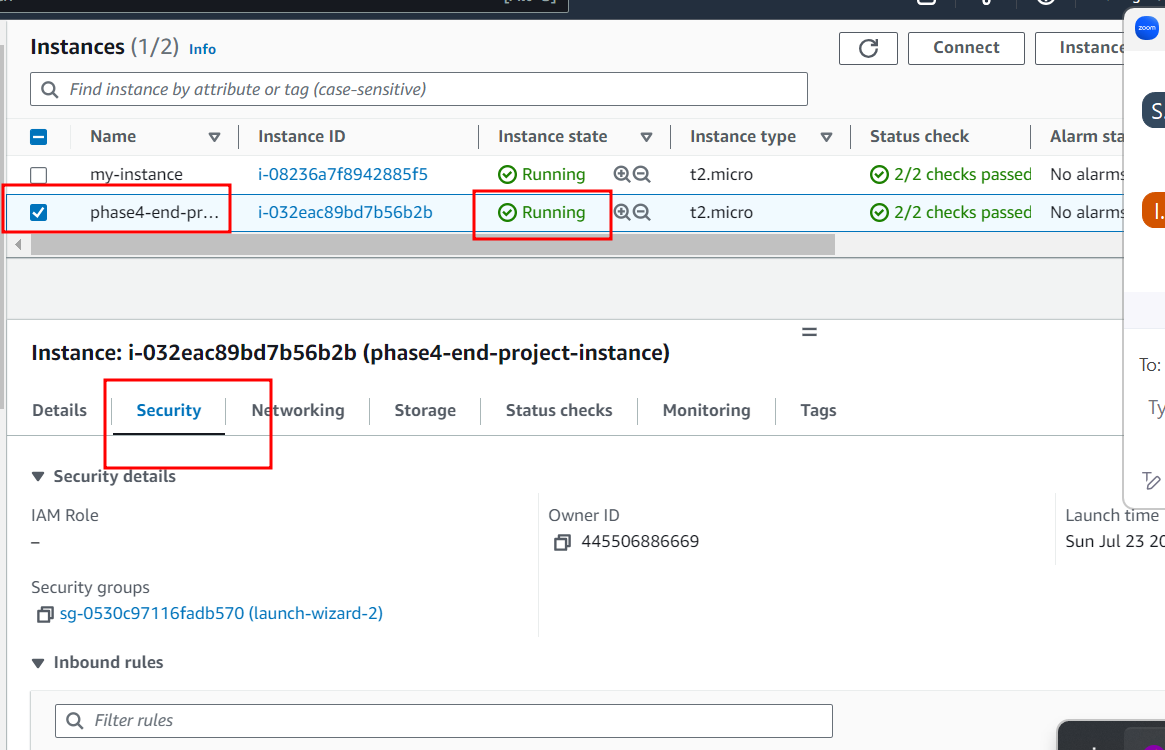


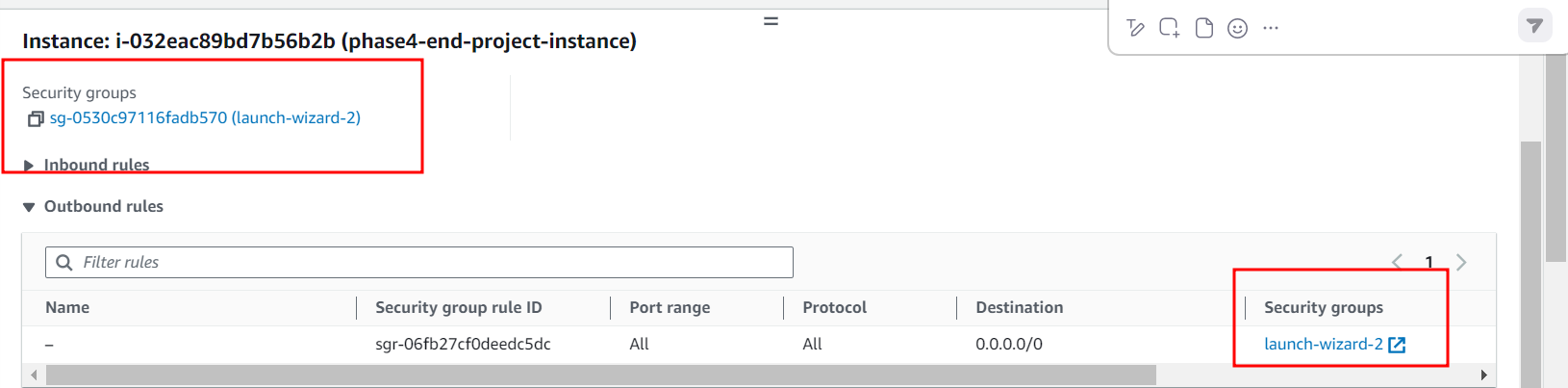
To open this application check ip address and port number (80)

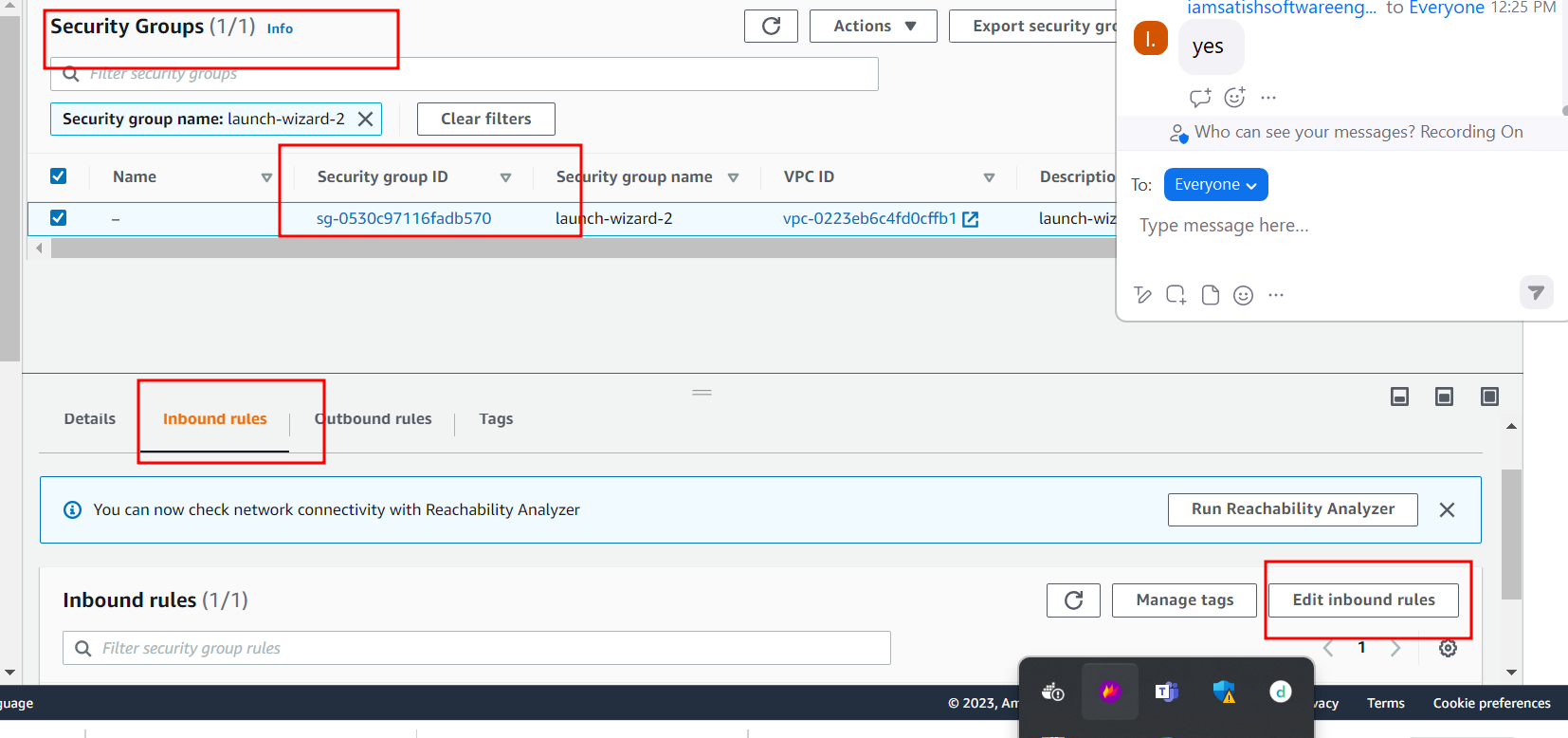
<http://18.234.191.44:80>



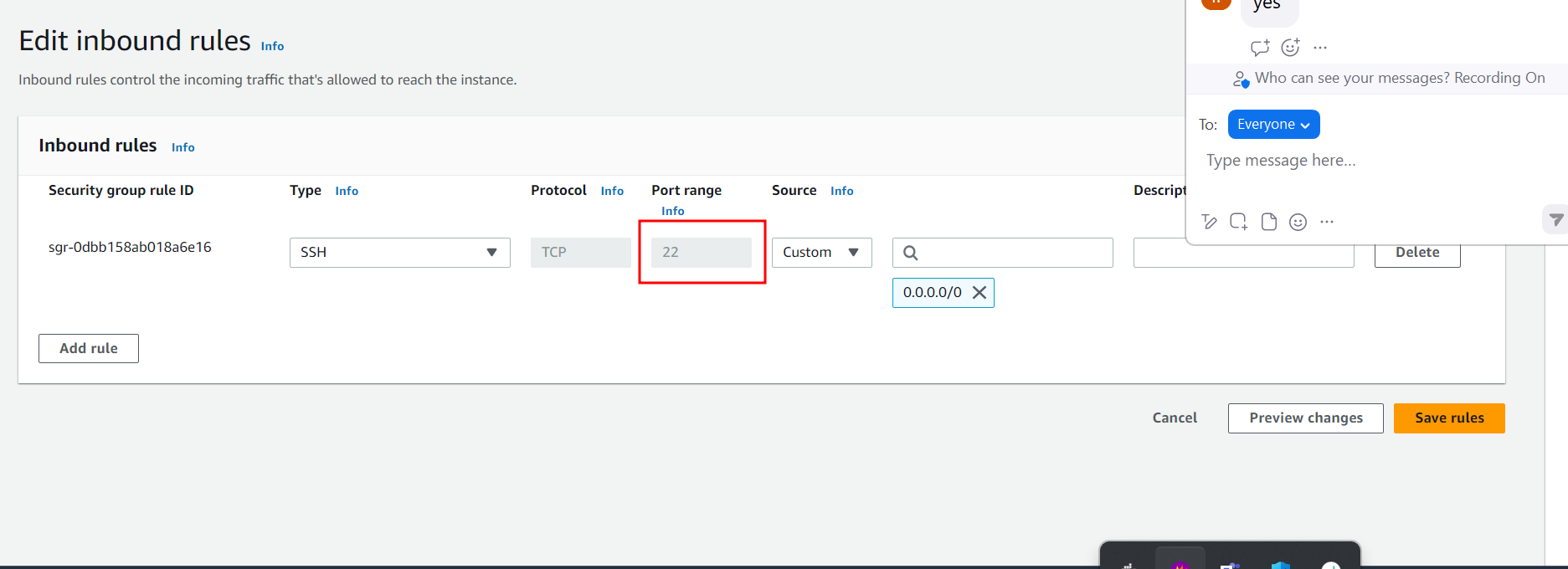
Still we can’t access we need to open 80 port number in EC2 instance for incoming call.

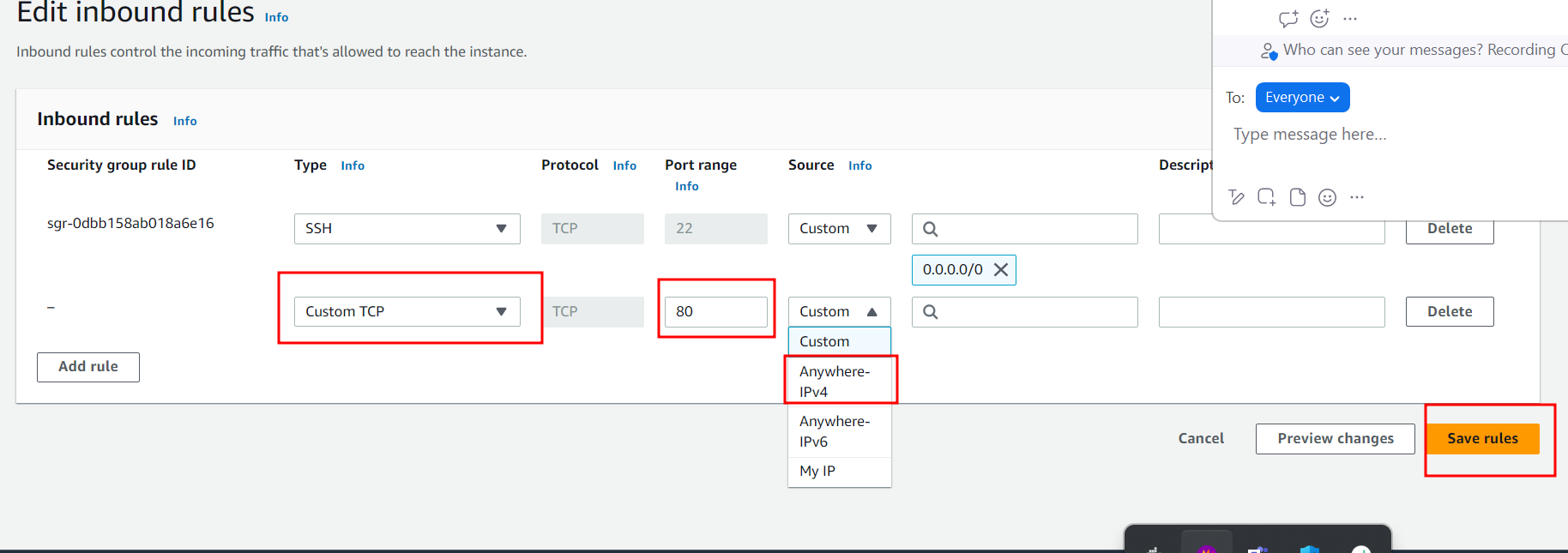


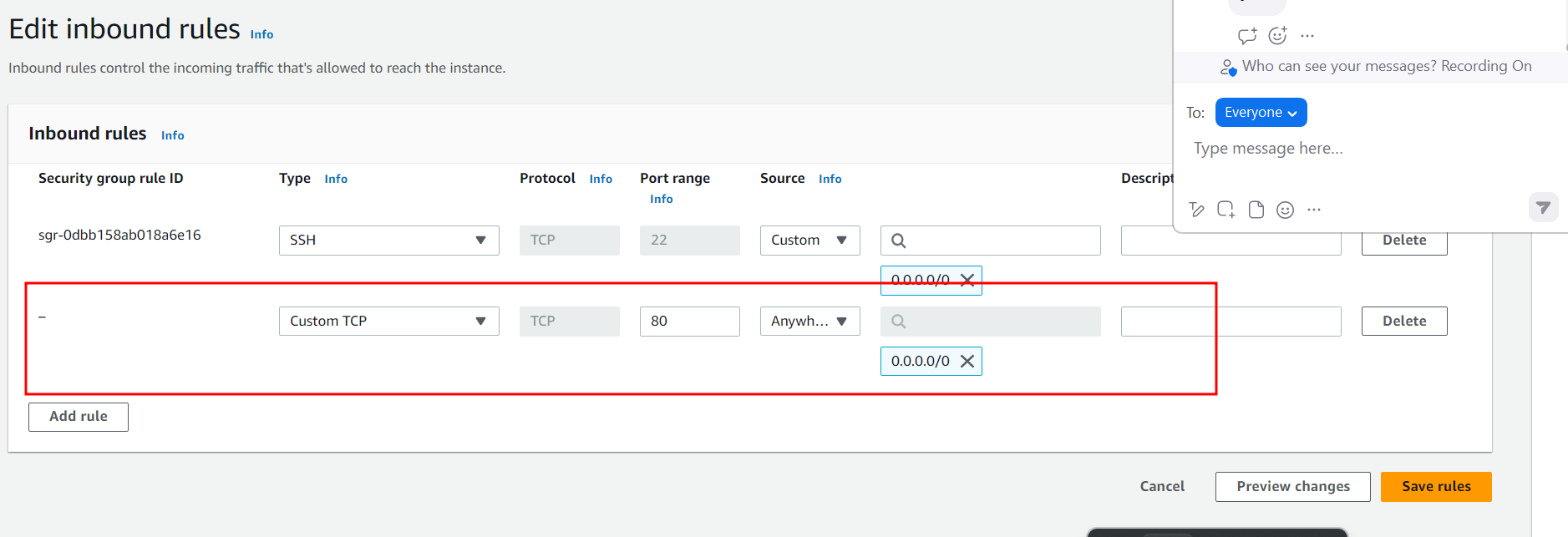




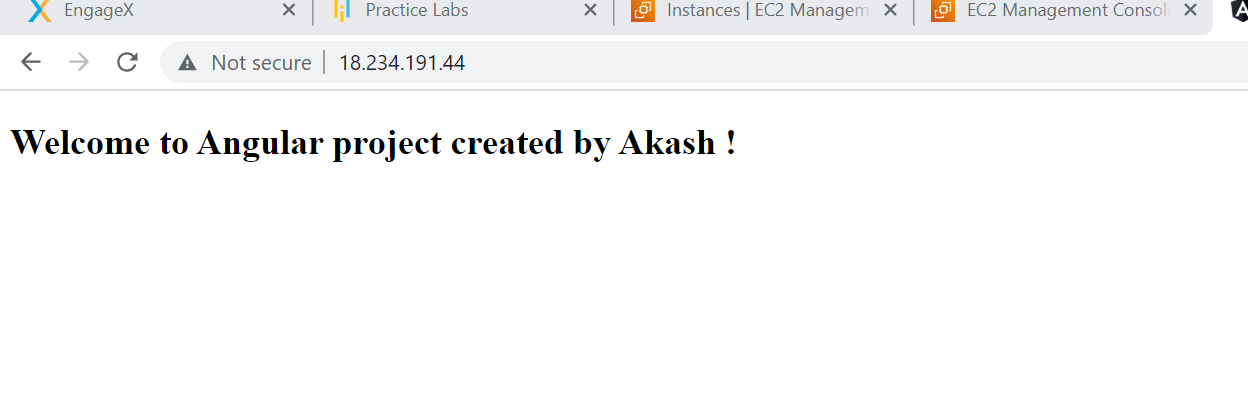
By default EC2 instance open 22 port number which help to connect ec2 instance through SSH.







Now you can access this application using your ip adderss with port number



Medicare

Backend Express js database mongo db and front end angular

Two types of users

Admin

SignIn for Admin but no Signup (account created when we run express js project).

Emailed is as [admin@gmail.com](mailto:admin@gmail.com) and password as admin@123

Admin Home Page

Admin add Medicine details

Remove medicine details

Update medicine details

View all medicine details

View all orders details

Customer

SignIn and SignUp

Customer Home Page

View all medicine and

Place the order

Can view its own orders details.

Customer can view its own account details can add amount.

Database :

Collection – login \_id ,emalid, password and typeofuser (table)

Medicine :

\_id, mname, price, qty, imageUrl etc

Orders

\_id, mid, date, qty, price,emailid etc

Account :

\_id, amount,emailed -🡪 once you create user at the beginning we need create the account with initial amount as 1000.

Creating REST API using express js with mongoose module with mongo db module.

Create the folder as

Capstone project

Backend folder

**npm init (it is use to create package.json file)**

please give some package name as com

then enter continuously.

It will create the packge.json file.

npm install express

npm install mongodb

Frontend folder

Grocers

Fashionista