Capstone Project

Food Delivery

Travel app

Medicine app

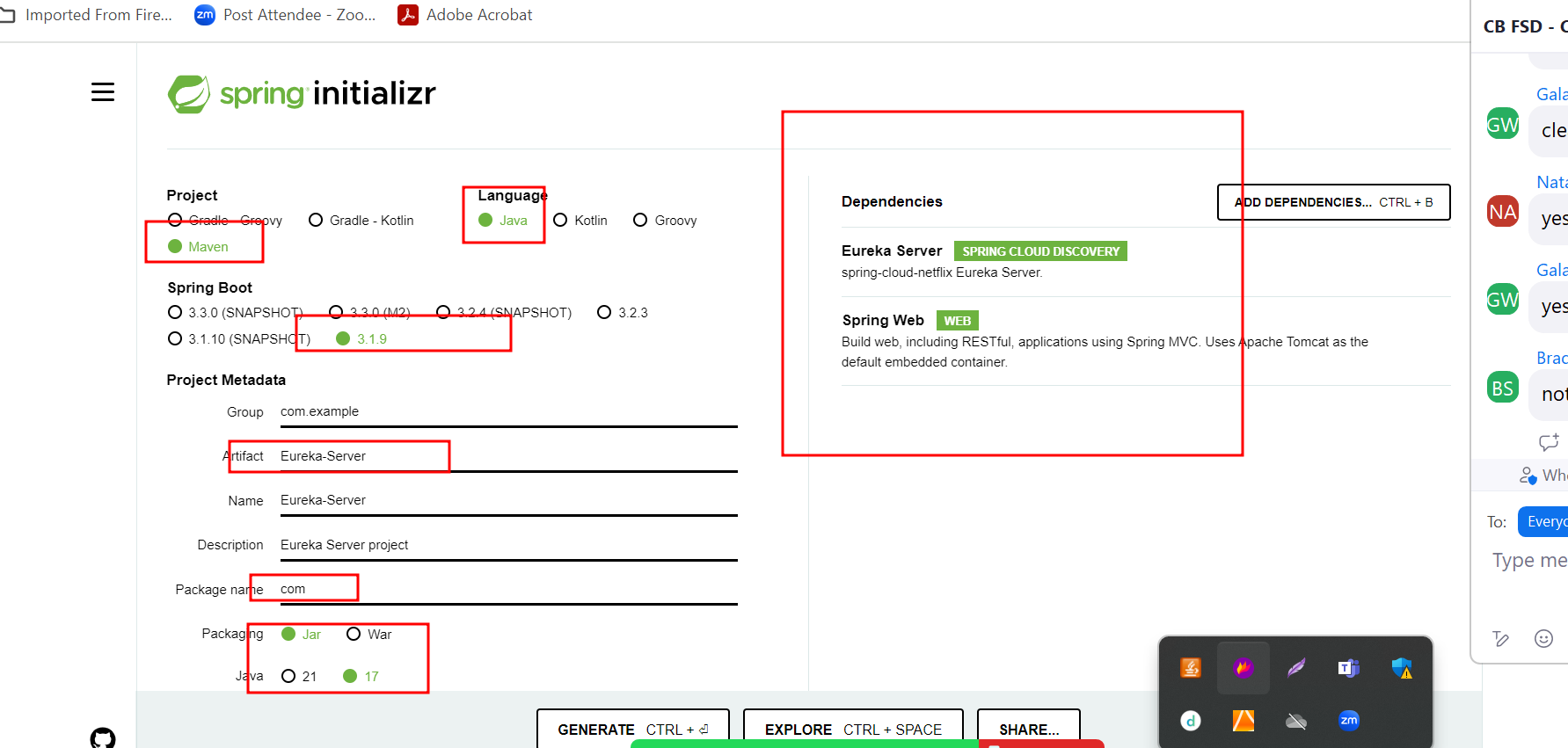
Create the folder with as Capstone project

backend

spring boot project

1. Eureka-Server-App 8761 port number

With starter as Eureka server and web starter

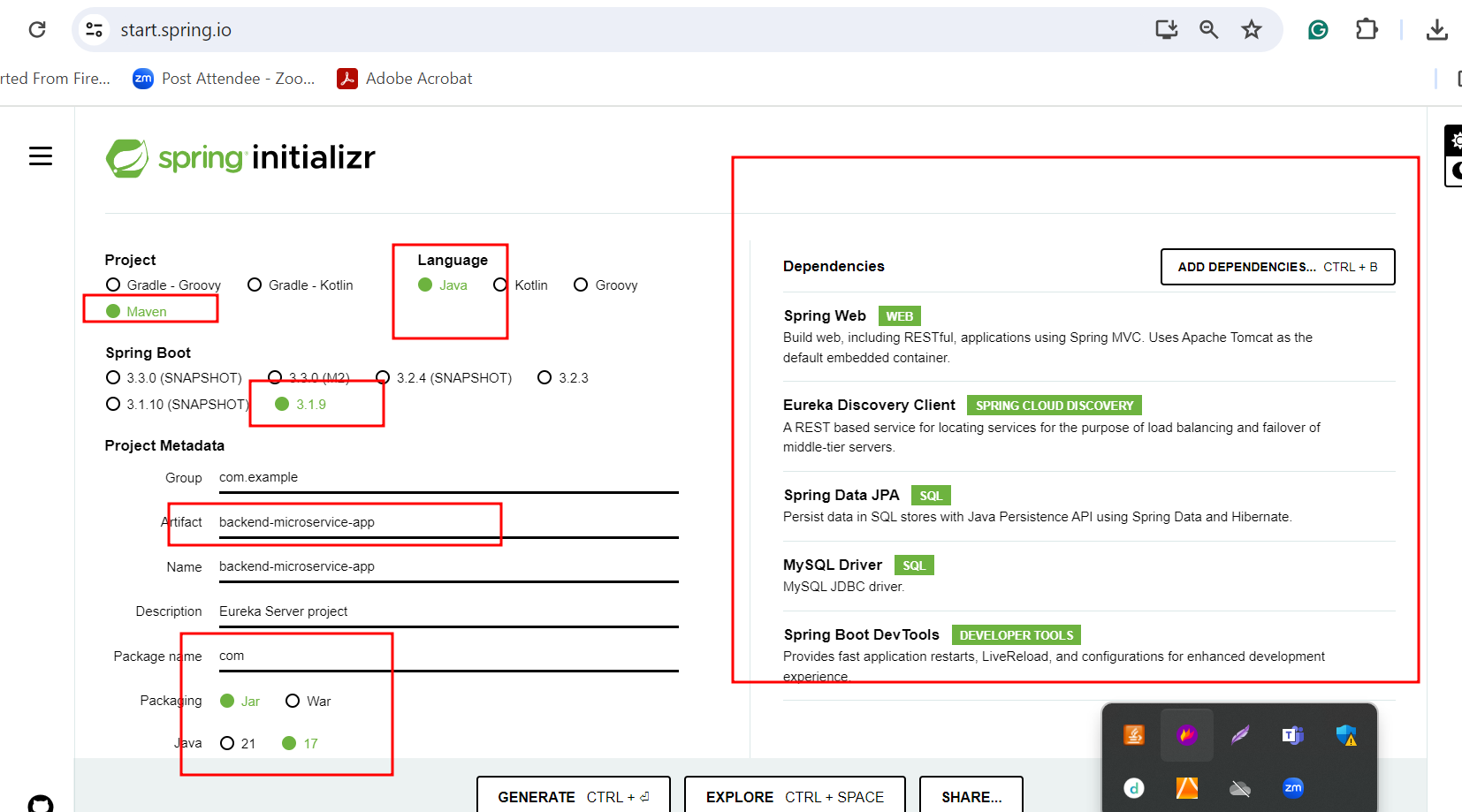


Dockerfile

1. Food-delivery-client, travel-app, medicine-app

9090 port number

Dockerfile



frontend

1. Food-delivery-frontend, travel-app-frontend, medicine-app-frontend

ng new frontend-app

routing 🡪 yes

styling 🡪 css

ng g c login

ng g c signup (customer)

ng g c admindashboard

ng g c customerdashboard

ng g class login ( model login details)

ng g s login two functionality for signin and signup

Dockerfile

docker-compose

we need to provide four containers

1. Mysql container
2. Frontend container
3. Eureka container
4. Backend container

Jenkinfile

EC2 instance :

Run jenkin in EC2 instance with pipeline to build the project.

Food Management System

Please Do CRUD Operation

Restaurants entity

Rid 🡪 PK auto increment

Restaurant name

List<FoodItem> listOfFoodItems

FoodItems entity

Fid 🡪 PK auto increment

FoodName 🡪

Description

Image -🡪

Qty 🡪

Admin login

Allow to add food details

Custom login

View the food details and place the order.

Travel Management system

Flight entity class

Fid 🡪 PK

AirlineBrand 1 2

List<FlightDetails> listOfFlights indigo airindia

FlightDetails

FDId -🡪 PK 1

Source -🡪 A

Destination 🡪 B

Date -🡪 19 Mar

Fare --🡪 5600

Customer

Source Destination Date

FID AirlineBrand source Destination Date

Now you need to create the jar file for Eureka Server and using this jar file we need to create Dockerfile code.

If you want to change the name for jar file do the changes in pom.xml file and using run with maven install which help to create jar file.



Dockerfile for Eureka Server

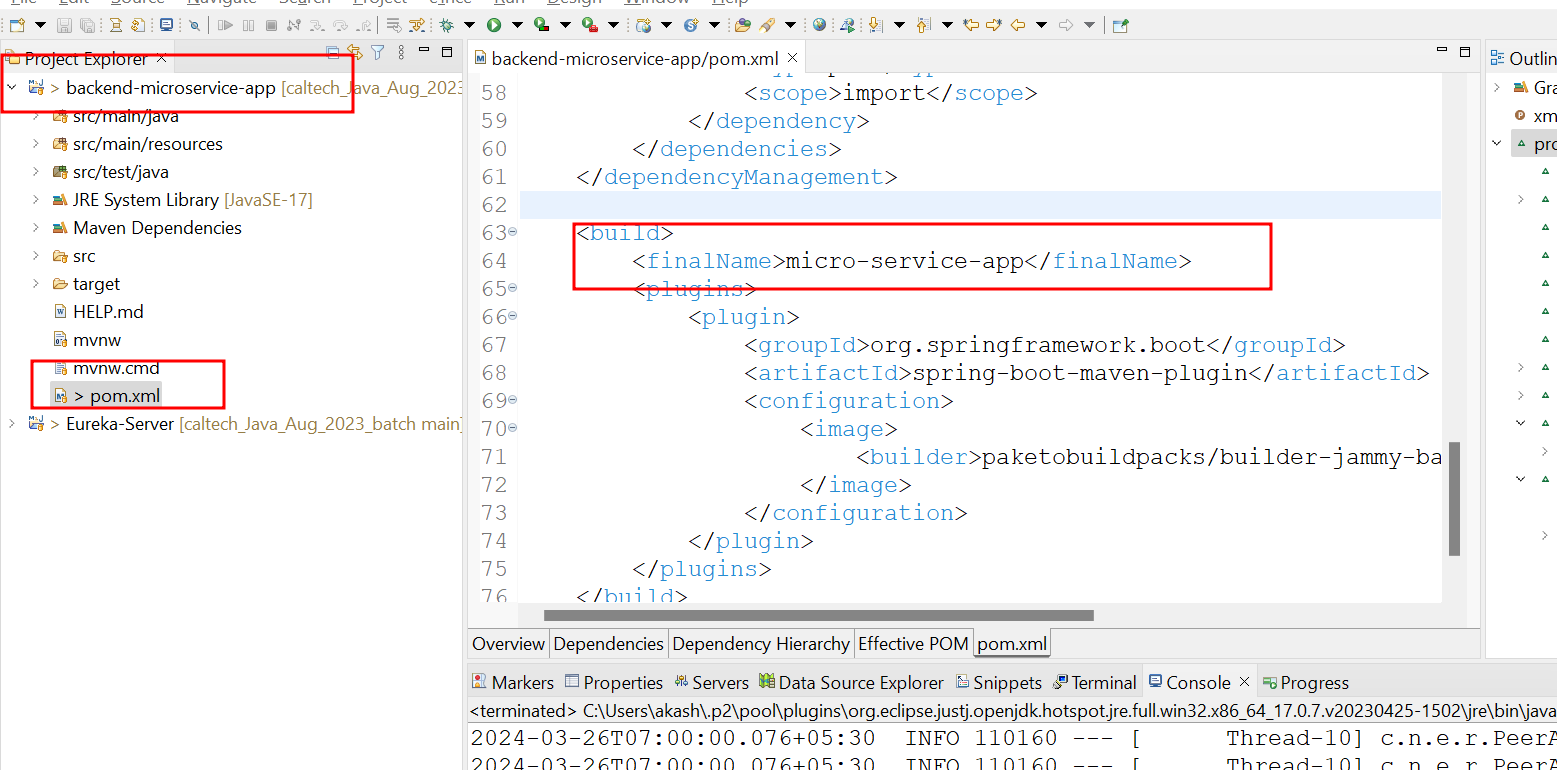
FROM openjdk:17

COPY ./target/eureka-server.jar .

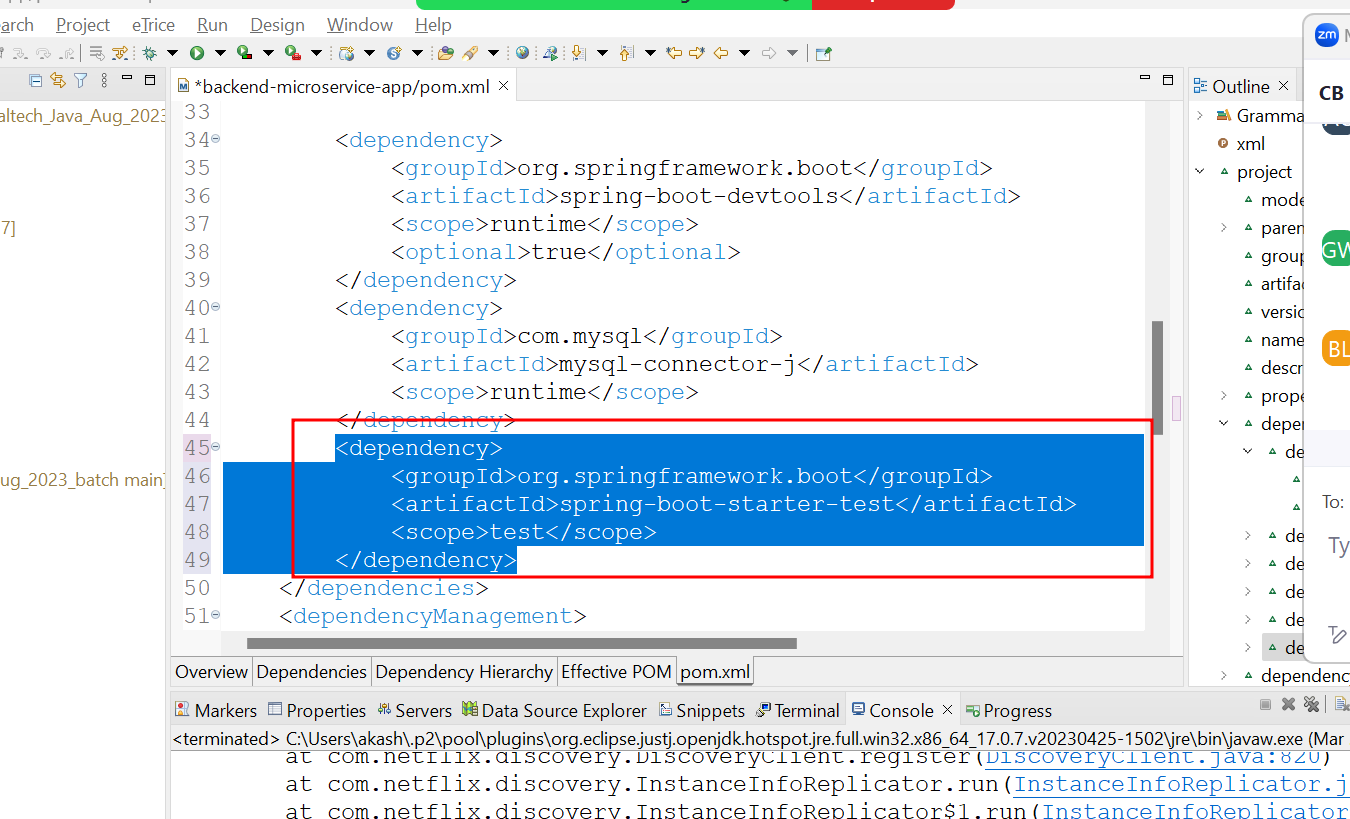
CMD ["java","-jar","eureka-server.jar"]

Now you need to create the jar file for microservice application and using this jar file we need to create Dockerfile code.

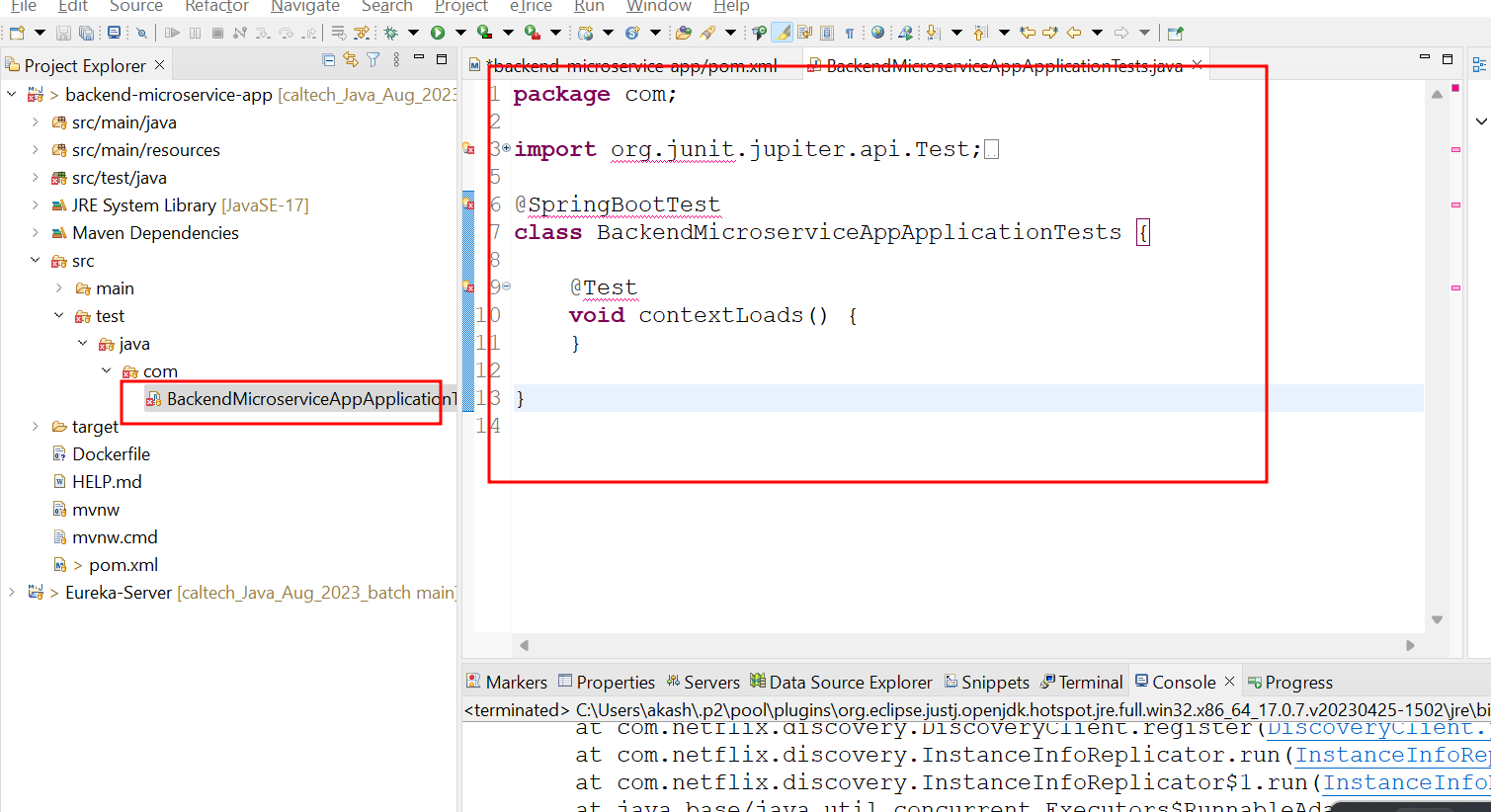
If you want to change the name for jar file do the changes in pom.xml file and using run with maven install which help to create jar file.



Please remove testing dependencies



As well as remove sample testing fille



Even we can remove dev tool dependencies

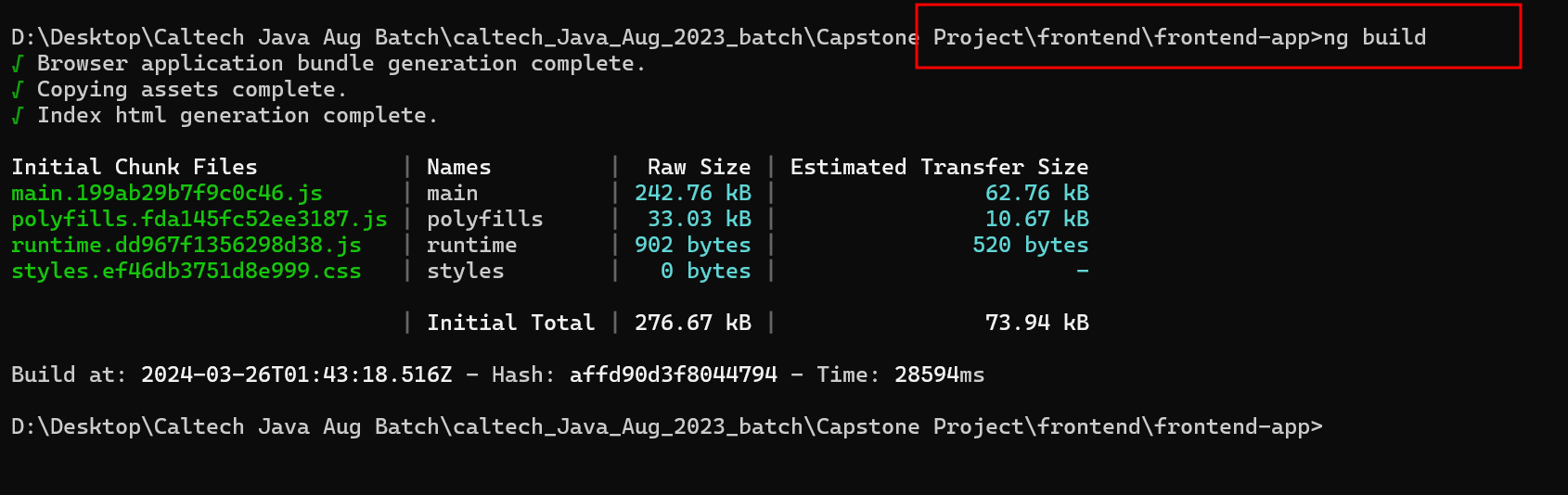
Dockerfile

FROM openjdk:17

COPY ./target/micro-service-app.jar .

CMD ["java","-jar","micro-service-app.jar"]

Now we need build angular project and create the Dockerfile



**Dockerfile**

FROM nginx

COPY ./dist/frontend-app/ /usr/share/nginx/html

We need to create docker-compose to run 4 containers

Mysql -🡪 database

Angular - container -🡪 frontend

Eureka – server -🡪 backend

Micro-server-app🡪backend

For mysql container

services:

  mysql-container:

    image: mysql:8

    container\_name: mysql-container

    environment:

      MYSQL\_ROOT\_PASSWORD: root

      MYSQL\_DATABASE: capstondb

    ports:

      - 3307:3306

    restart: always

Please do the changes in application.properties file for the database.

spring.application.name=backend-microservice-app

server.port=9090

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.datasource.url=jdbc:mysql://mysql-container/capstondb

spring.datasource.username=root

spring.datasource.password=root

spring.jpa.hibernate.ddl-auto=update

After changes done in application.properites file please re-create jar file.

Service for eureka server and micro service application

  eureka-server:

    build: ./backend/Eureka-Server/Eureka-Server/

    ports:

      - "8761:8761"

    environment:

      - EUREKA\_CLIENT\_SERVICEURL\_DEFAULTZONE=http://eureka-server:8761/eureka/

  microservice1:

    build: ./backend/backend-microservice-app/backend-microservice-app/

    ports:

      - "9090:9090"

    environment:

      - EUREKA\_CLIENT\_SERVICEURL\_DEFAULTZONE=http://eureka-server:8761/eureka/

      - SPRING\_APPLICATION\_NAME=microservice1

    depends\_on:

      - mysql-container

      - eureka-server

    restart: always

docker-compose.yml

version: '3.8'

services:

  mysql-container:

    image: mysql:8

    container\_name: mysql-container

    environment:

      MYSQL\_ROOT\_PASSWORD: root

      MYSQL\_DATABASE: capstondb

    ports:

      - 3307:3306

    restart: always

  eureka-server:

    build: ./backend/Eureka-Server/Eureka-Server/

    ports:

      - "8761:8761"

    environment:

      - EUREKA\_CLIENT\_SERVICEURL\_DEFAULTZONE=http://eureka-server:8761/eureka/

  microservice1:

    build: ./backend/backend-microservice-app/backend-microservice-app/

    ports:

      - "9090:9090"

    environment:

      - EUREKA\_CLIENT\_SERVICEURL\_DEFAULTZONE=http://eureka-server:8761/eureka/

      - SPRING\_APPLICATION\_NAME=microservice1

    depends\_on:

      - mysql-container

      - eureka-server

    restart: always

  angular-container:

    build: ./frontend/frontend-app/

    container\_name: angular-container

    ports:

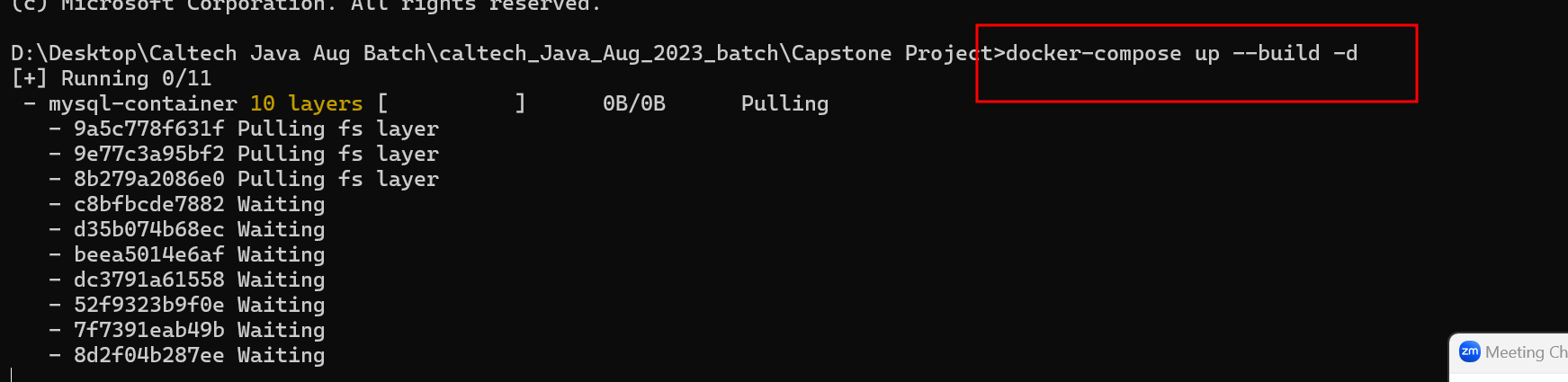
      - 80:80

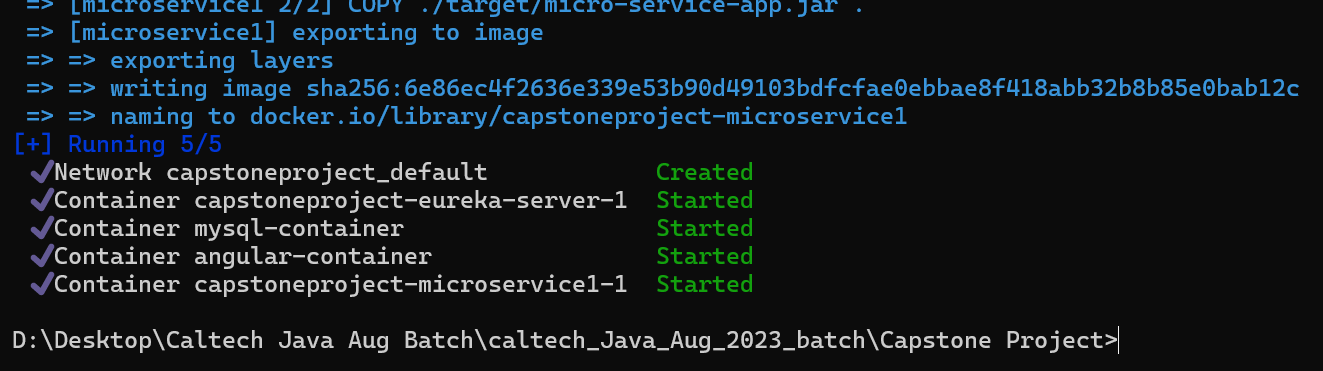
    restart: always

Please test you application using docker-compose in local machine.

Please open the command prompt in the in the place where docker-compose file present.

docker-compose up --build -d

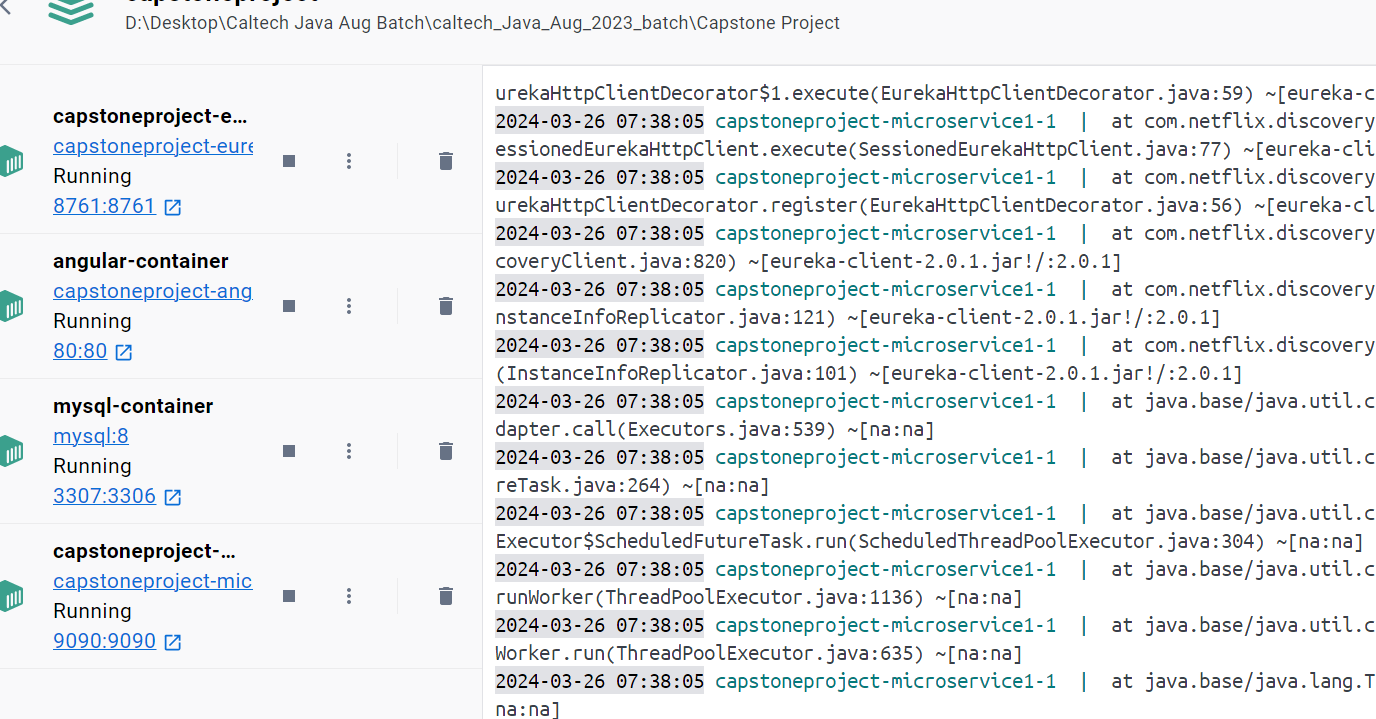




Docker ps

Or

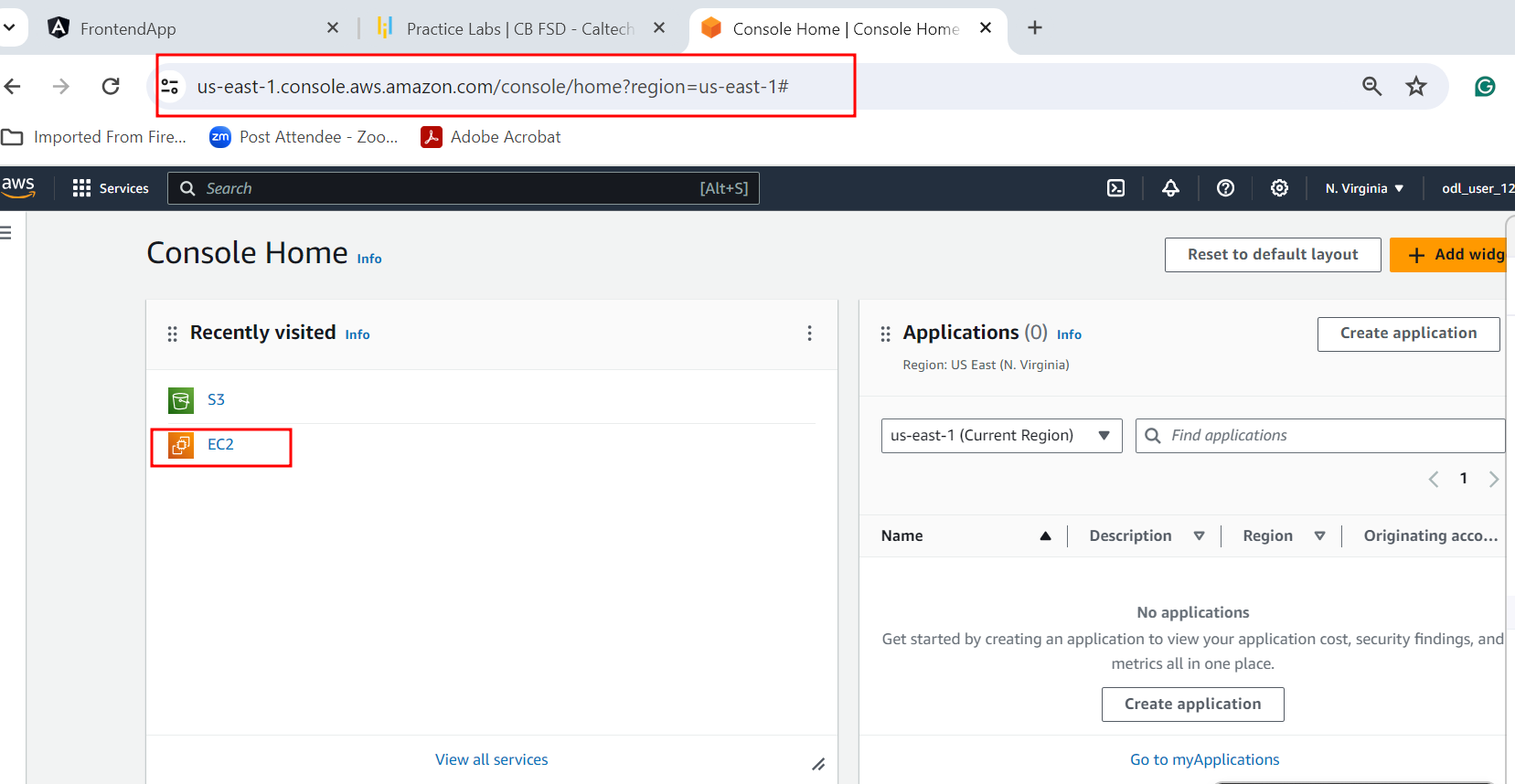
Docker desktop please check all container running or not.

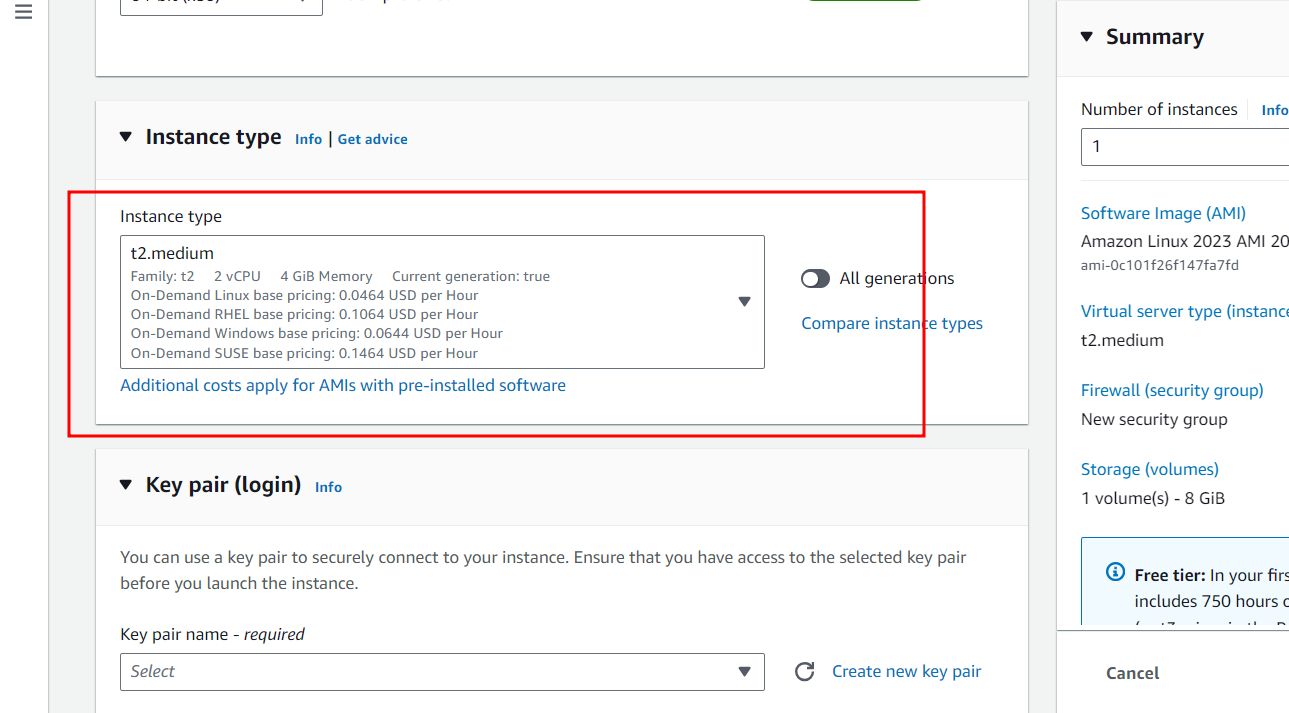


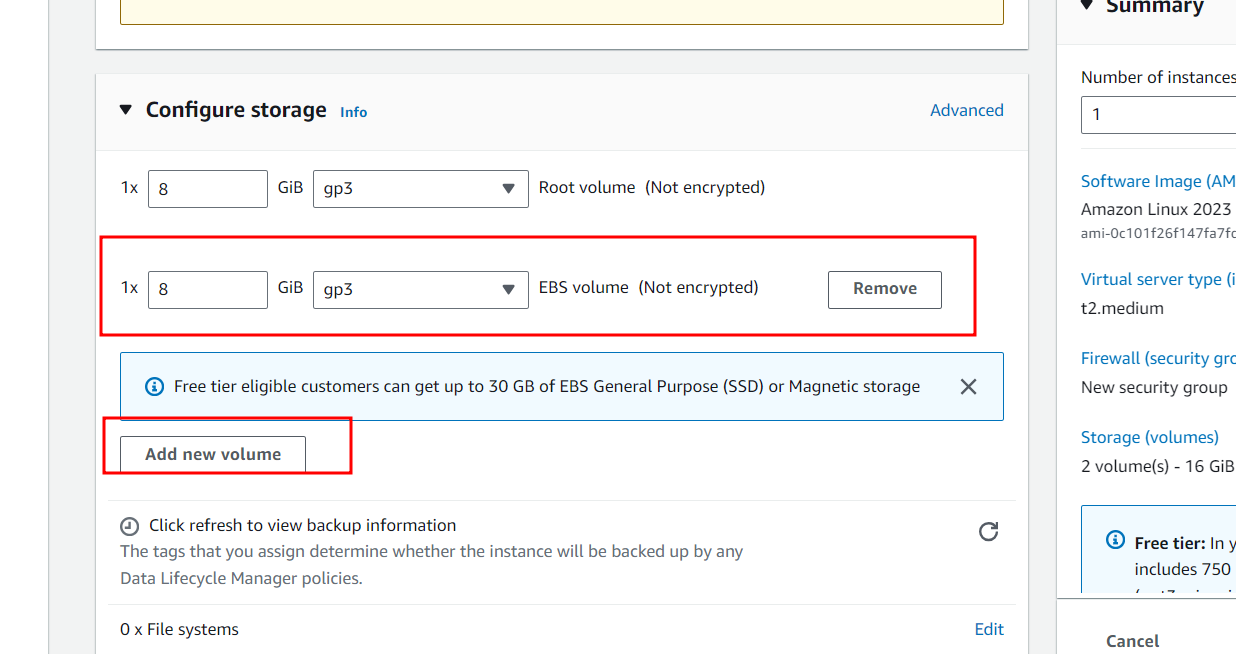
If run properly then open the browser and test the application

<http://localhost:80>

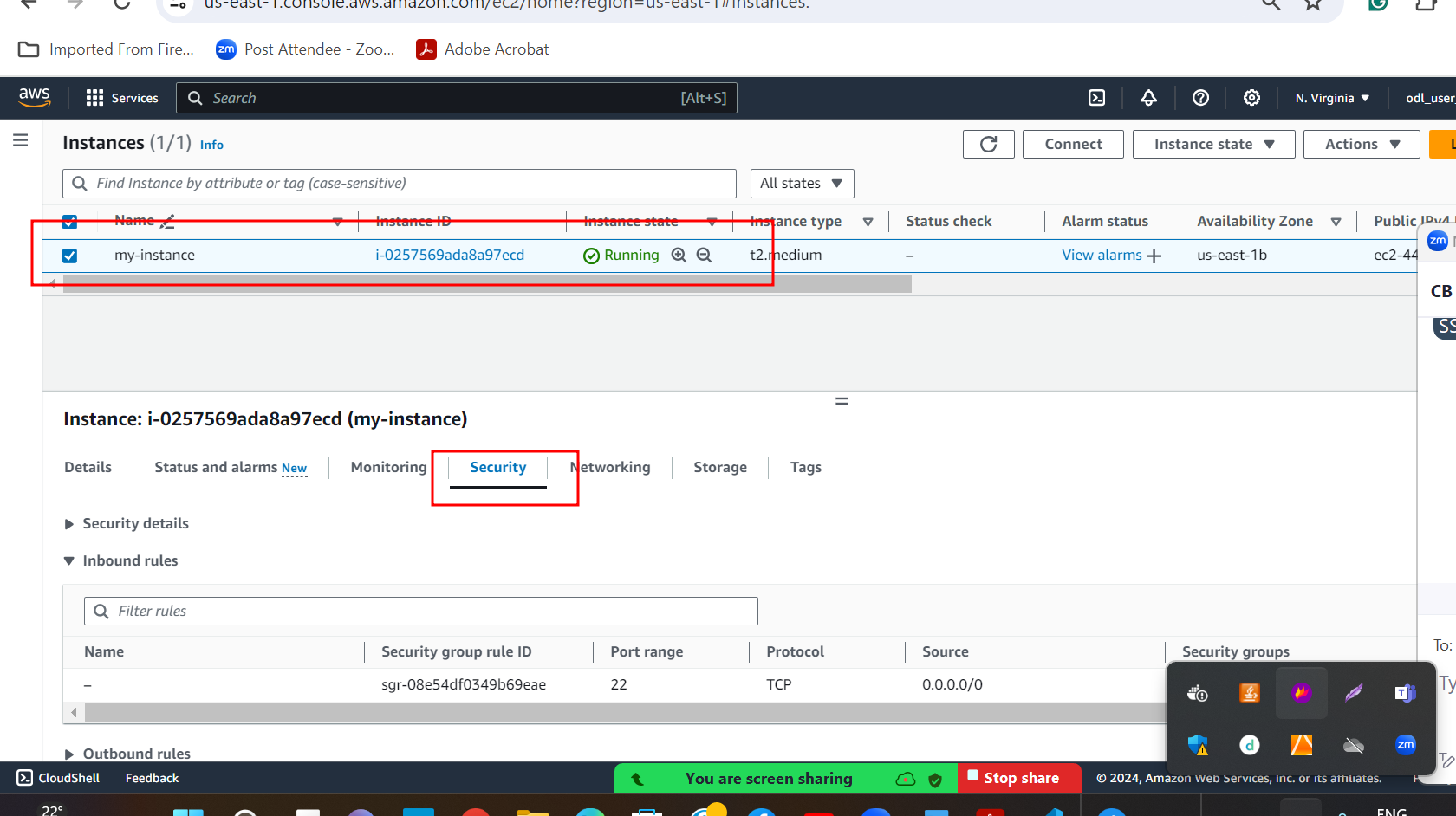
Now we need to login for AWS and create E2 instance.



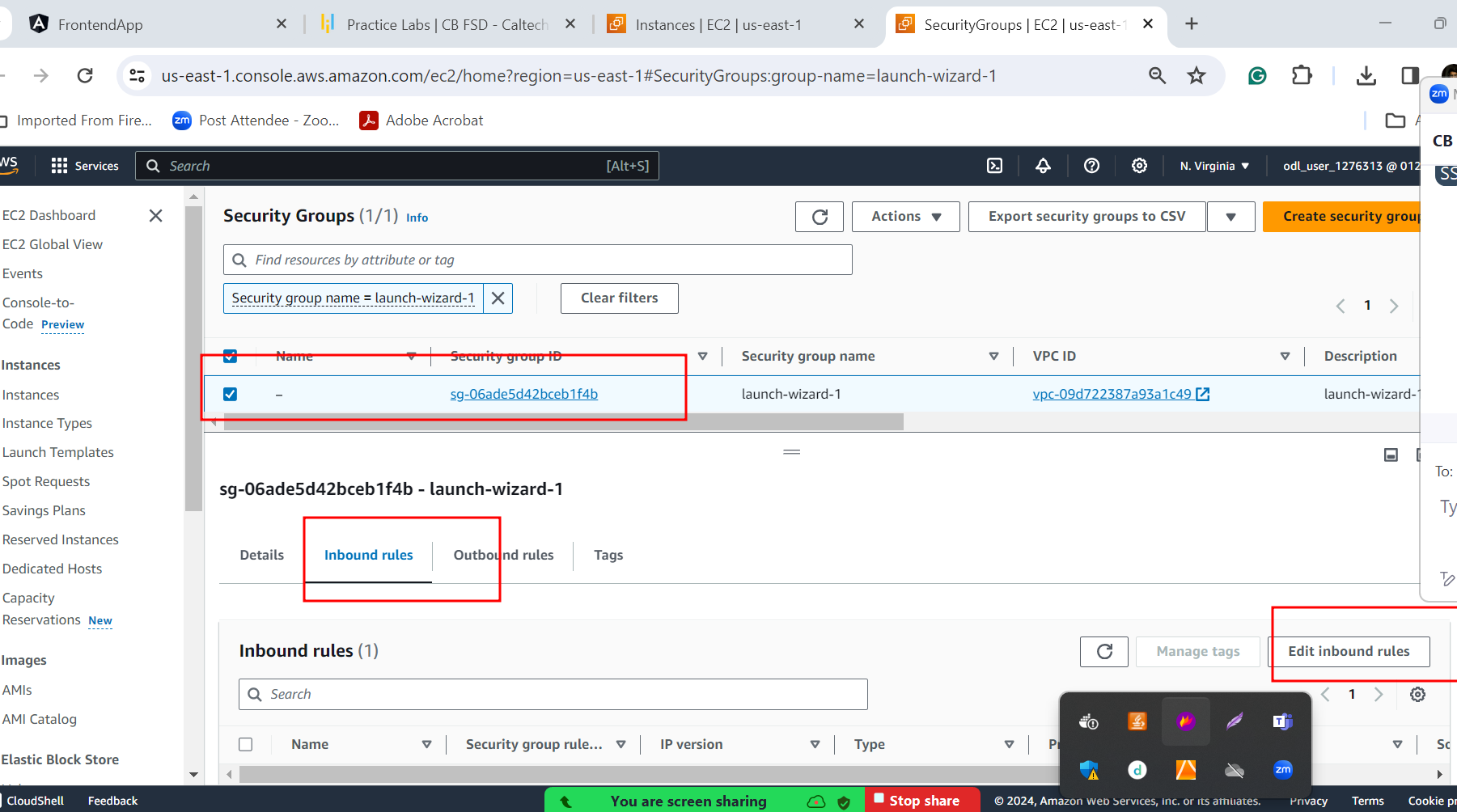


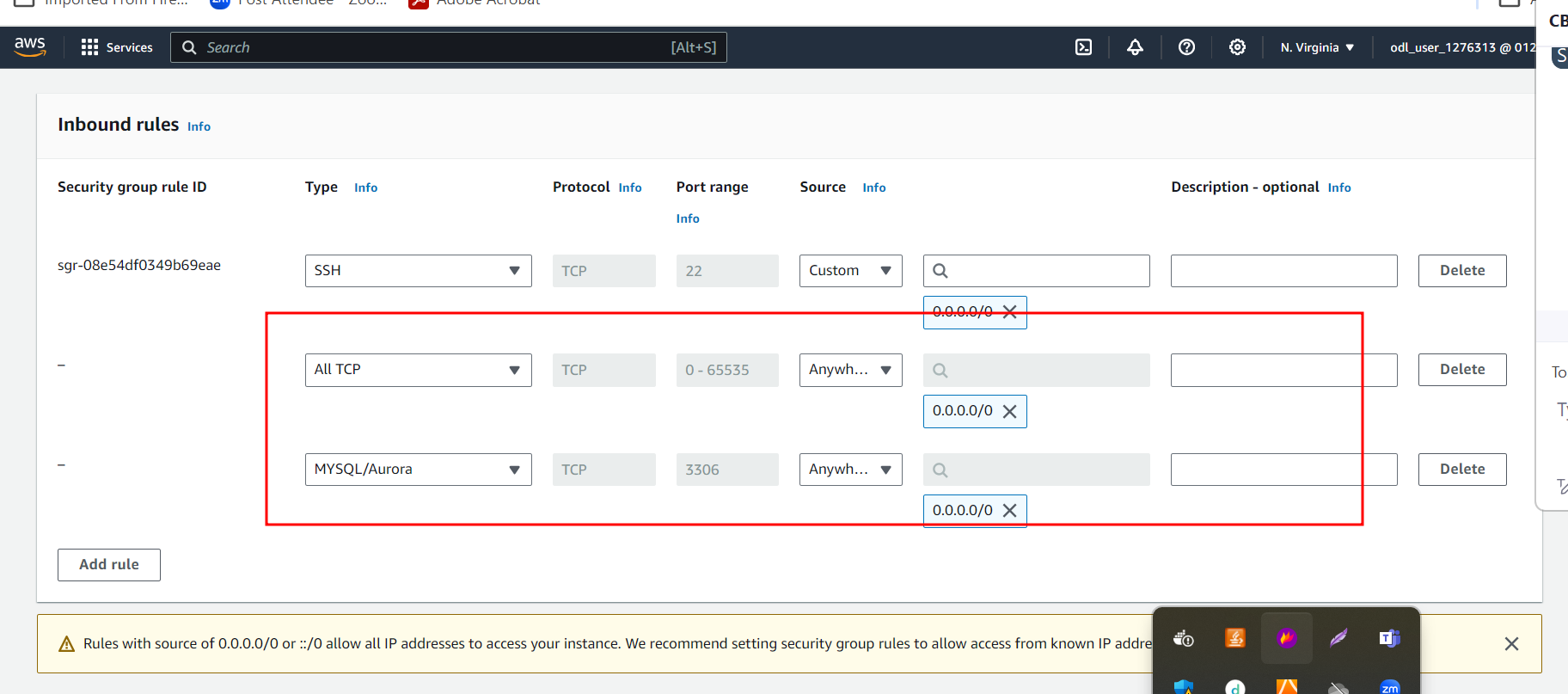


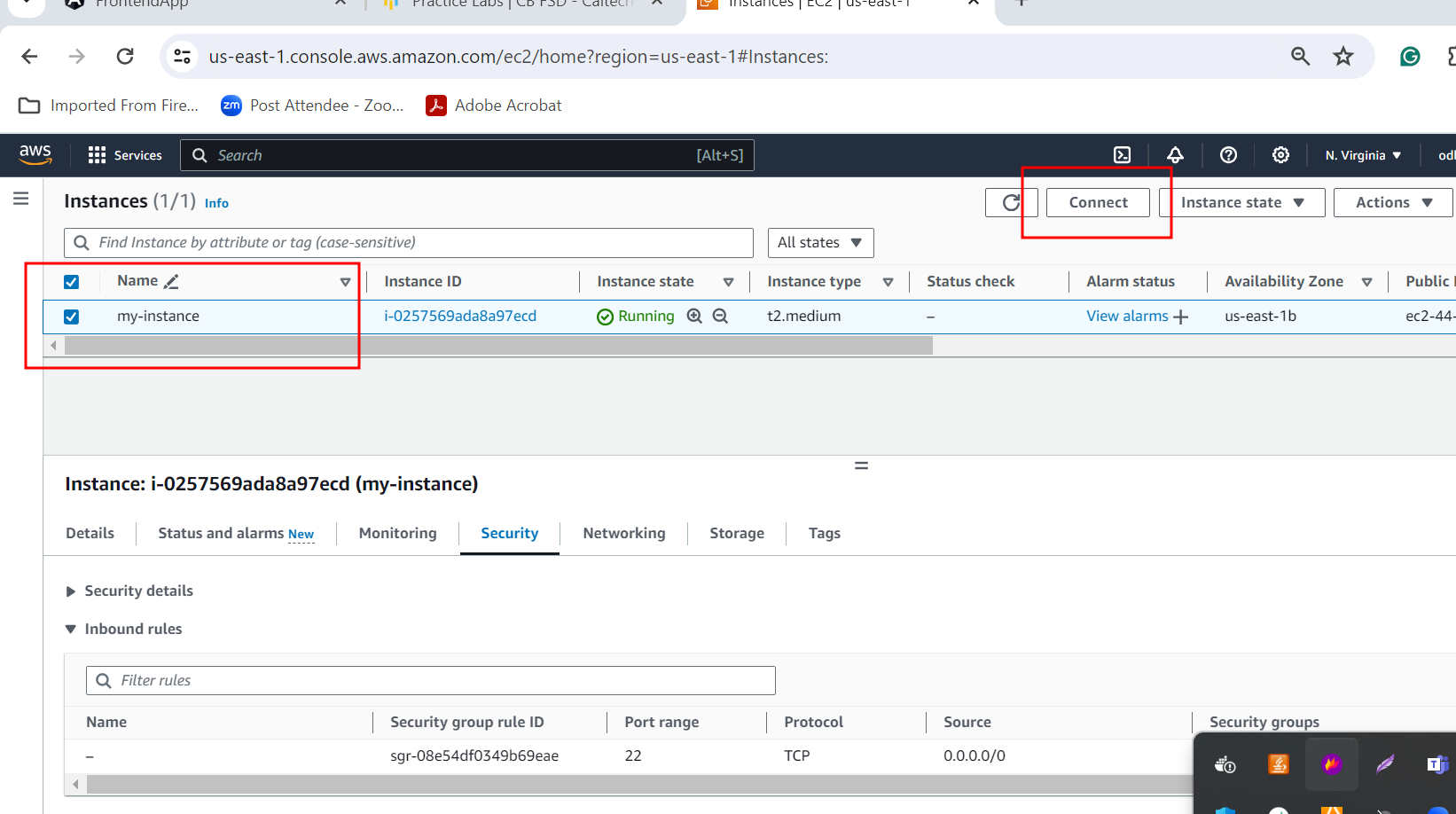
Once instance created. Please open port number in security group.

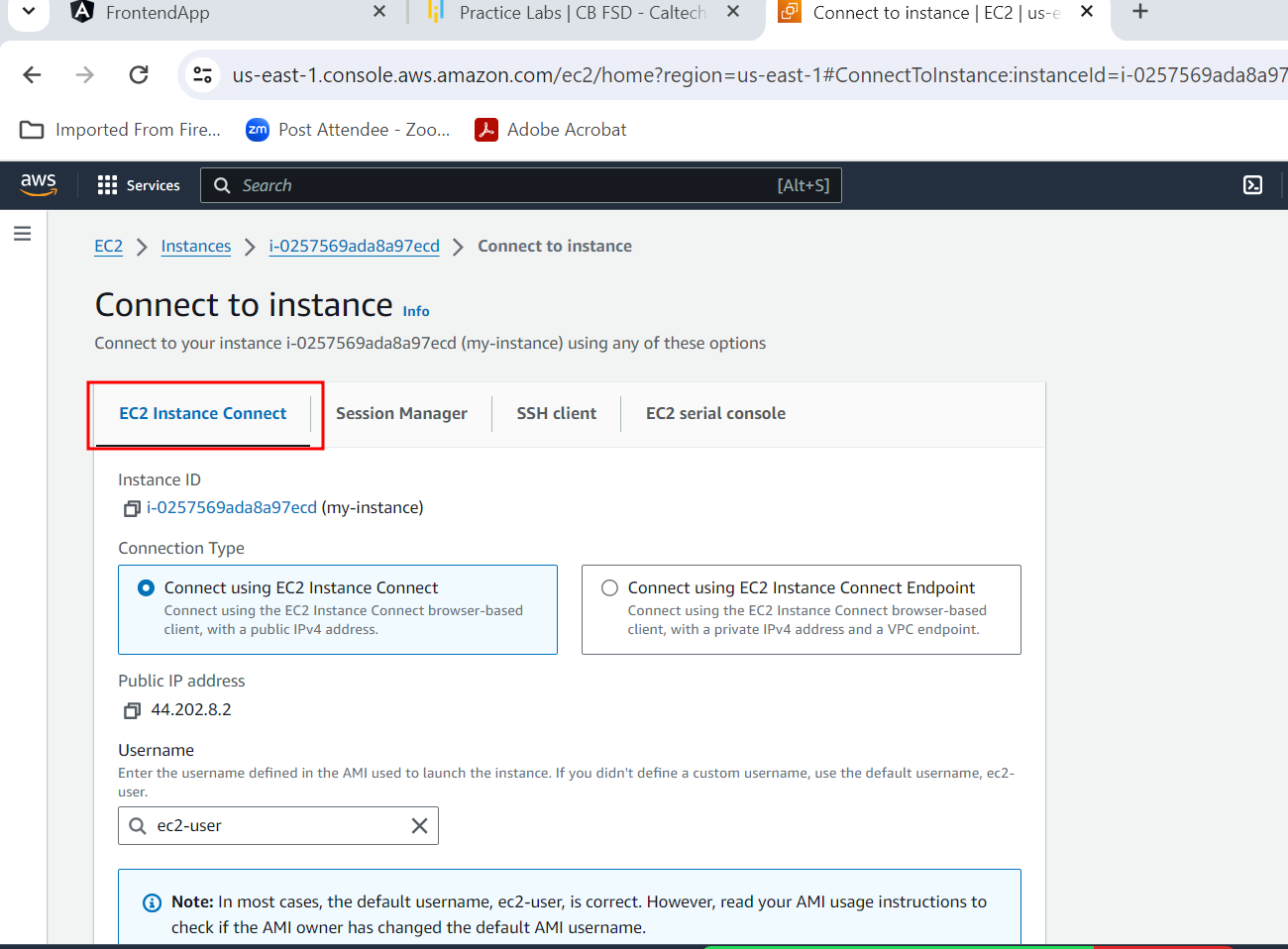


Click on launch wizard

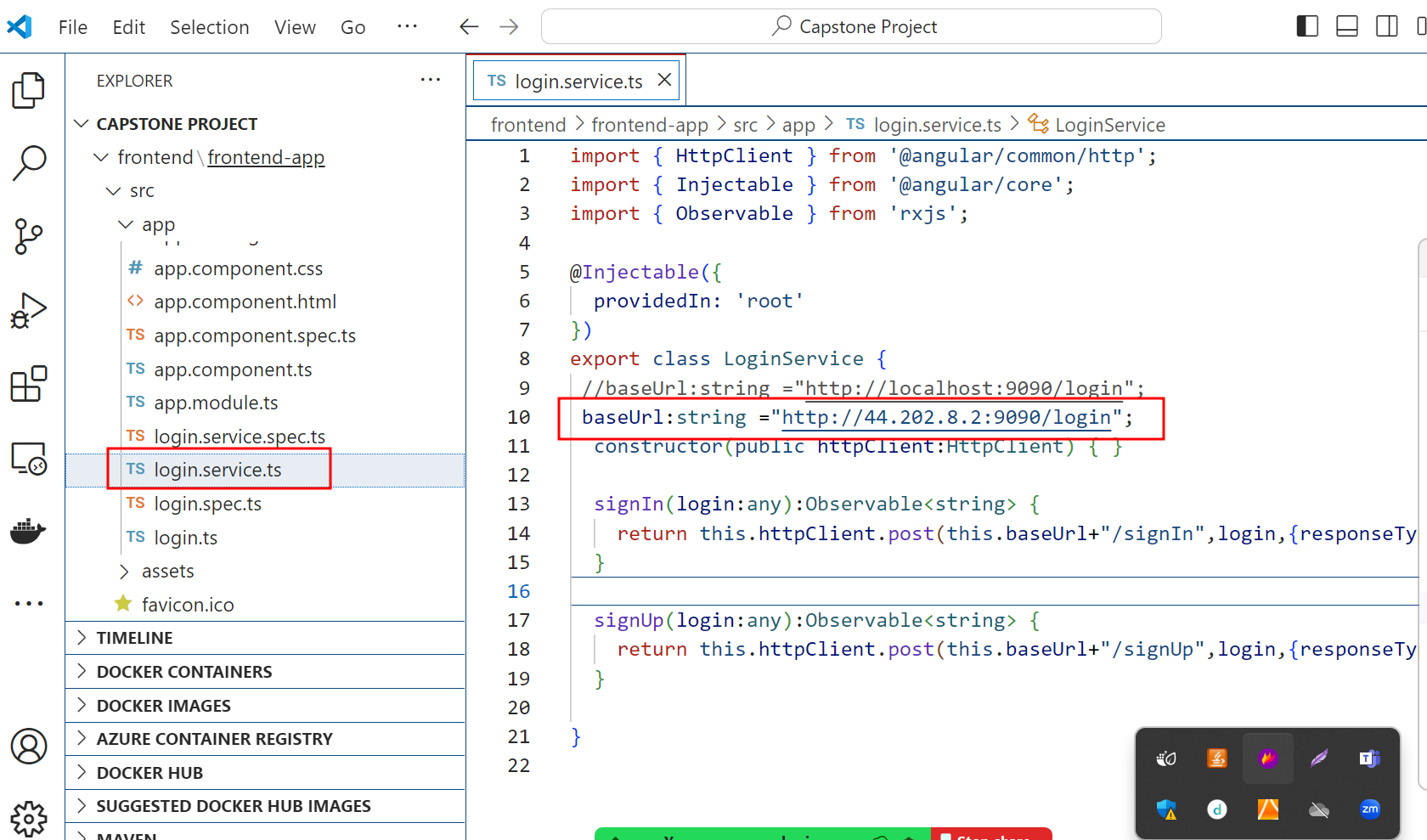


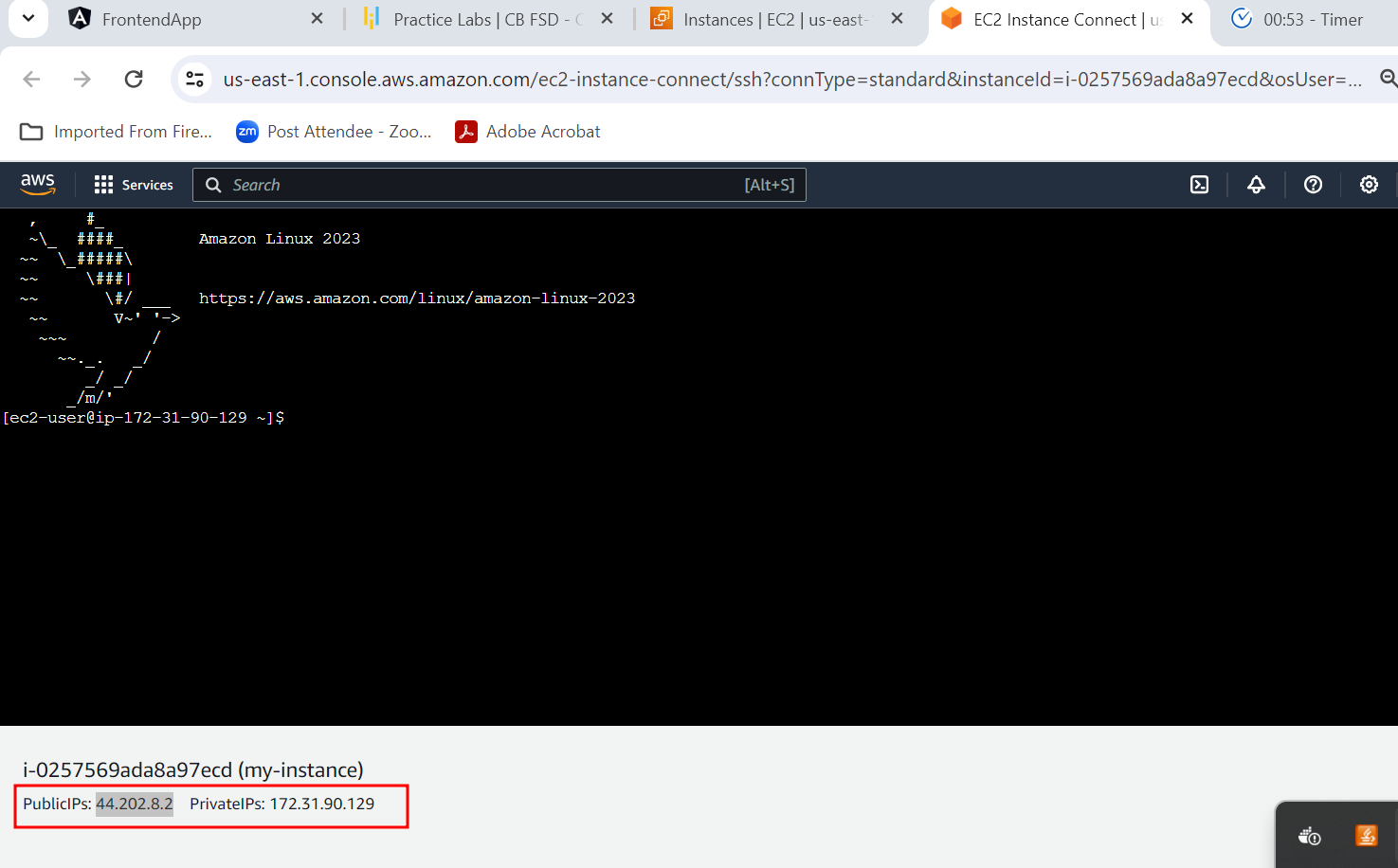






Do the changes in angular service file. Please change local host with EC2 instance IP Address.

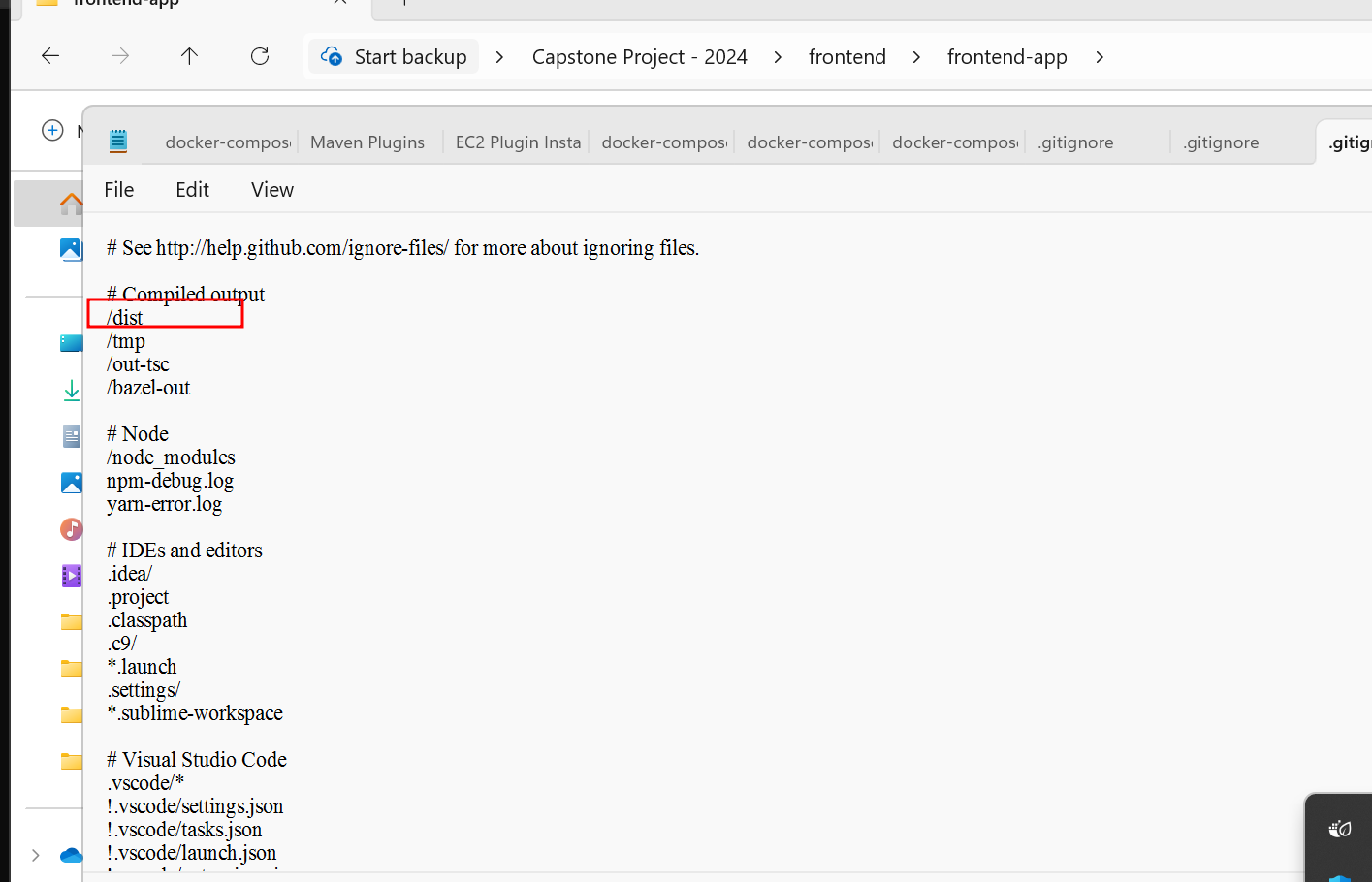




After changes done in frontend application. Re build this application once again.

Now you need to push this project which contains backend and frontend folder to github account.

By default git ignore dist and target folder. So please check in .gitingore file dist path as well as target path.



We need to push this project in remote repository

git init

git add .

git commit -m “initial commit”

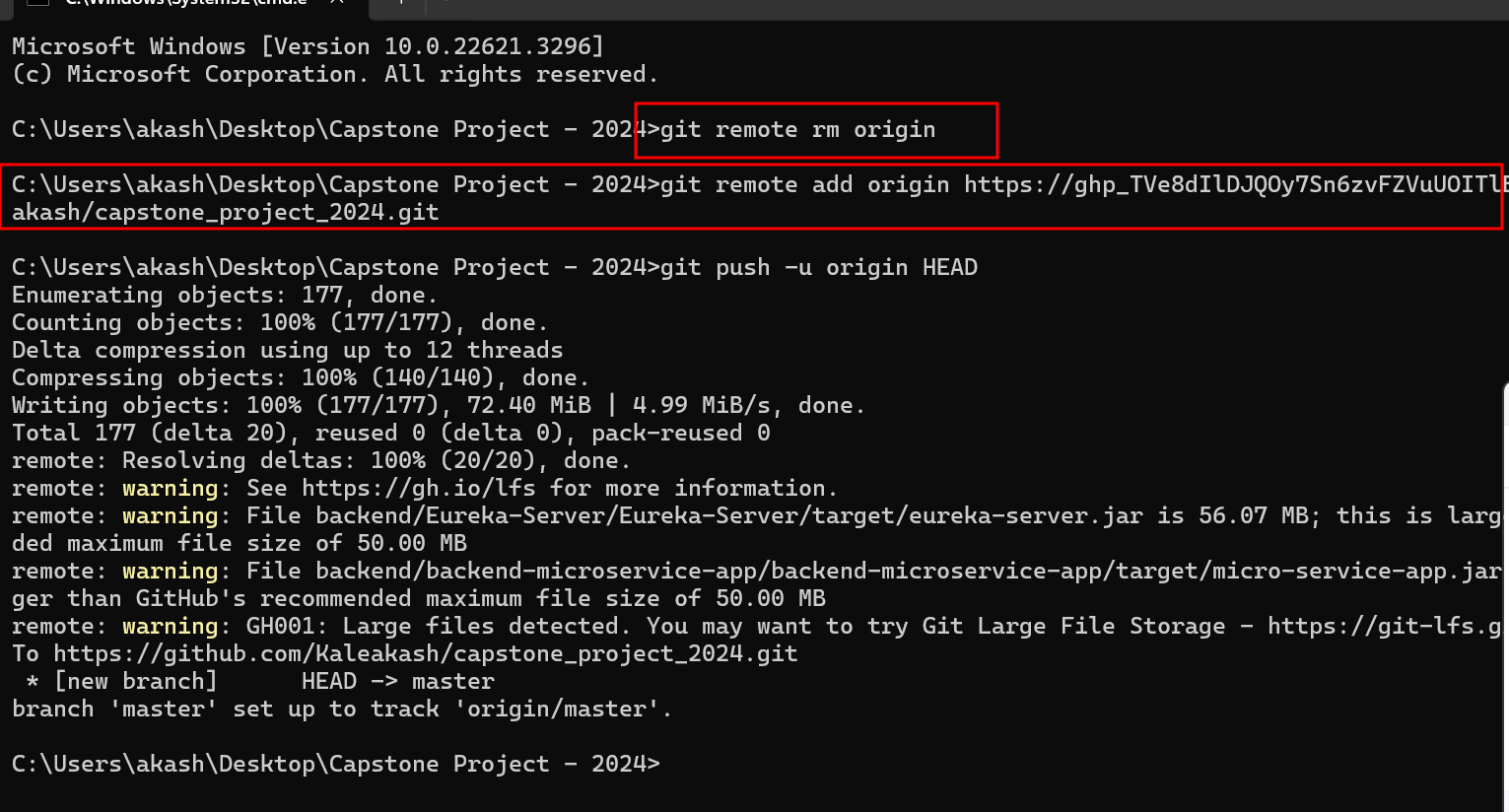
git remote add origin URLwithToken

git push -u origin HEAD

ghp\_TVe8dIlDJQOy7Sn6zvFZVuUOITlB7n1FnOM3

git remote add origin https:// ghp\_TVe8dIlDJQOy7Sn6zvFZVuUOITlB7n1FnOM3

@github.com/Kaleakash/capstone\_project\_2024.git



While pushing the code if any error generate please origin using below command as

git remote rm origin

<https://github.com/Kaleakash/capstone_project_2024.git>

Now we need install required software in EC2 instance

1 git

sudo yum install git -y

2 java

sudo yum install java

3 Jenkin

sudo wget -O /etc/yum.repos.d/jenkins.repo <https://pkg.jenkins.io/redhat/jenkins.repo>

sudo rpm --import <https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key>

sudo yum install jenkins

sudo service jenkins start

sudo cat /var/lib/jenkins/secrets/initialAdminPassword

4 docker

sudo yum install docker

sudo service docker start

5 docker compose

sudo curl -L https://github.com/docker/compose/releases/latest/download/docker-compose-$(uname -s)-$(uname -m) -o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose

we are planning to in docker and docker-compose in Jenkin environment

sudo groupadd docker

sudo usermod -a -G docker ec2-user

-----if you want to run docker and docker-compose in jenkin then please execute these command ---------------------

sudo usermod -a -G docker jenkins

sudo usermod -a -G docker ec2-user

sudo chmod 777 /var/run/docker.sock

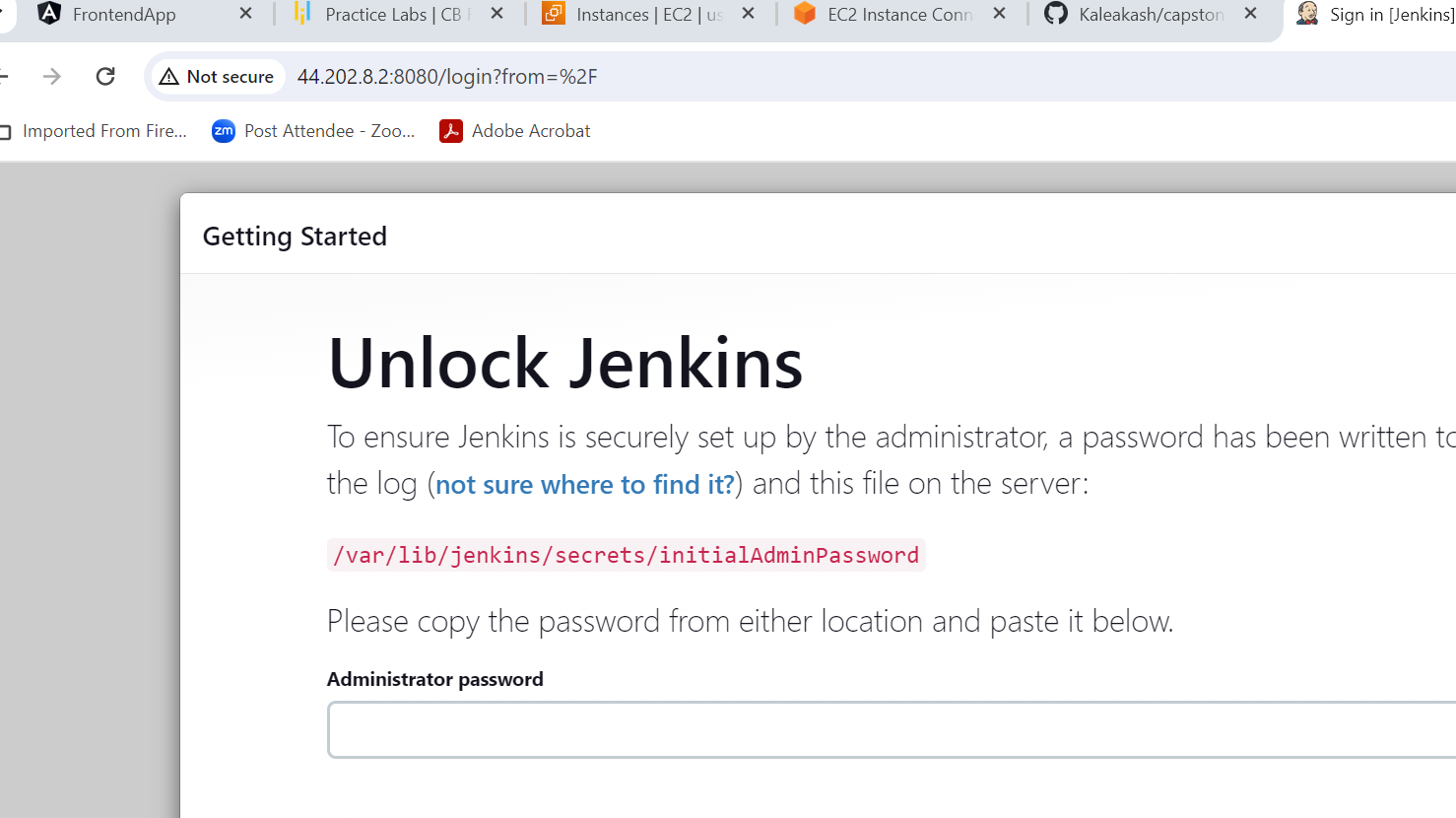
sudo service jenkins restart

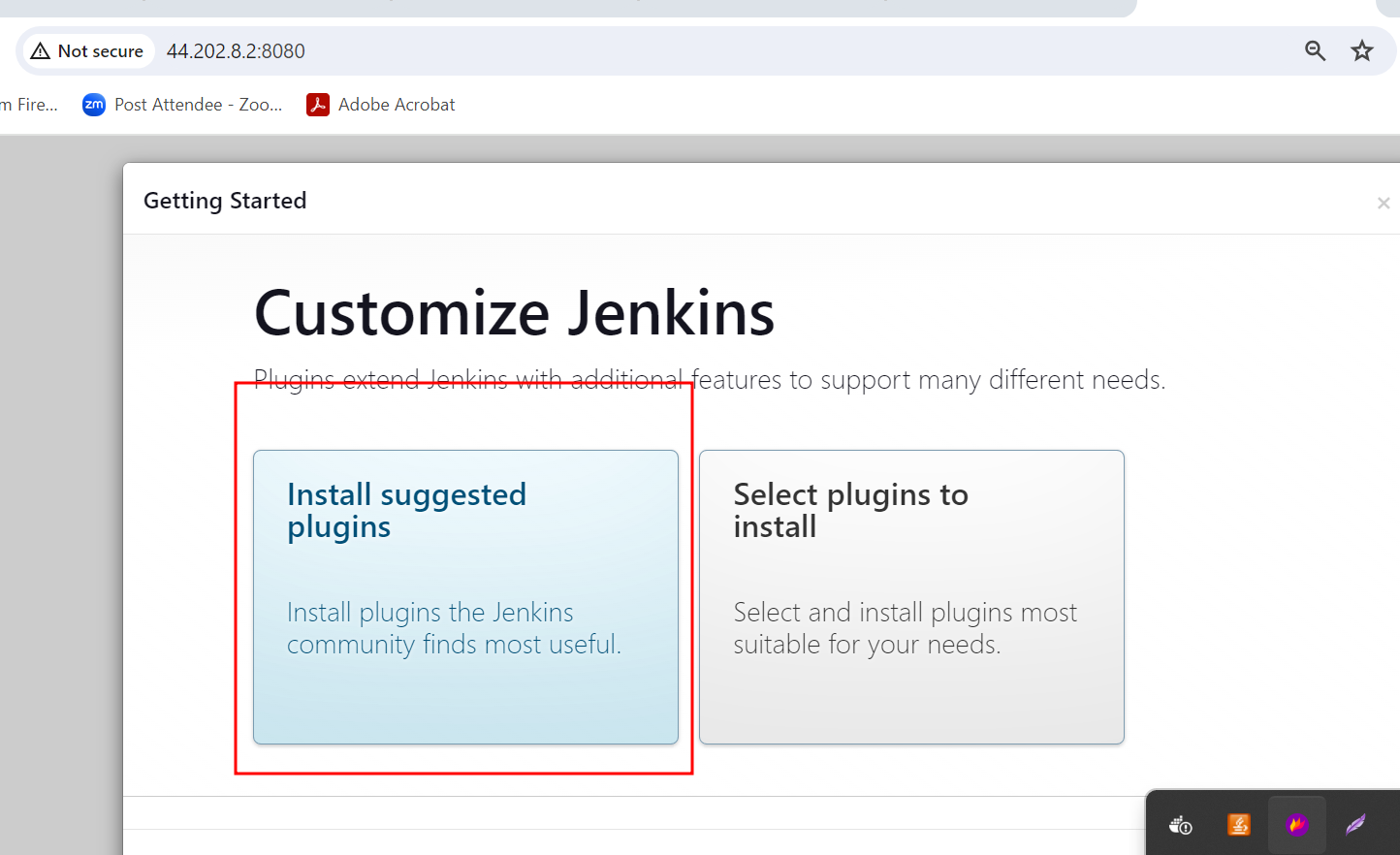
After install all required software please find Jenkin default password using below command

sudo cat /var/lib/jenkins/secrets/initialAdminPassword

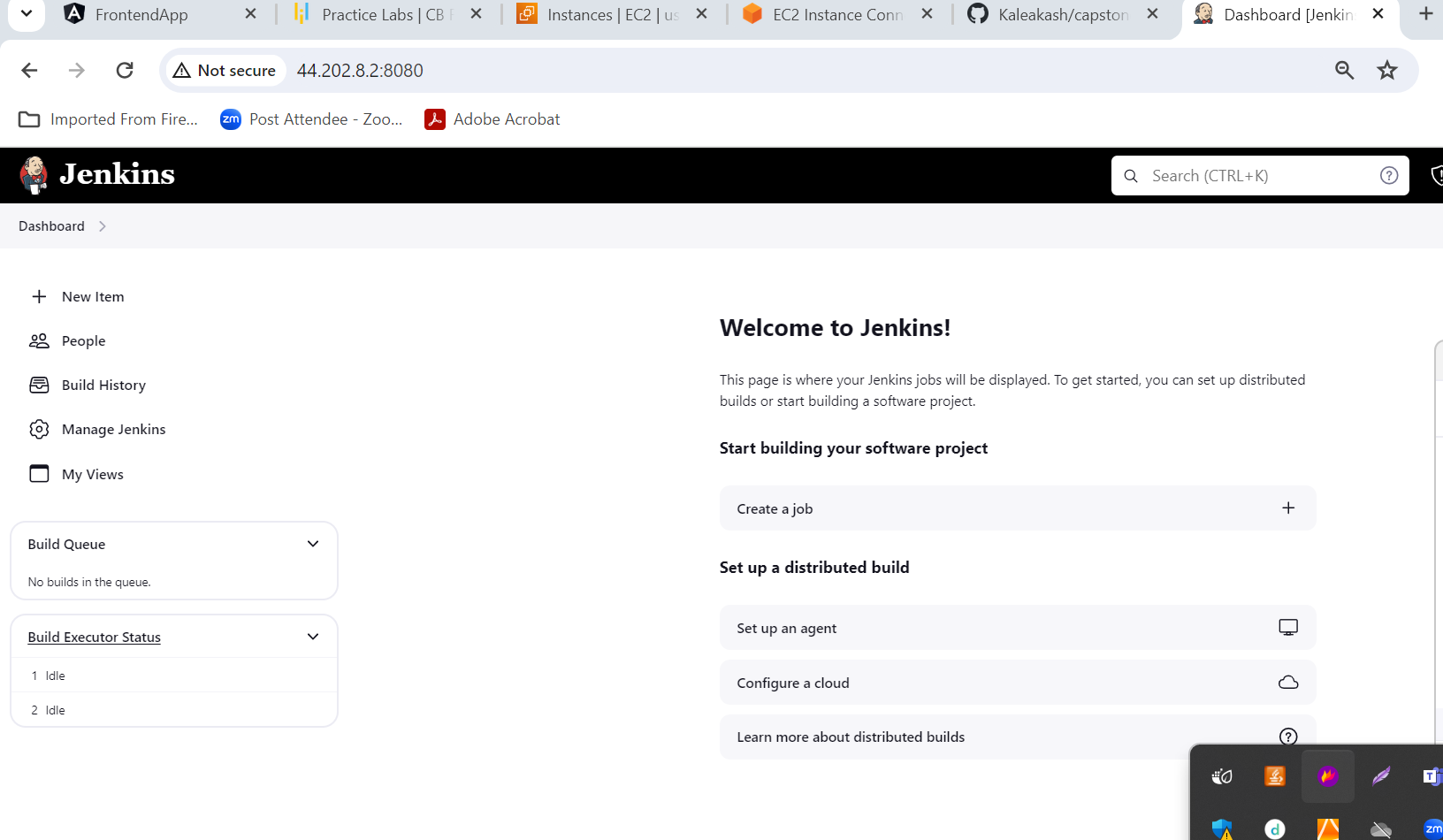
then open Jenkin dashboard using below URL.

<http://publicIdAddress:8080>



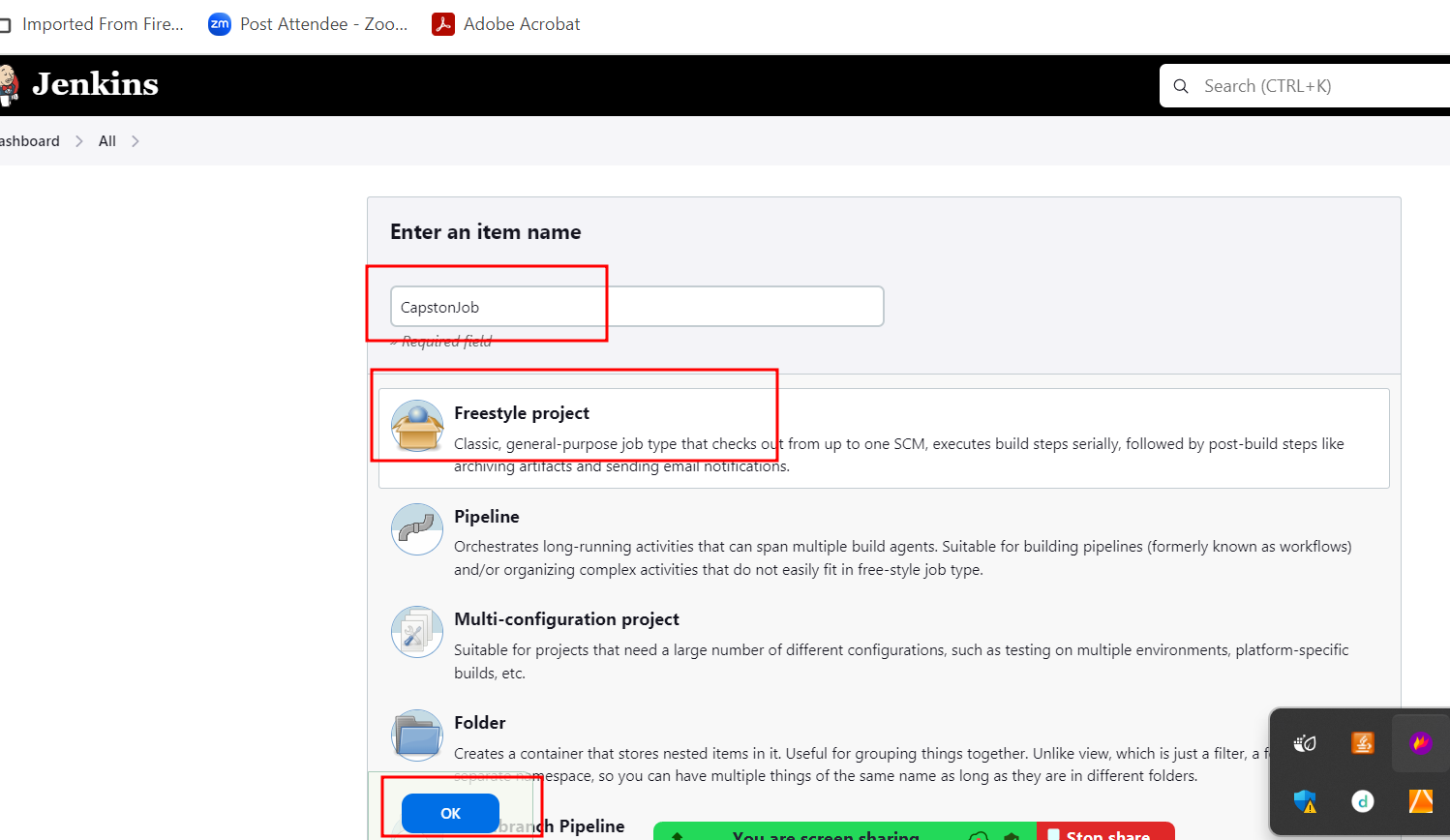


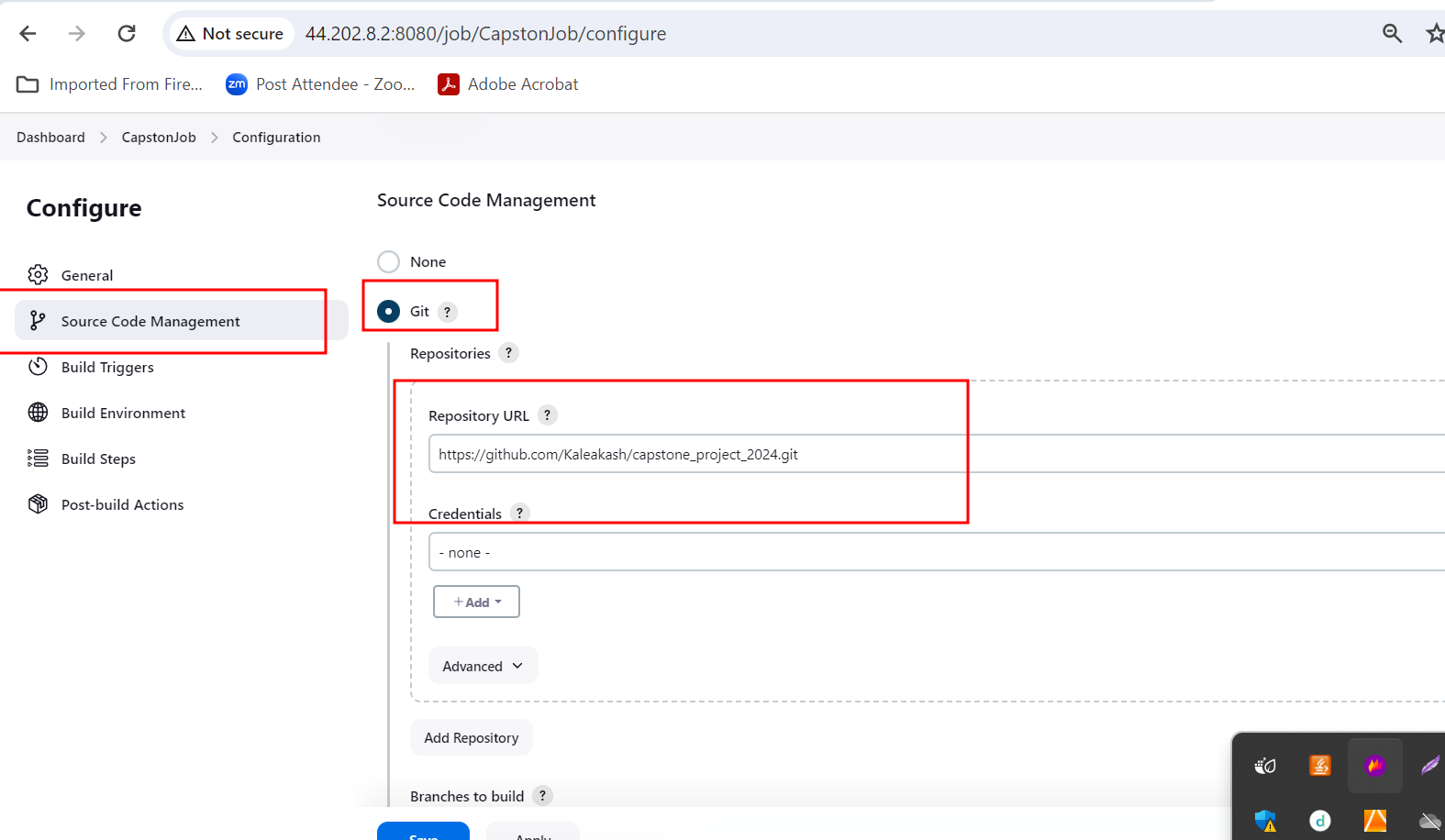
Login to Jenkin dashboard

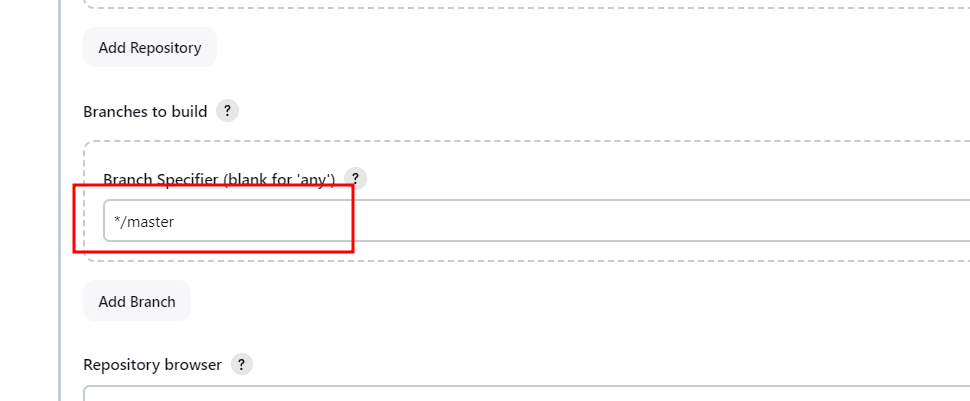


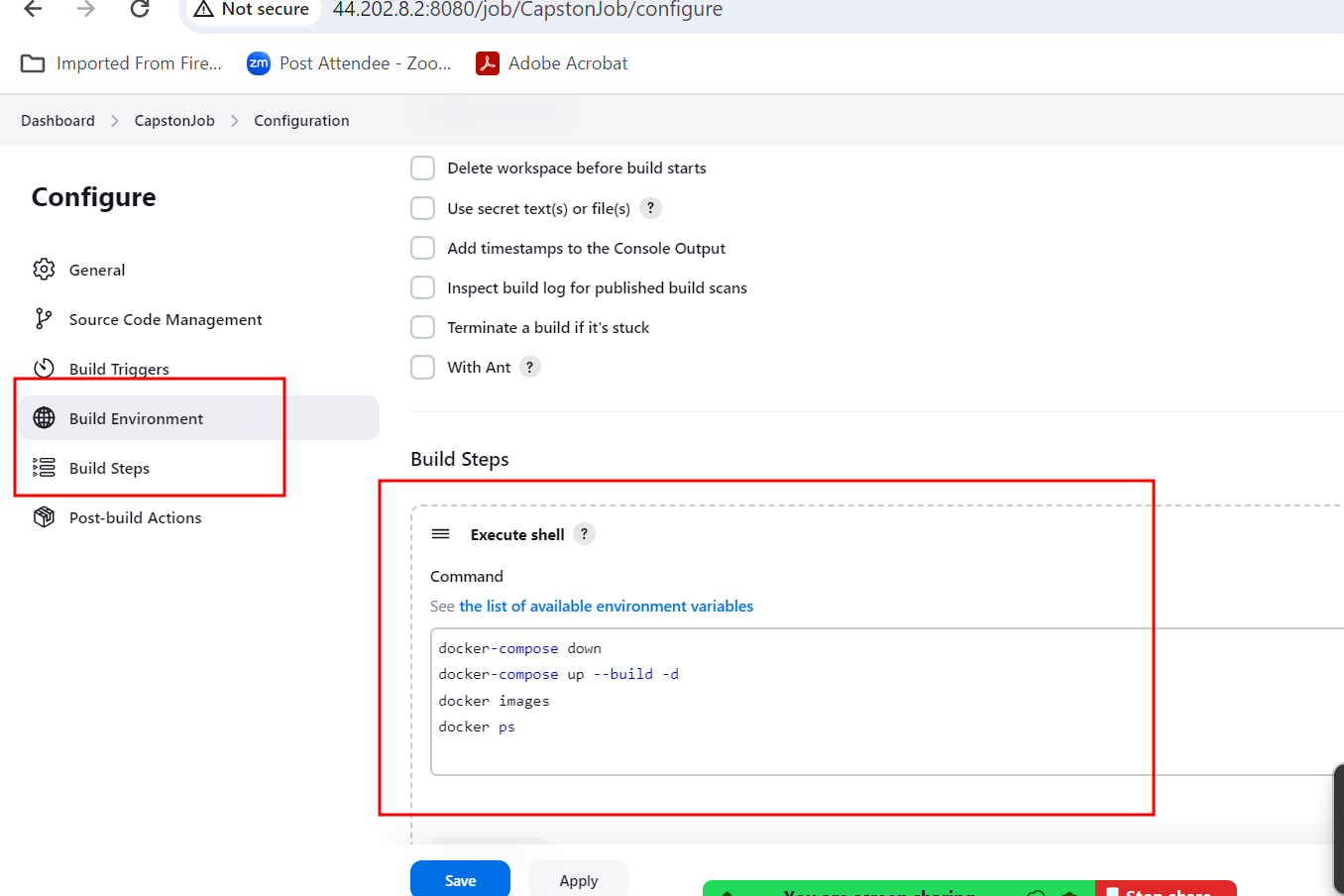
We need to create Jenkin job or pipe line job

Which is responsible to pull the project from github account and run docker-compose file.









Apply and save

Then build it

