First create backend project spring boot with mysql database

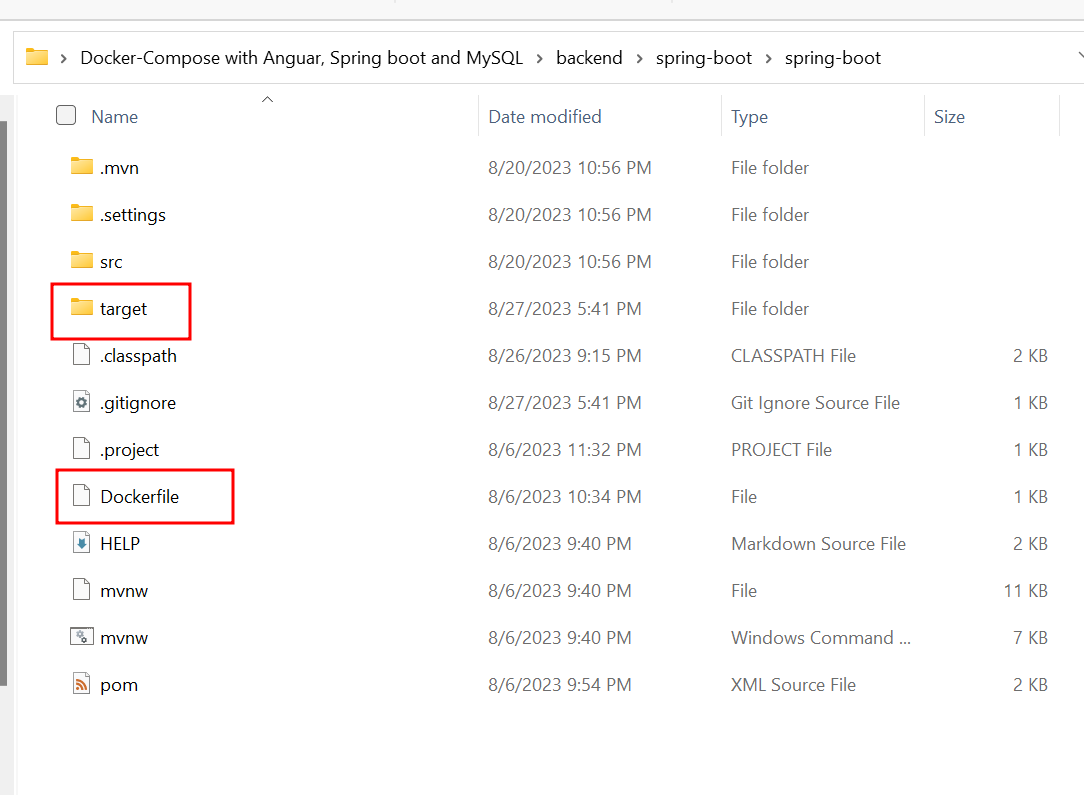
Then create frontend project using angular

using ng command build angular project

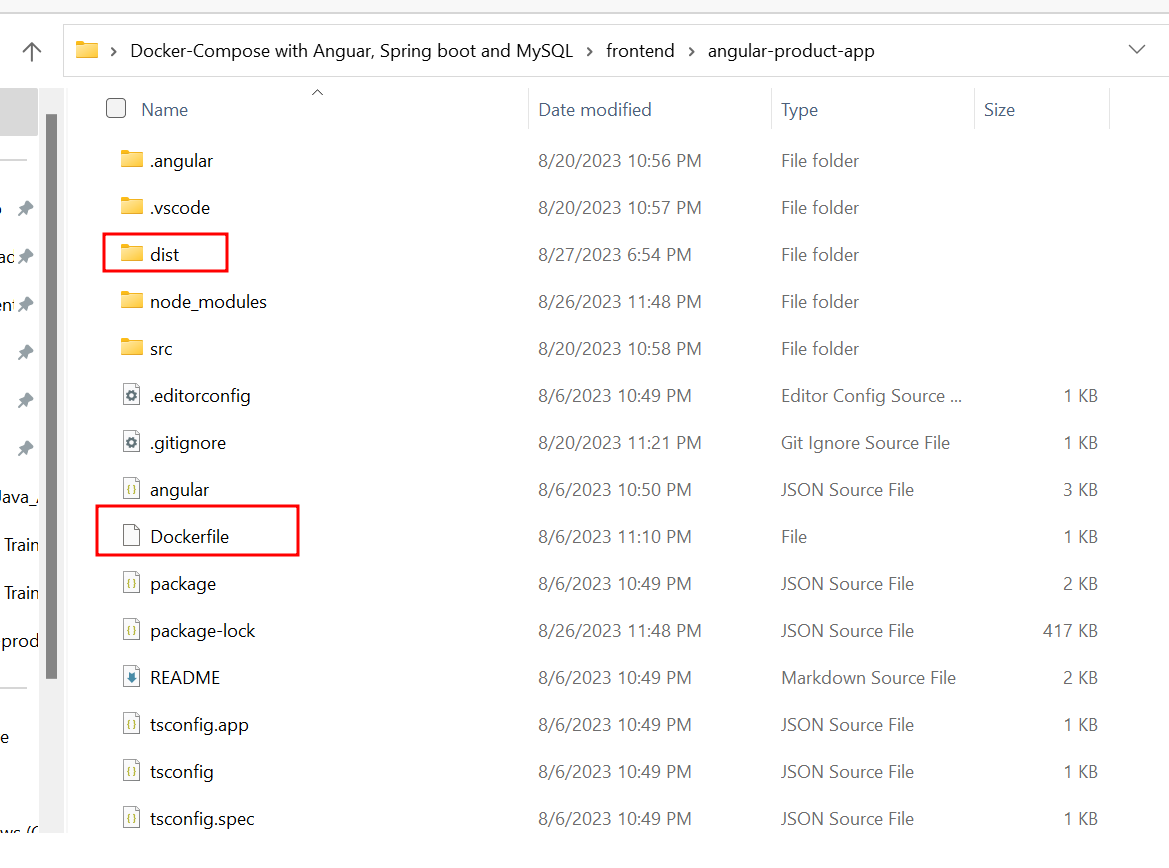
using mvn command create jar file for spring boot project.

Then create dockerFile for both frontend and backend project.

Backend project structure



Frontend project structure



Once development task done in local machine you need to push this project to remove repository

Please create docker-compose.yml file which is responsible to run more than one container

Ie mysql container

Spring boot container

Angular container

version: '3.3'

services:

  mysql-container:

    image: mysql:8

    environment:

      MYSQL\_ROOT\_PASSWORD: root

      MYSQL\_DATABASE: mydb

    ports:

      - 3306:3306

    restart: always

  springboot-container:

    build: ./backend/spring-boot/spring-boot/

    depends\_on:

      - mysql-container

    ports:

      - 9090:9090

    restart: always

  angular-container:

    build: ./frontend/angular-product-app/

    ports:

      - 80:80

So inside project folder (which contains backend and frontend create git repository)

git init

git add .

git commit -m “initial project ready”

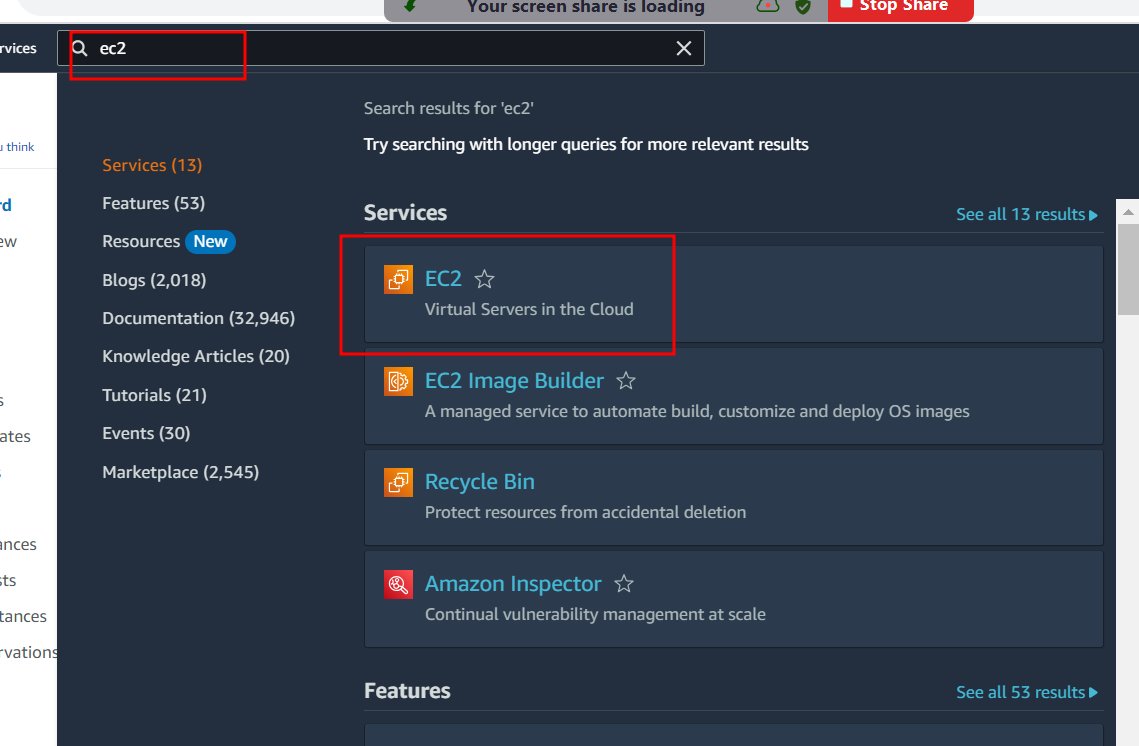
git remote add origin URL (URL remote repository URL)

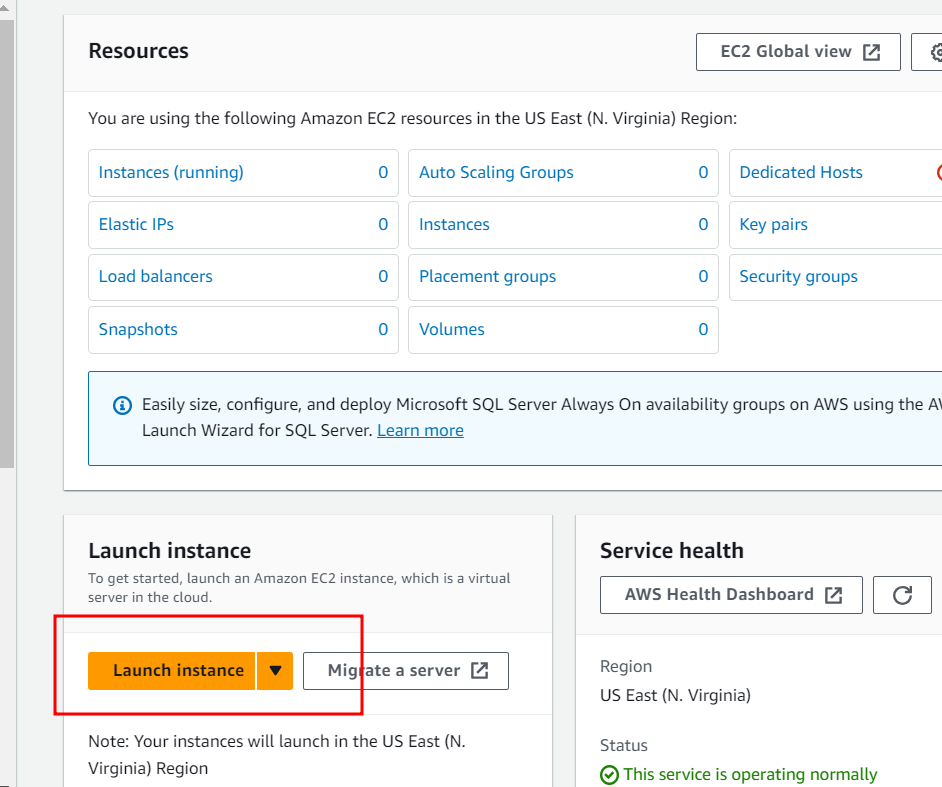
git push -u origin main/master

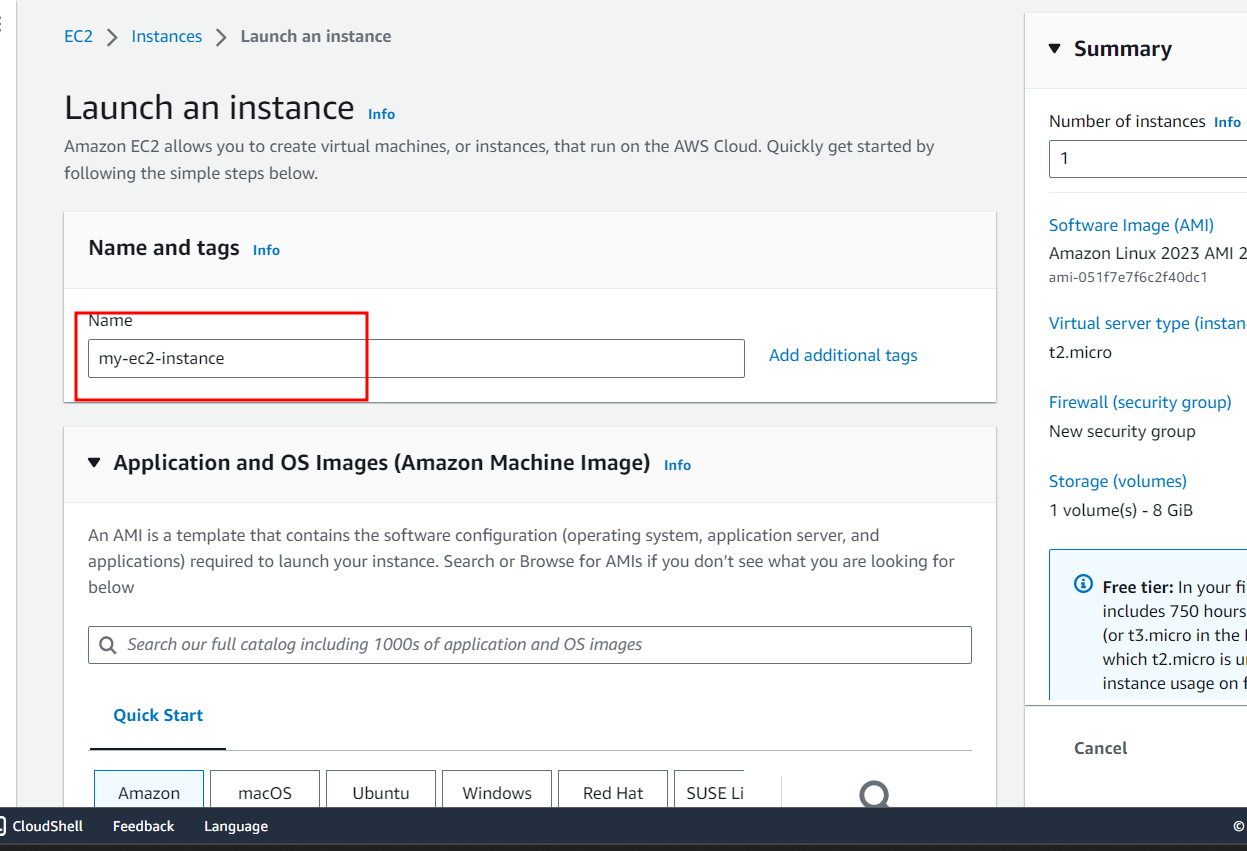
after push please check this code in your remote repository.

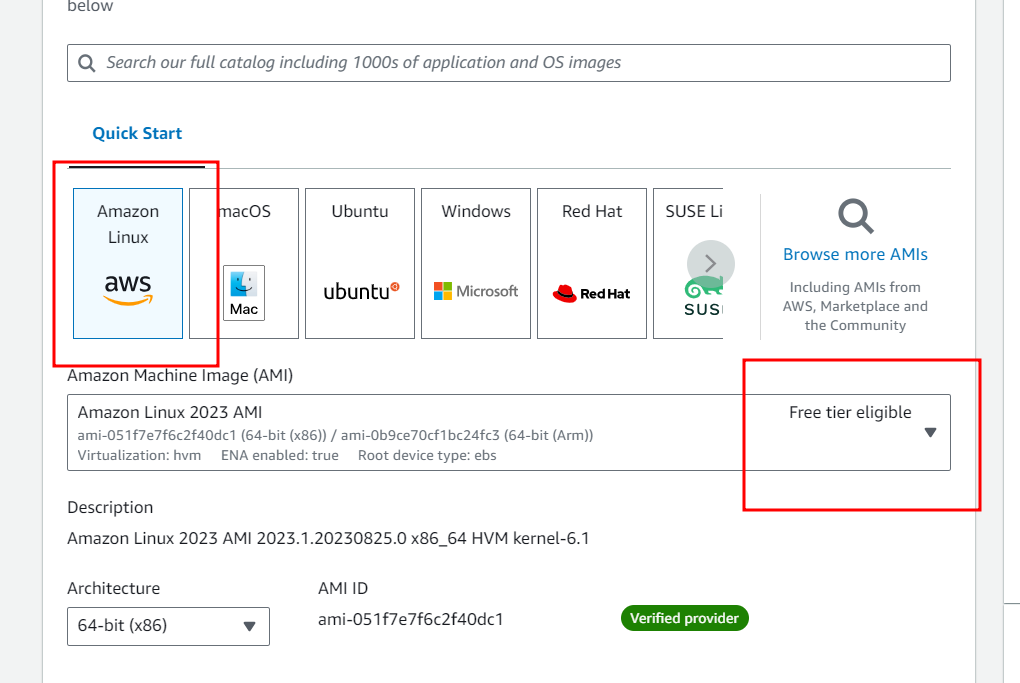
Please login to AWS account.

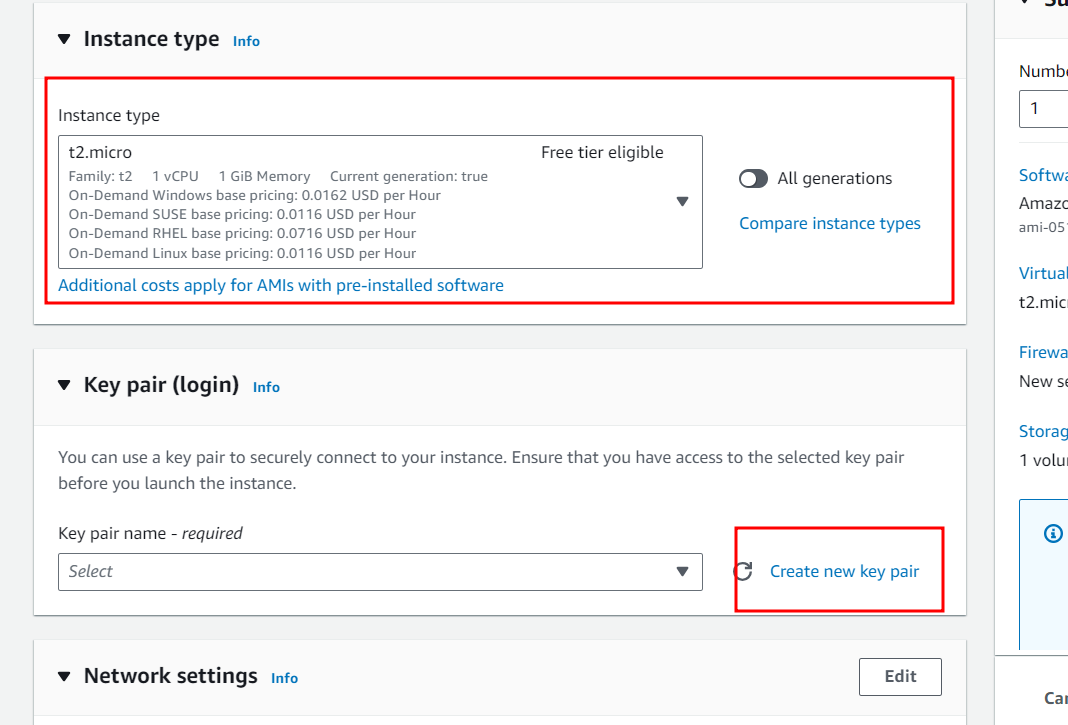
Then search EC2 instance option

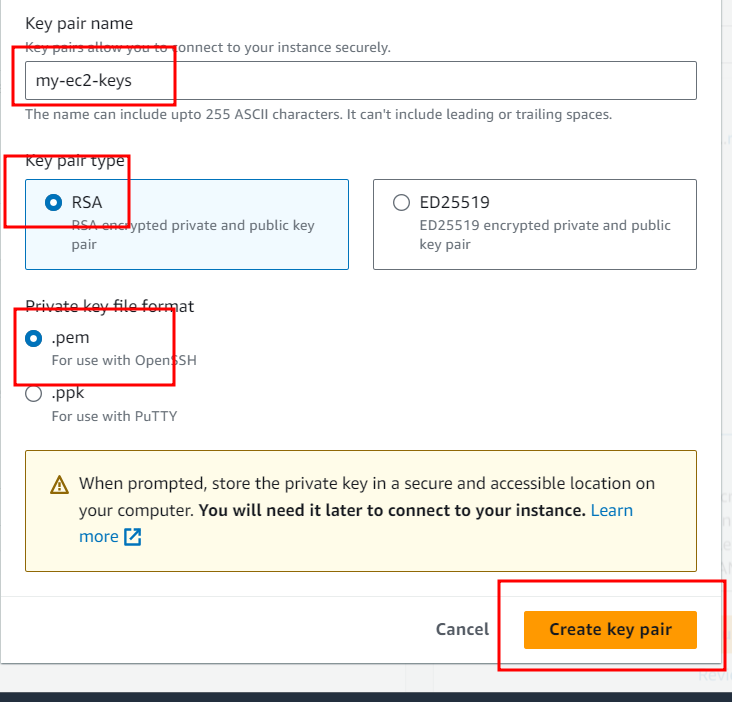




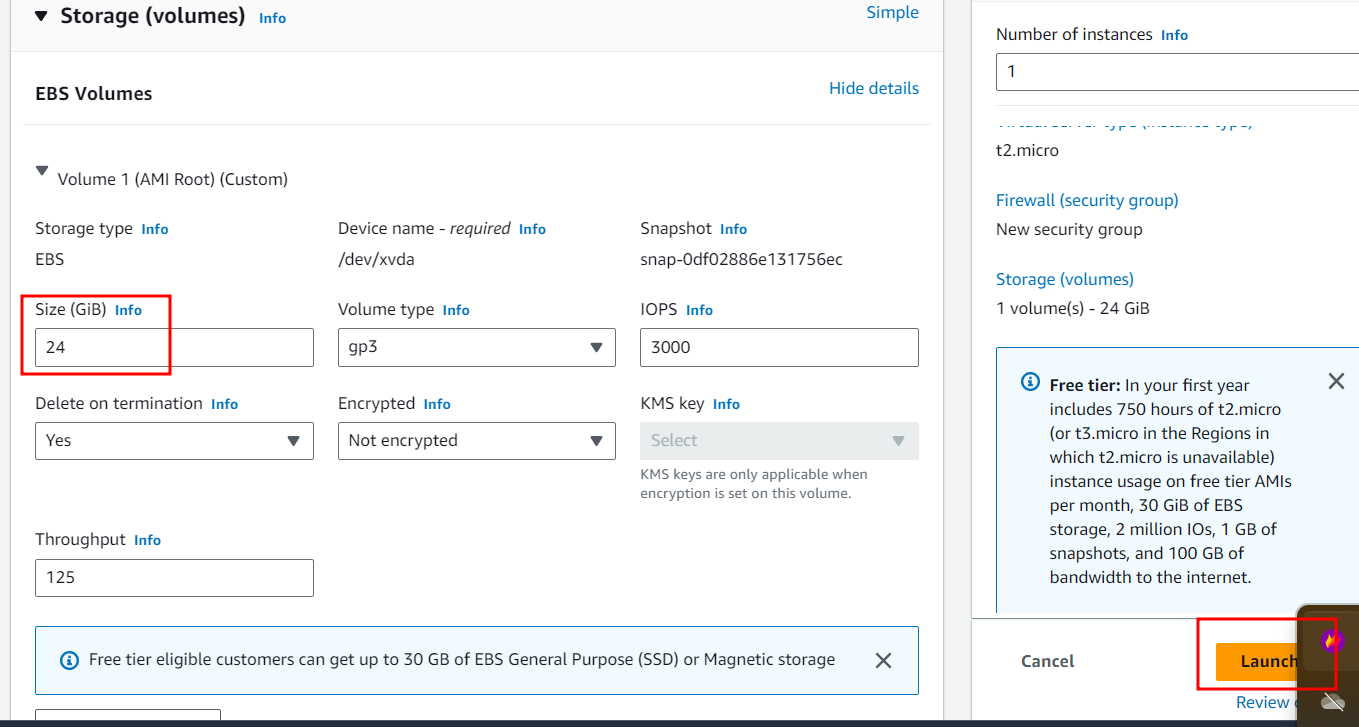


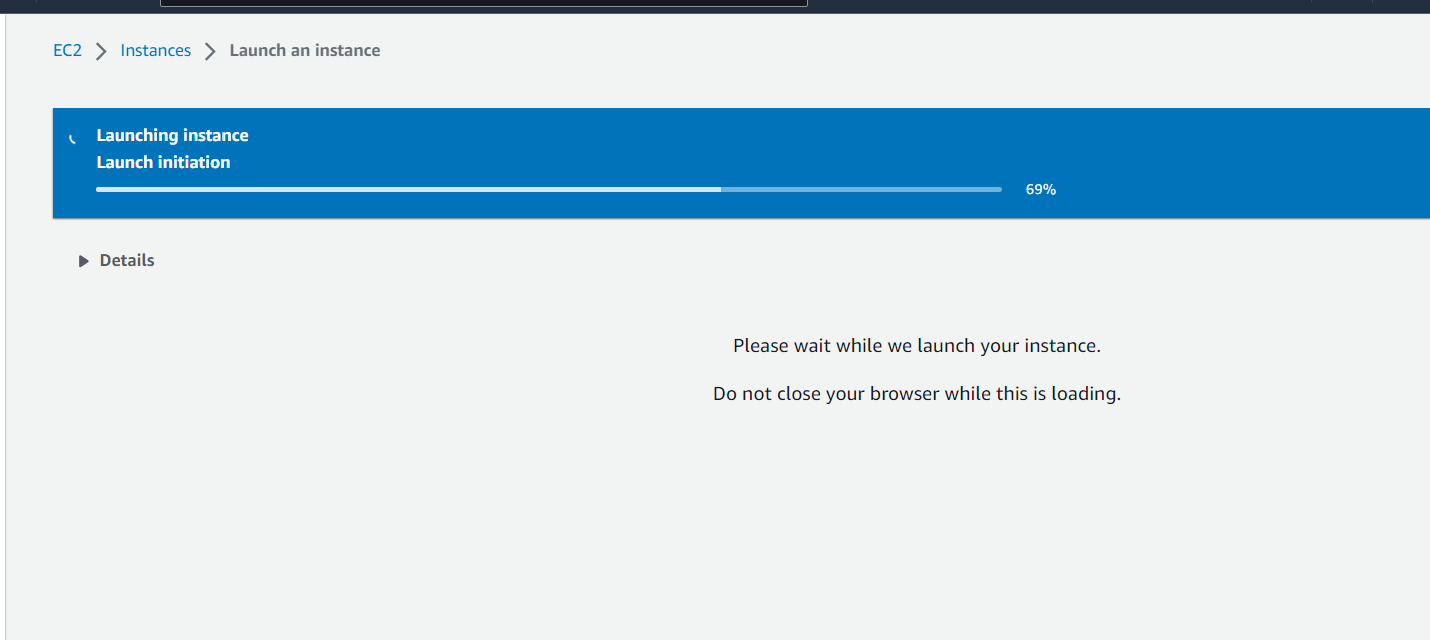


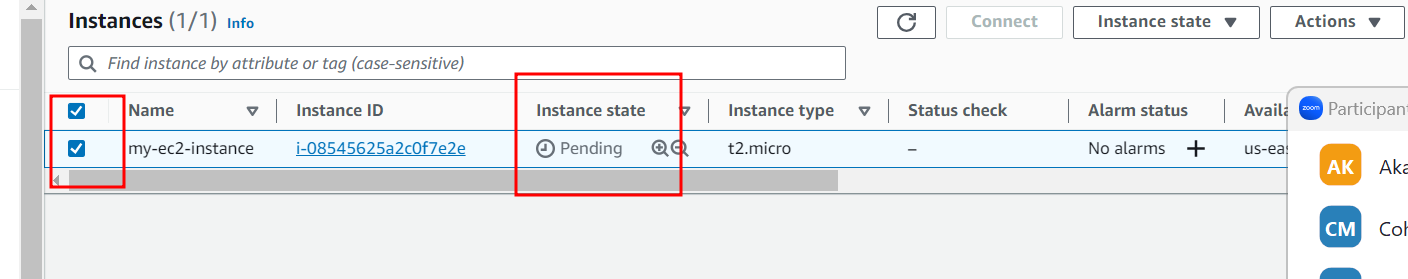


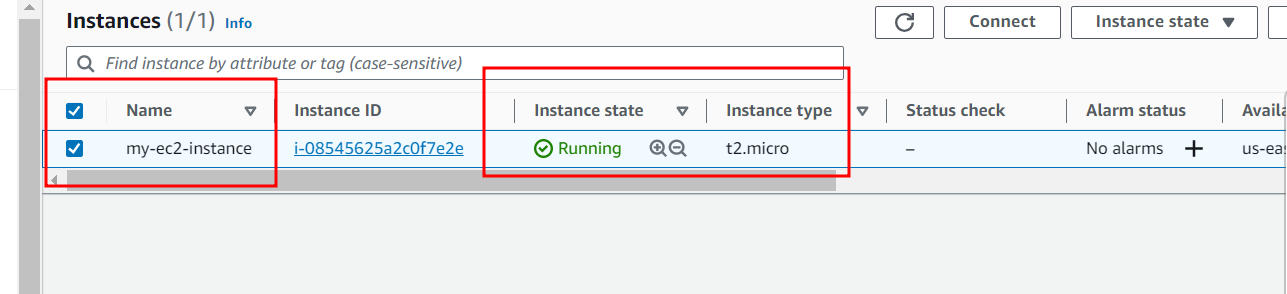


Download key and remember key path.









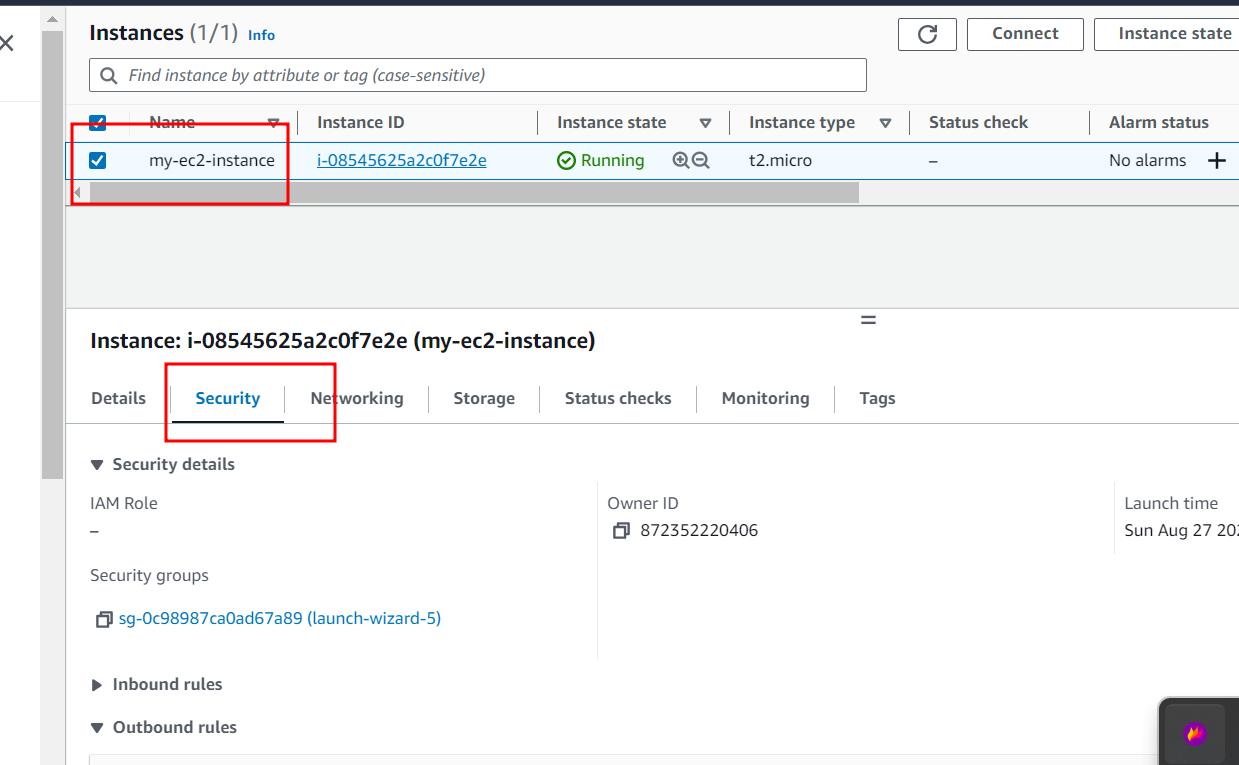
Now in EC2 instance we will open all port

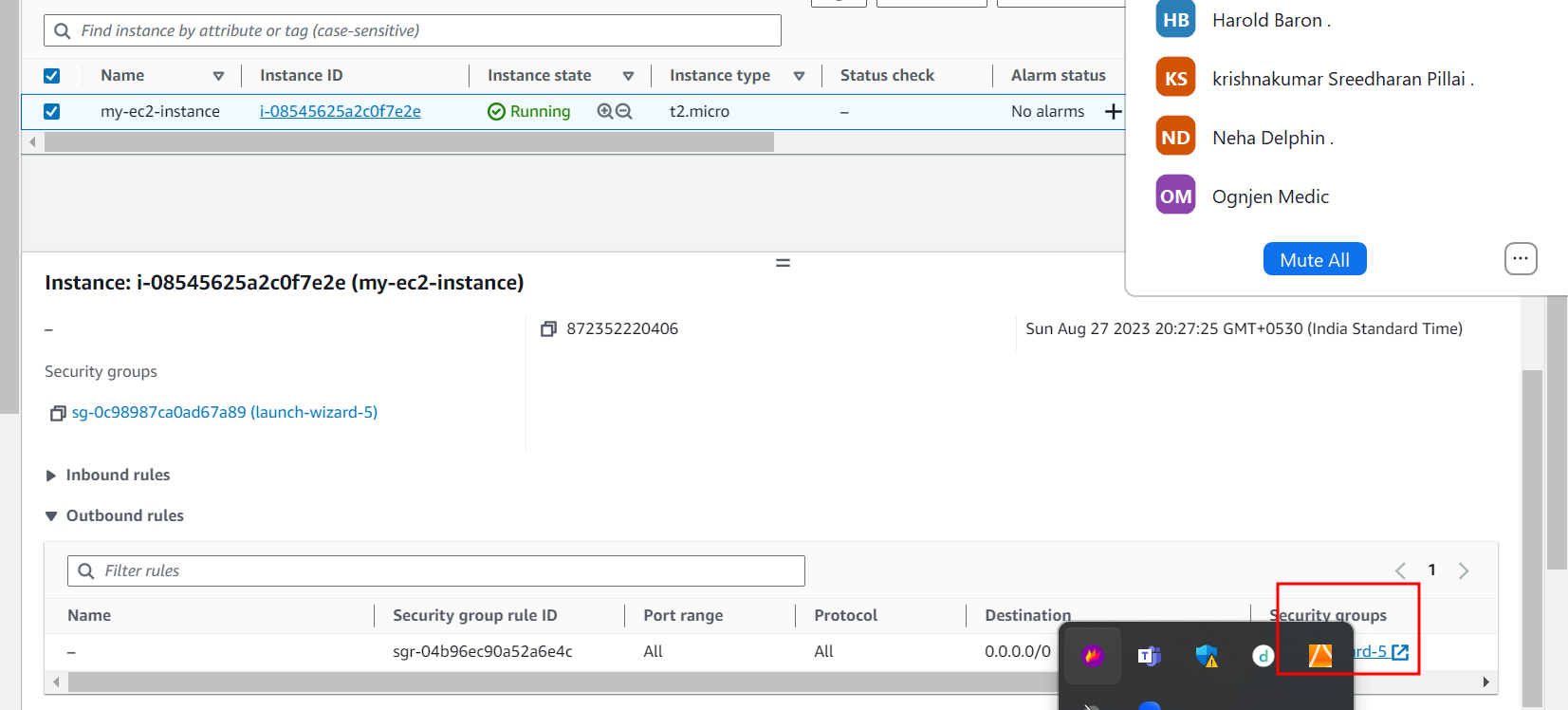
3306 🡪 mysql database

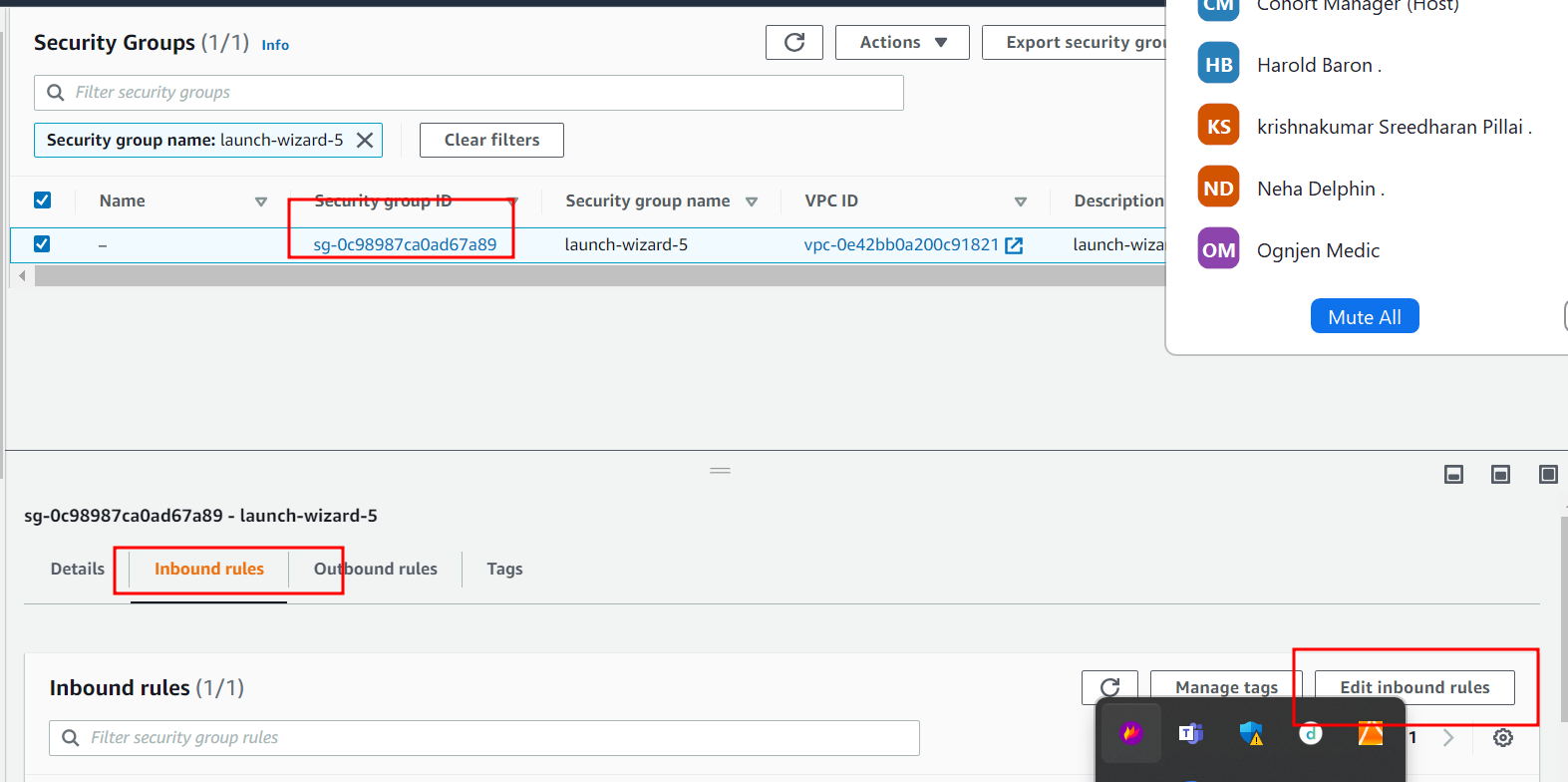
9090🡪 spring boot

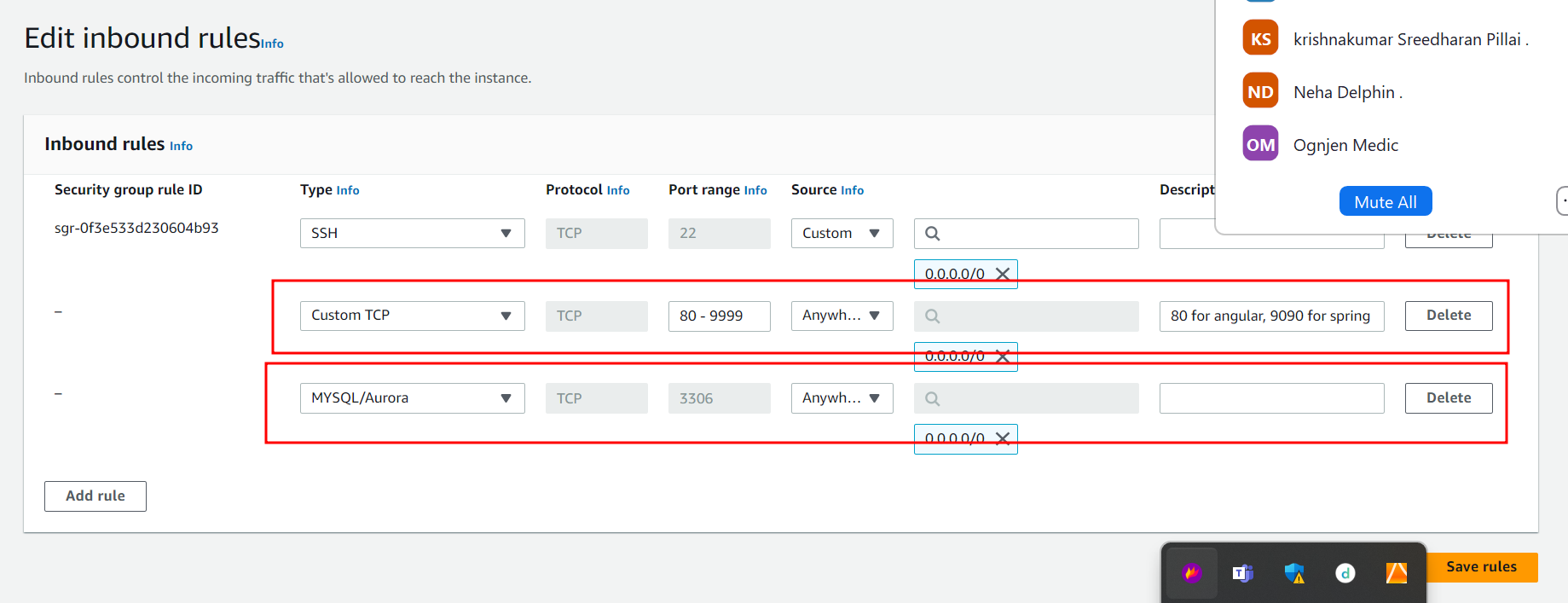
80🡪 angular with nginx server

8080🡪 Jenkin







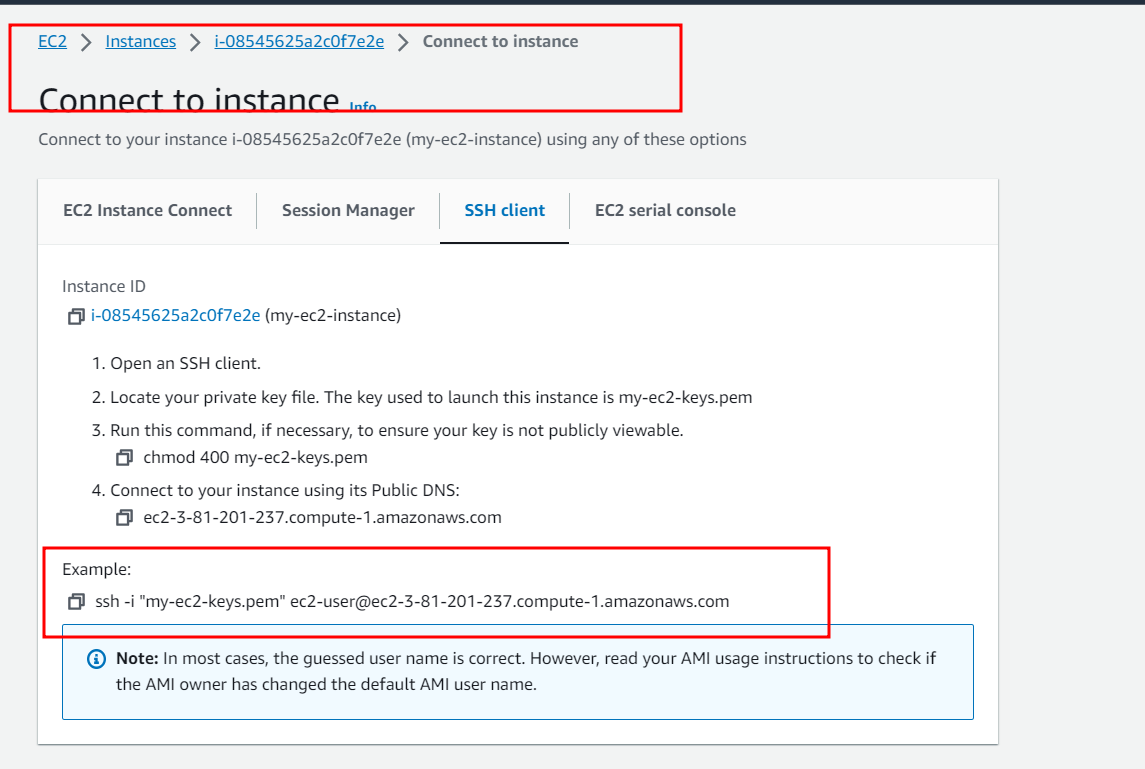


Close security wizard tab.

Now we are going to connect EC2 instance using terminal

Ie git bash (for window user)

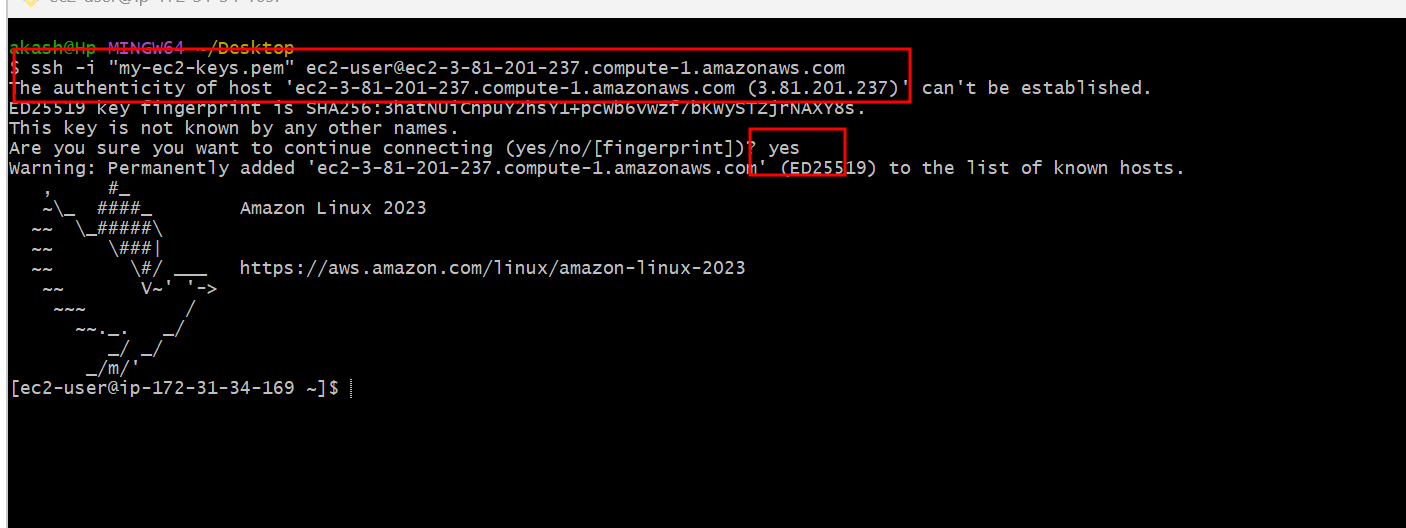
Non window user (normal terminal)



Please copy example ssh client commands

Then open git bash or normal terminal (but make sure .pem key file present int that location).

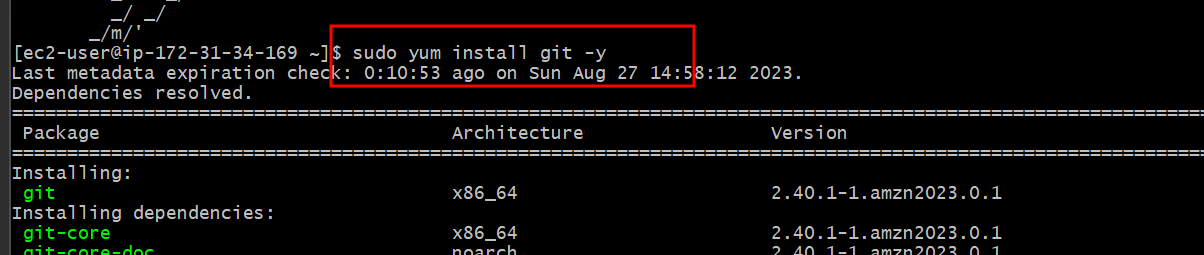




Now we need to install all required software in EC2 instance.

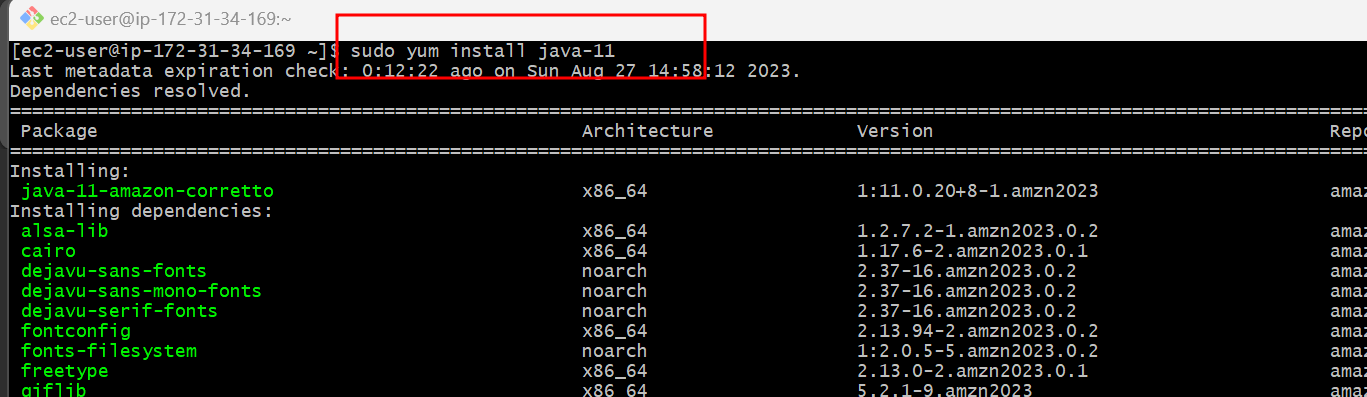
Git install

sudo yum install git -y



Java install

sudo yum install java-11



**Jenkin install**

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.key>

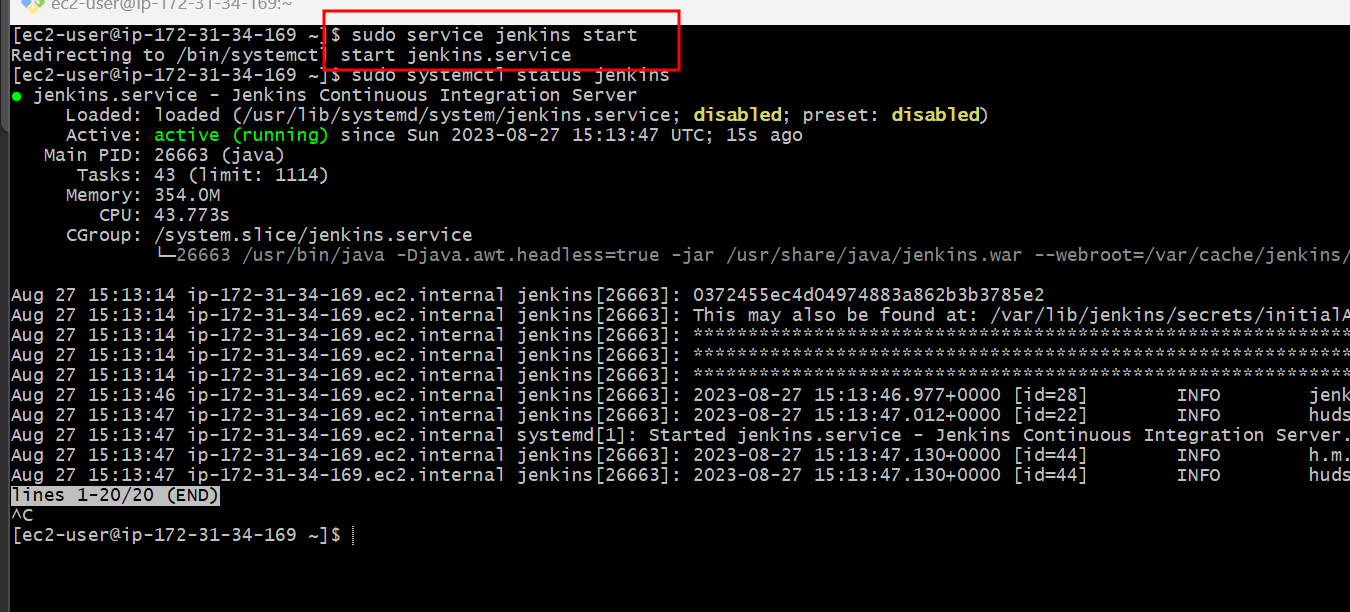
sudo yum install Jenkins

after installation we need to start the Jenkin

sudo service jenkins start this command to start

sudo systemctl status Jenkins this is use to check the status

to exit that flow please type cntr + c twice



Docker install

sudo yum install docker

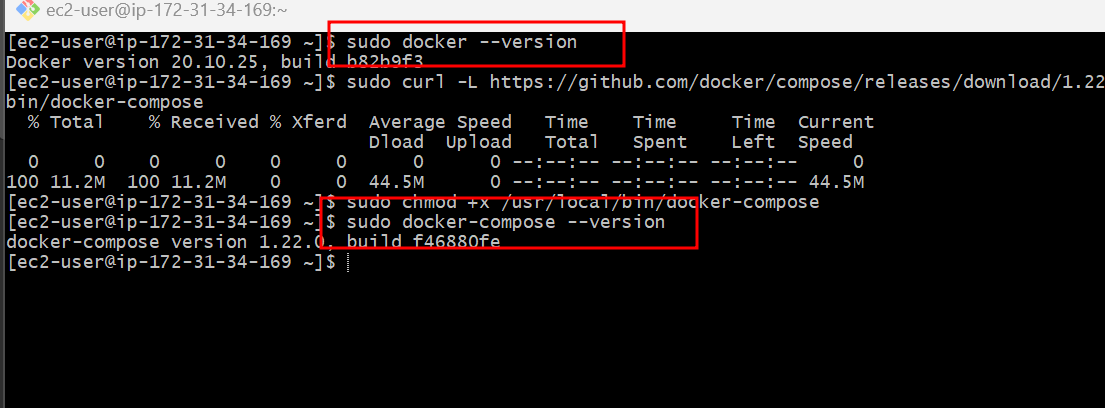
sudo service docker start

sudo docker –version

Docker compose

sudo curl -L https://github.com/docker/compose/releases/download/1.22.0/docker-compose-$(uname -s)-$(uname -m) -o /usr/local/bin/docker-compose

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



Provide permission to run docker and docker compose command in Jenkin environment.

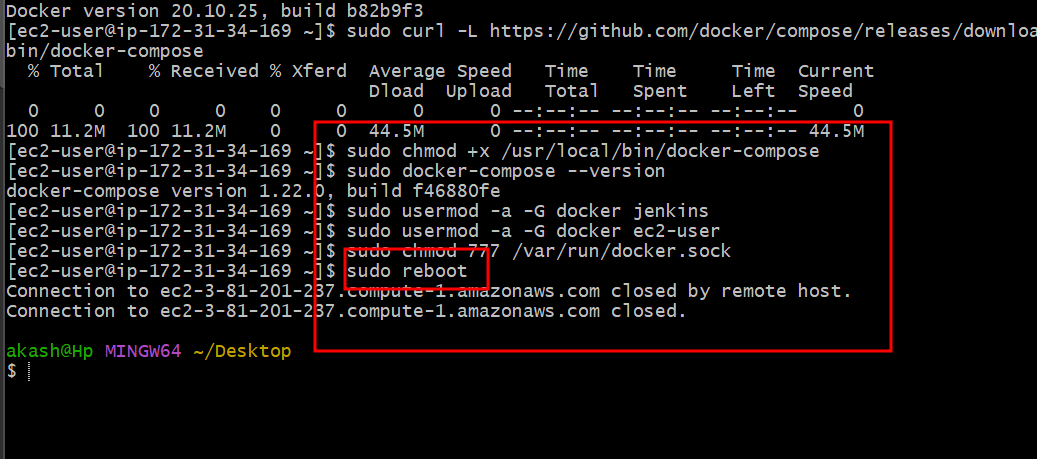
sudo usermod -a -G docker jenkins

sudo usermod -a -G docker ec2-user (ec2-user is user name of instance )

sudo chmod 777 /var/run/docker.sock

after this please restart Jenkins or ec2 instance.

Please sudo reboot command restart ec2 instance to update all changes.



Wait for few minutes to start ec2 instance.

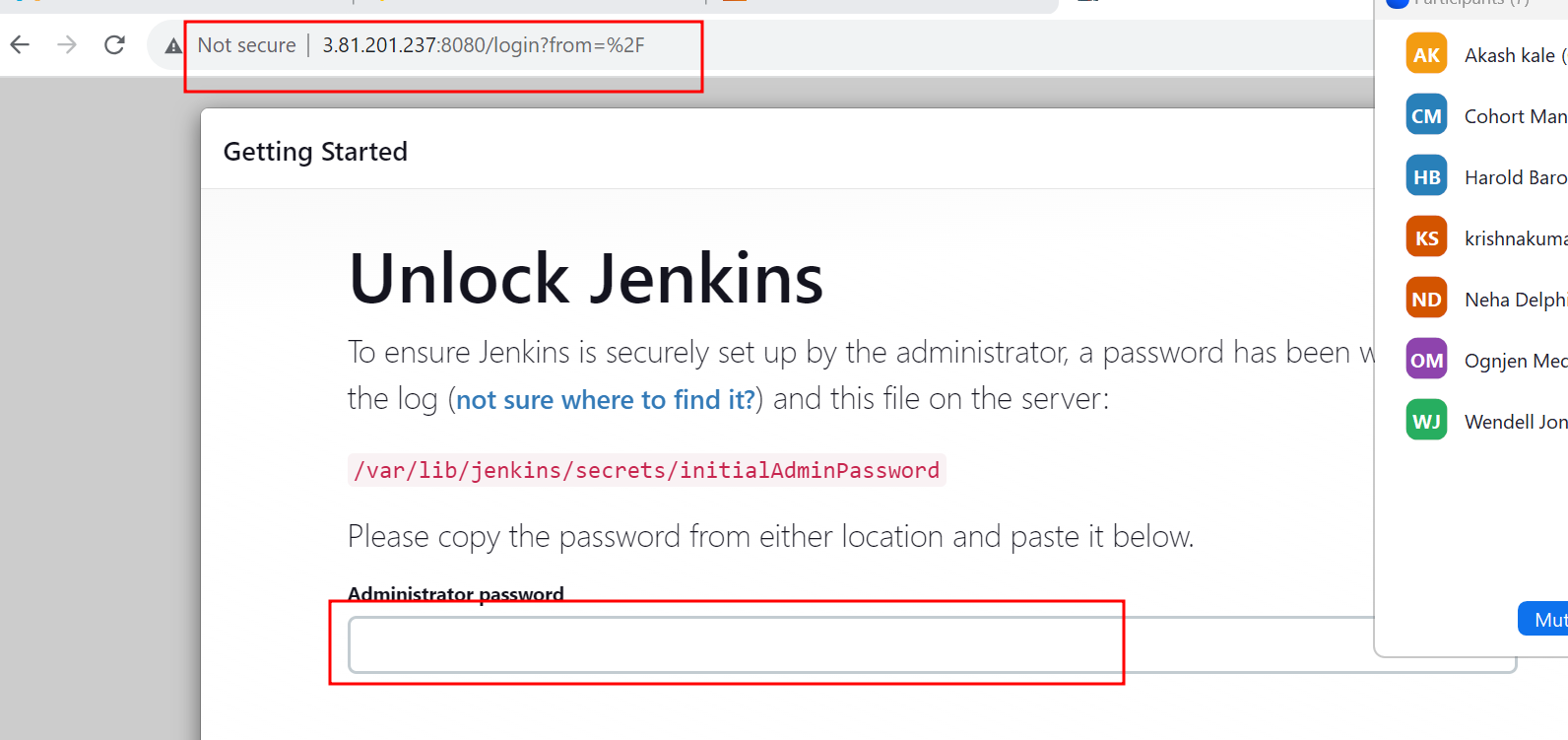
After reboot please connect to ec2 instance using ssh client command

Then start Jenkin service once again using command as

sudo service jenkins start

after start Jenkin please open ec2 dashboard then check Ec2 instance IP Address

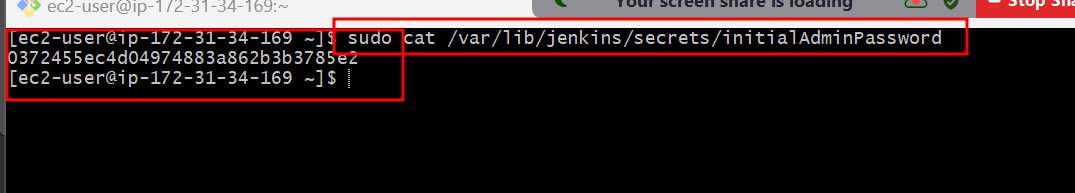
hit on browser as <http://ipaddress:8080>

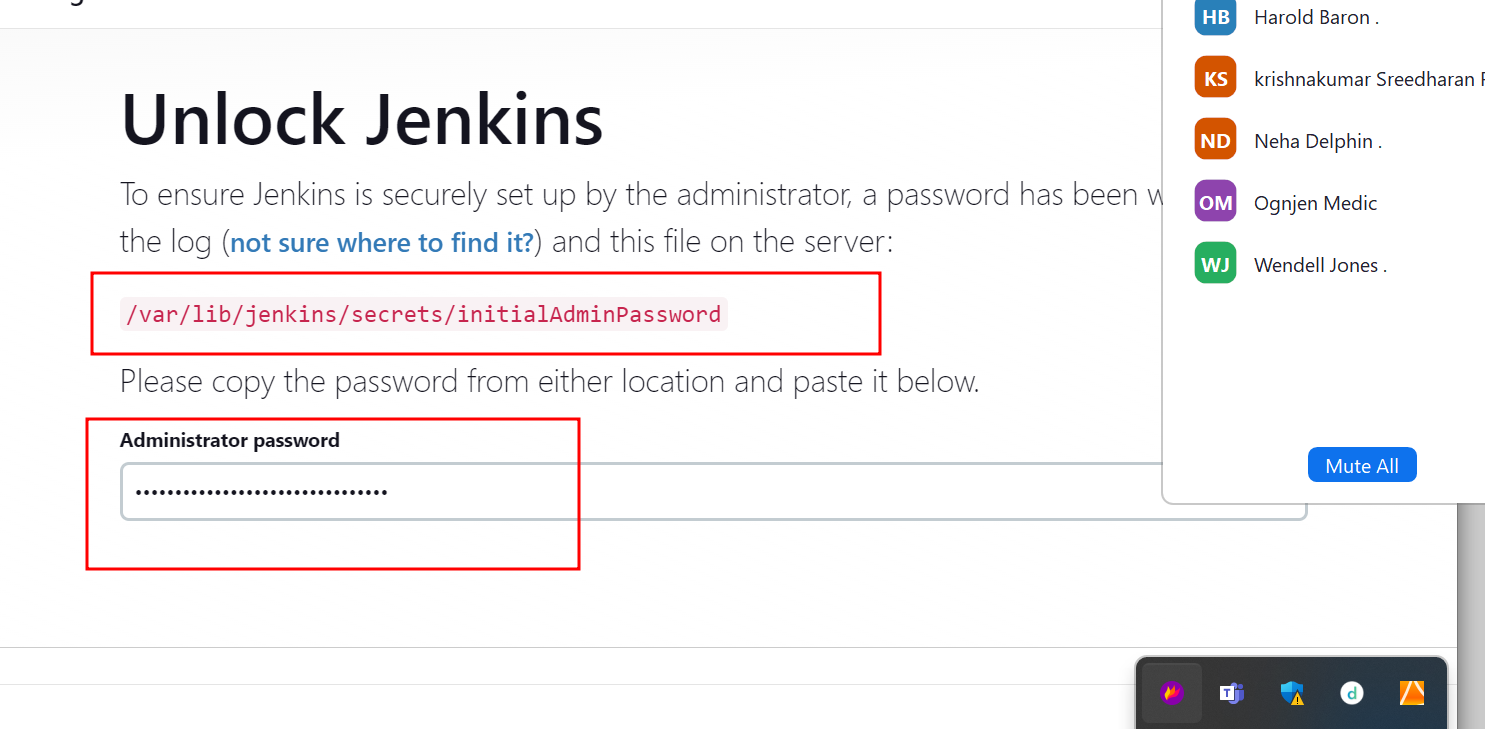


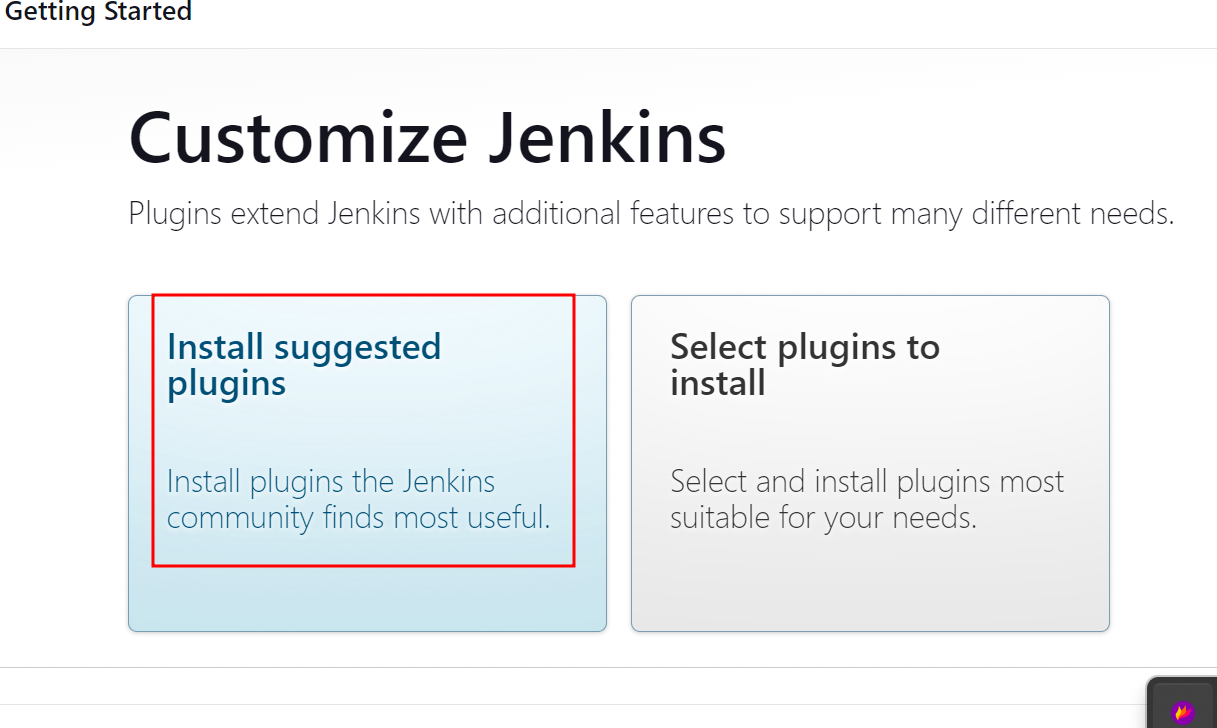
At the first time it will ask random password.

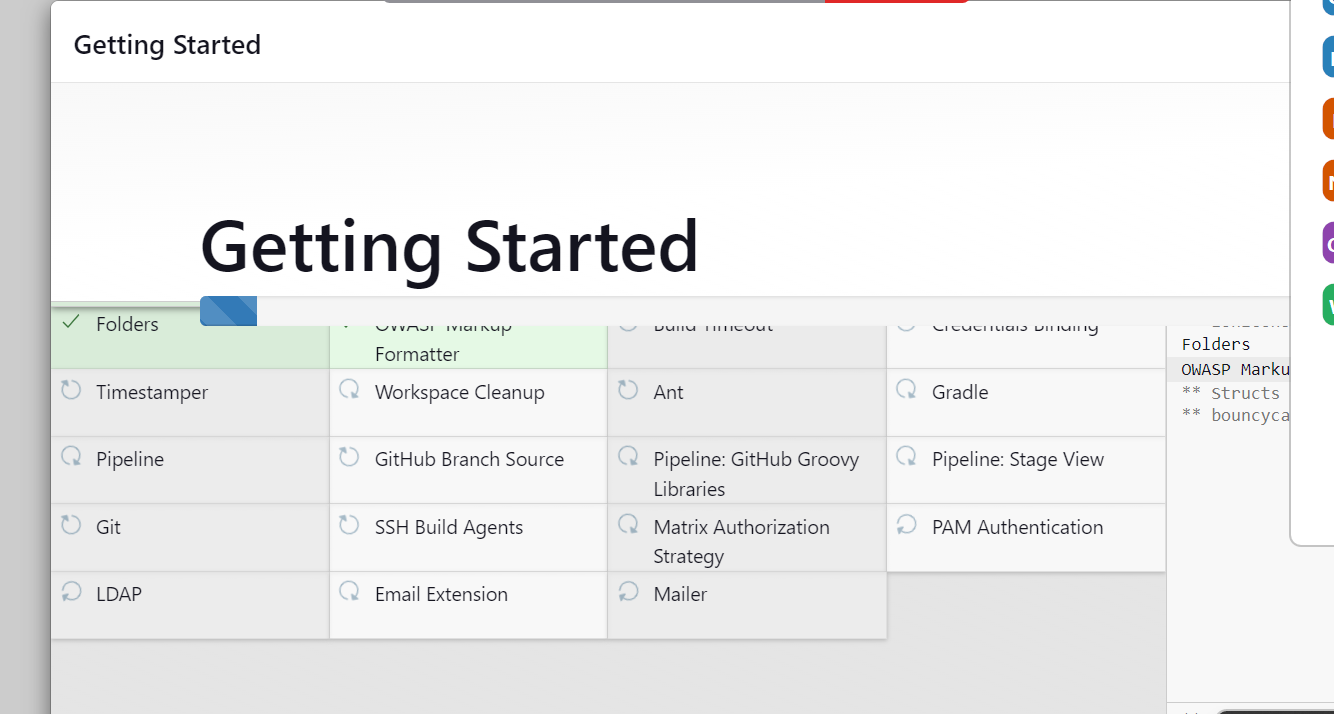
To get the password please write the command in ec2 instance as

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

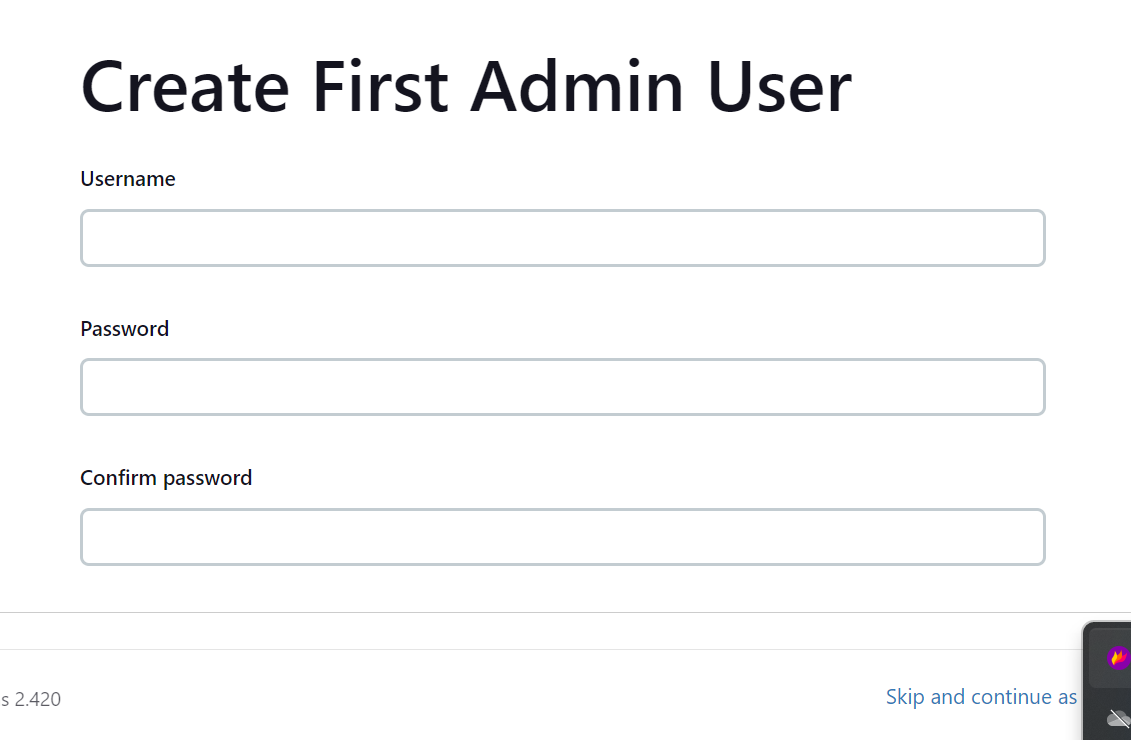


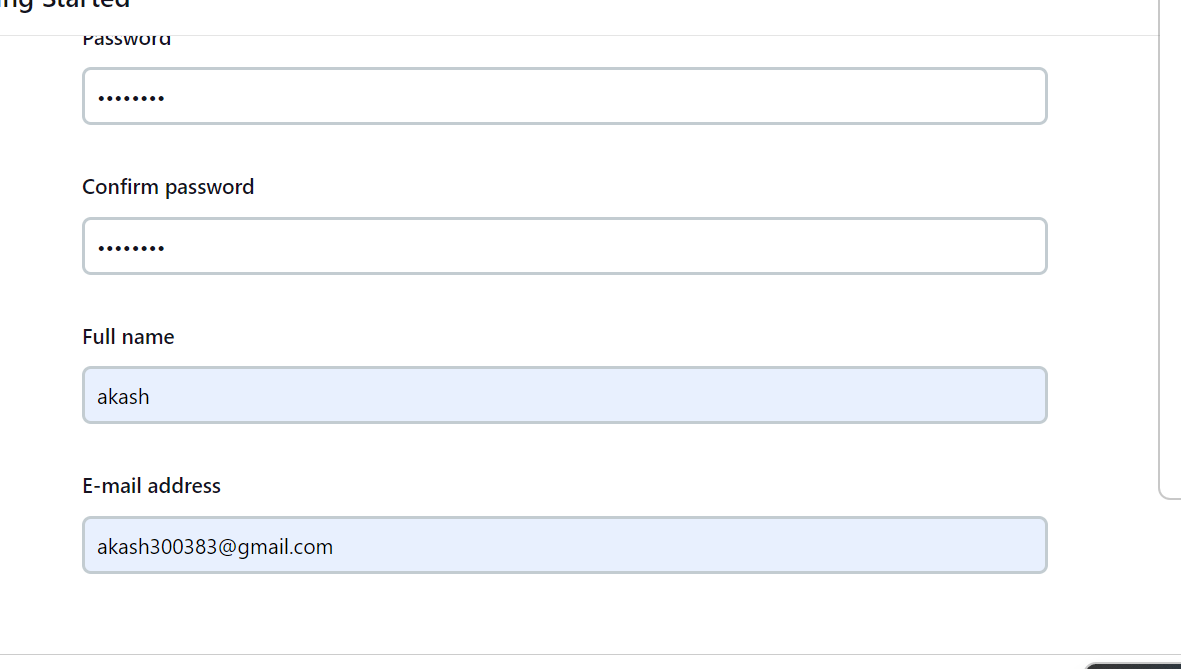


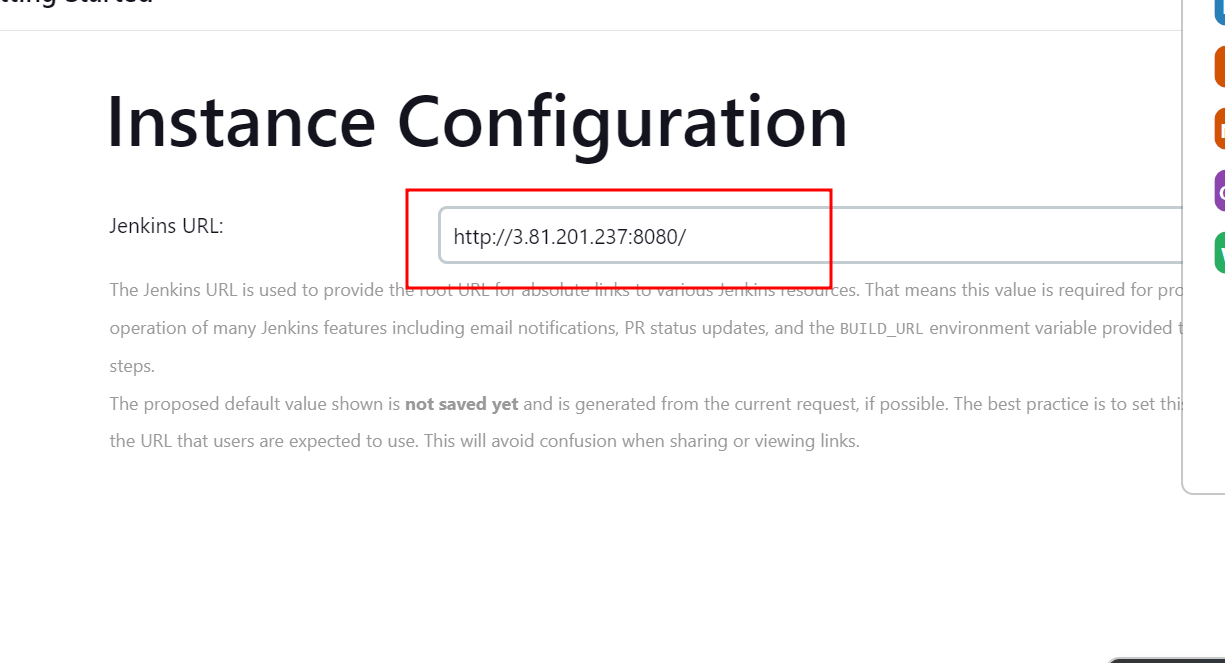


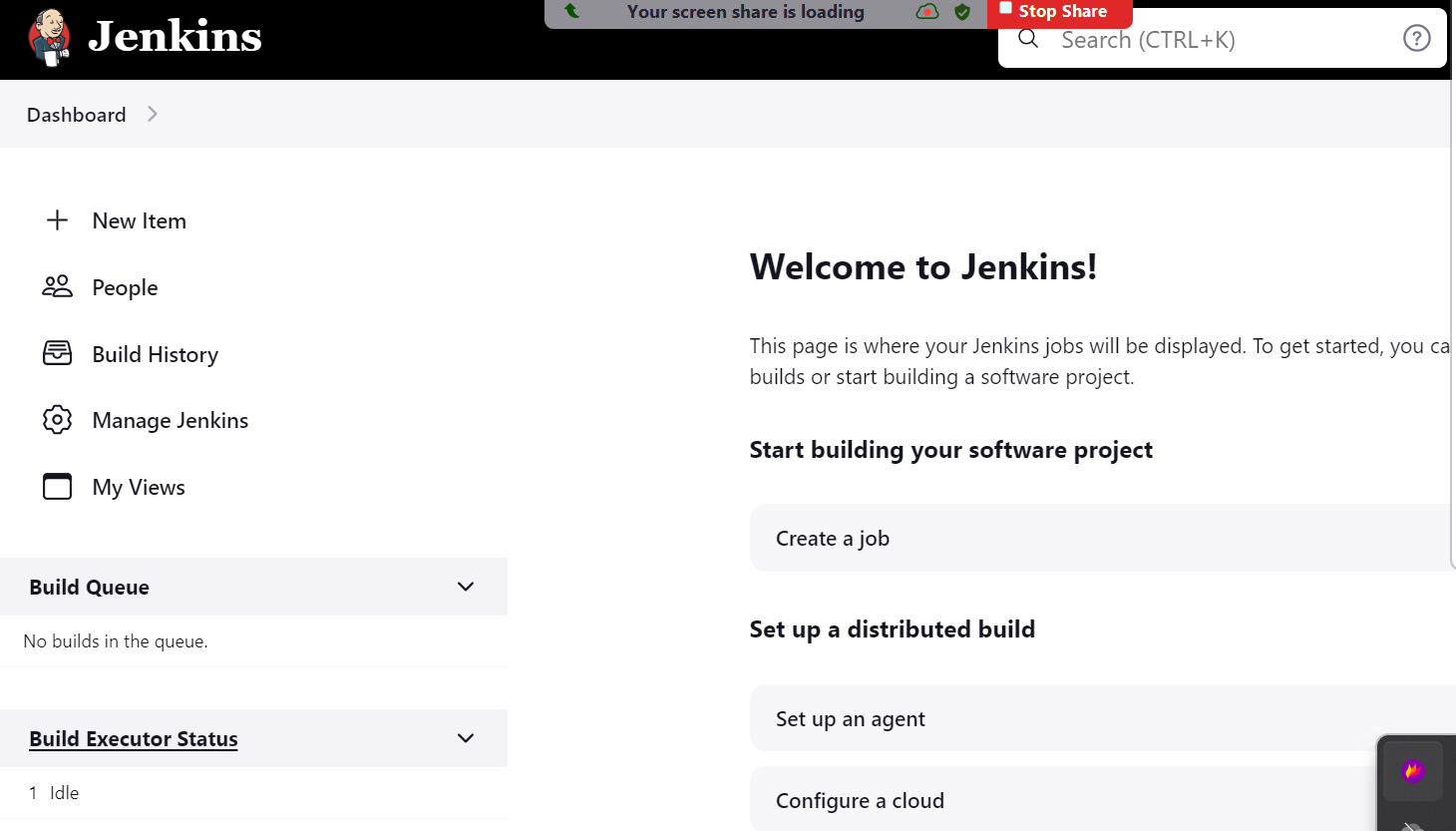


Then create your own account.

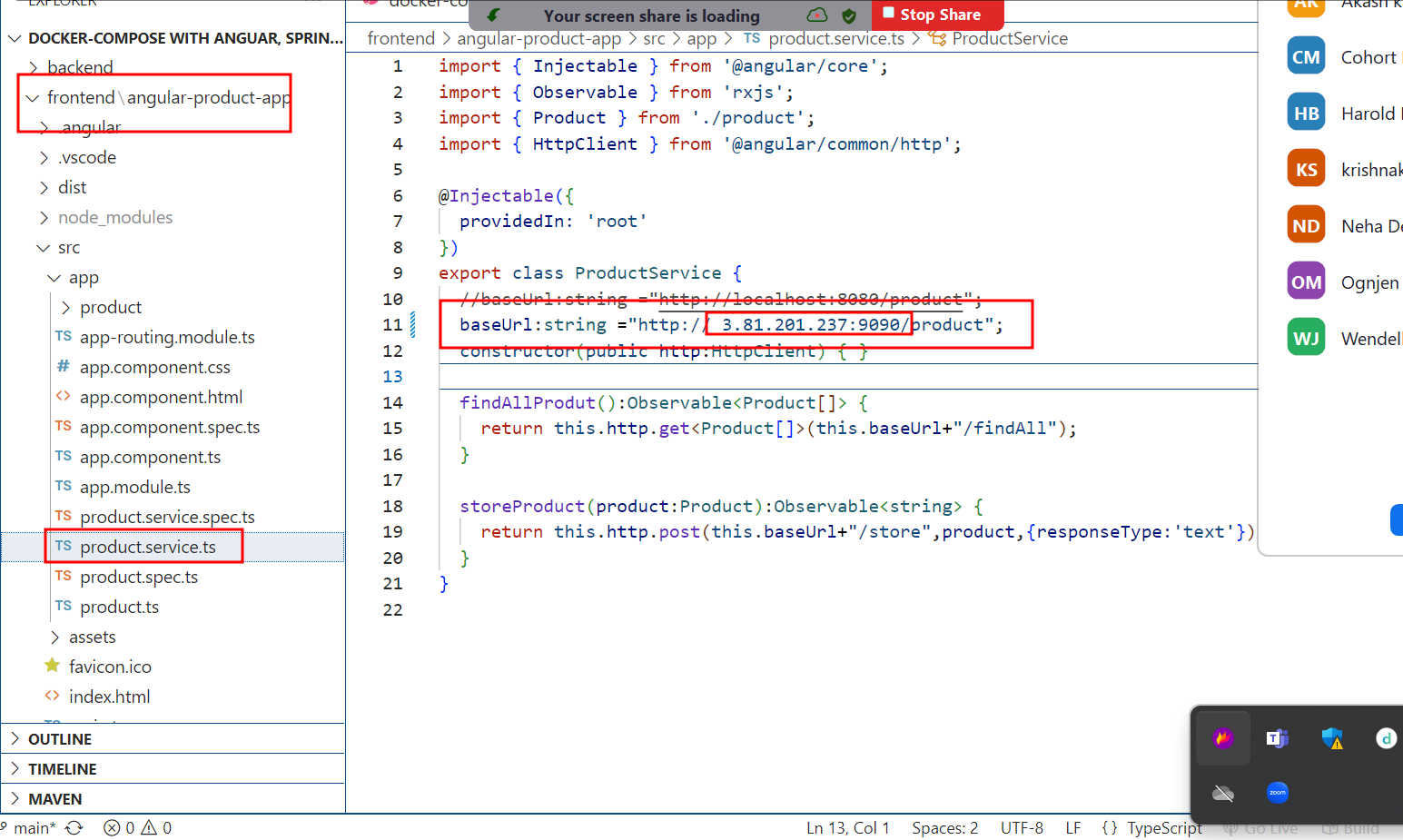








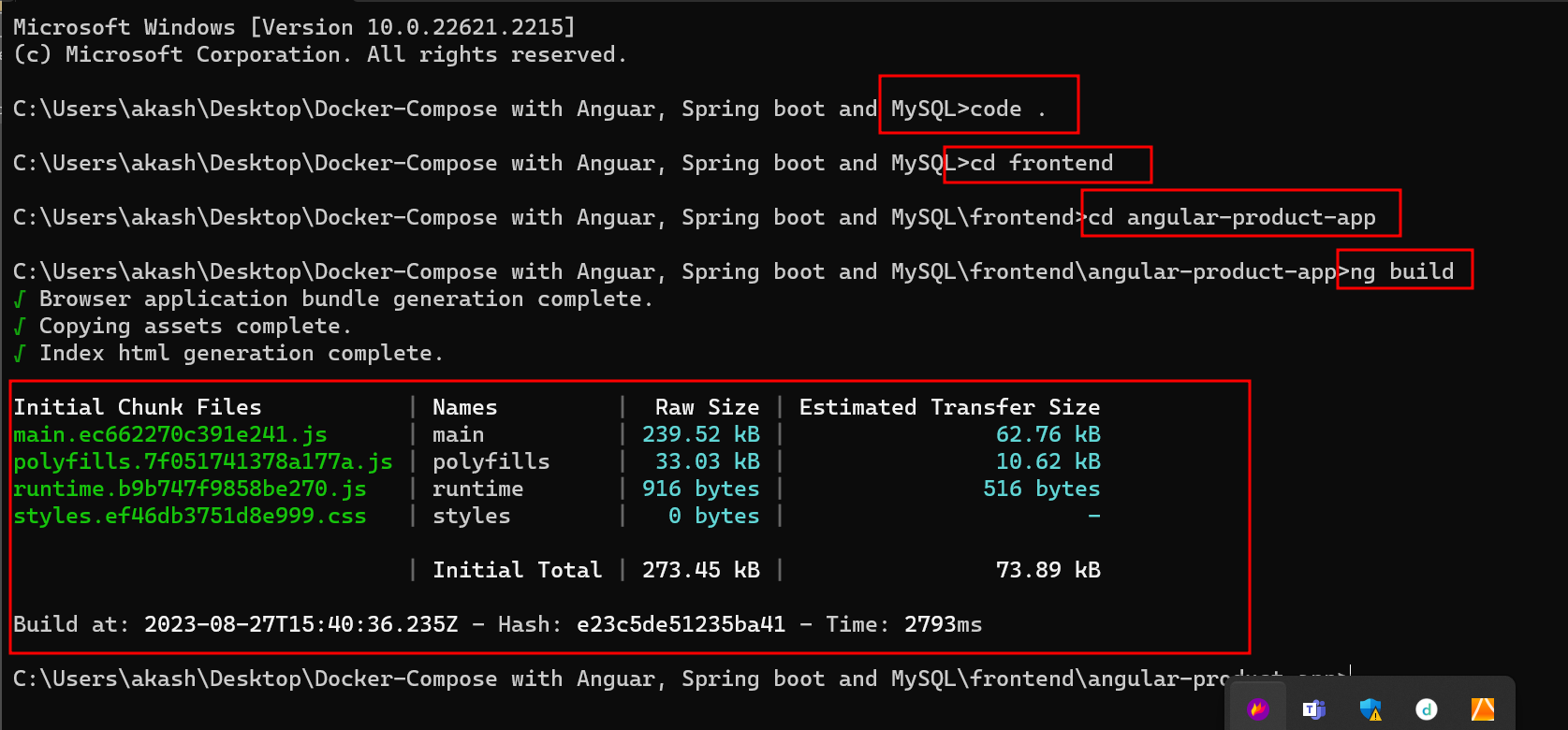
In angular project inside service class we need to write the IPAddress of EC2 instance to call frontend to backend.



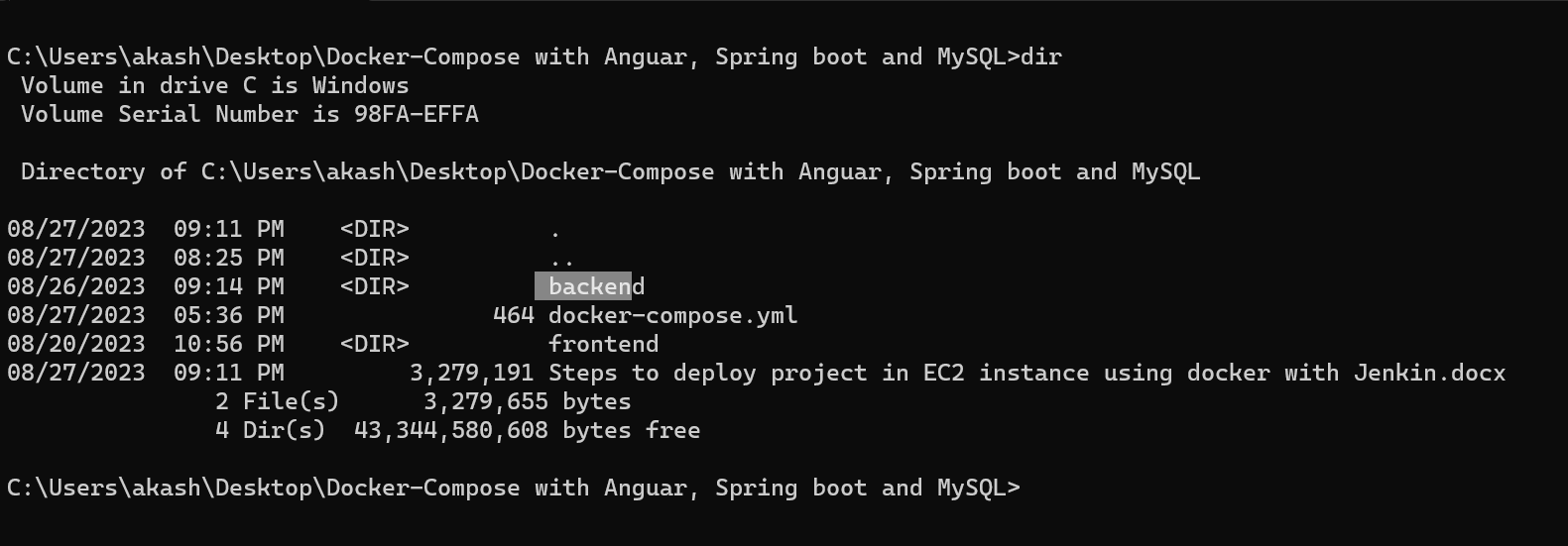
Please write your ec2 instance id address in every service file as of now.

Once we do any changes in frontend or backend we need re-build in local machine.

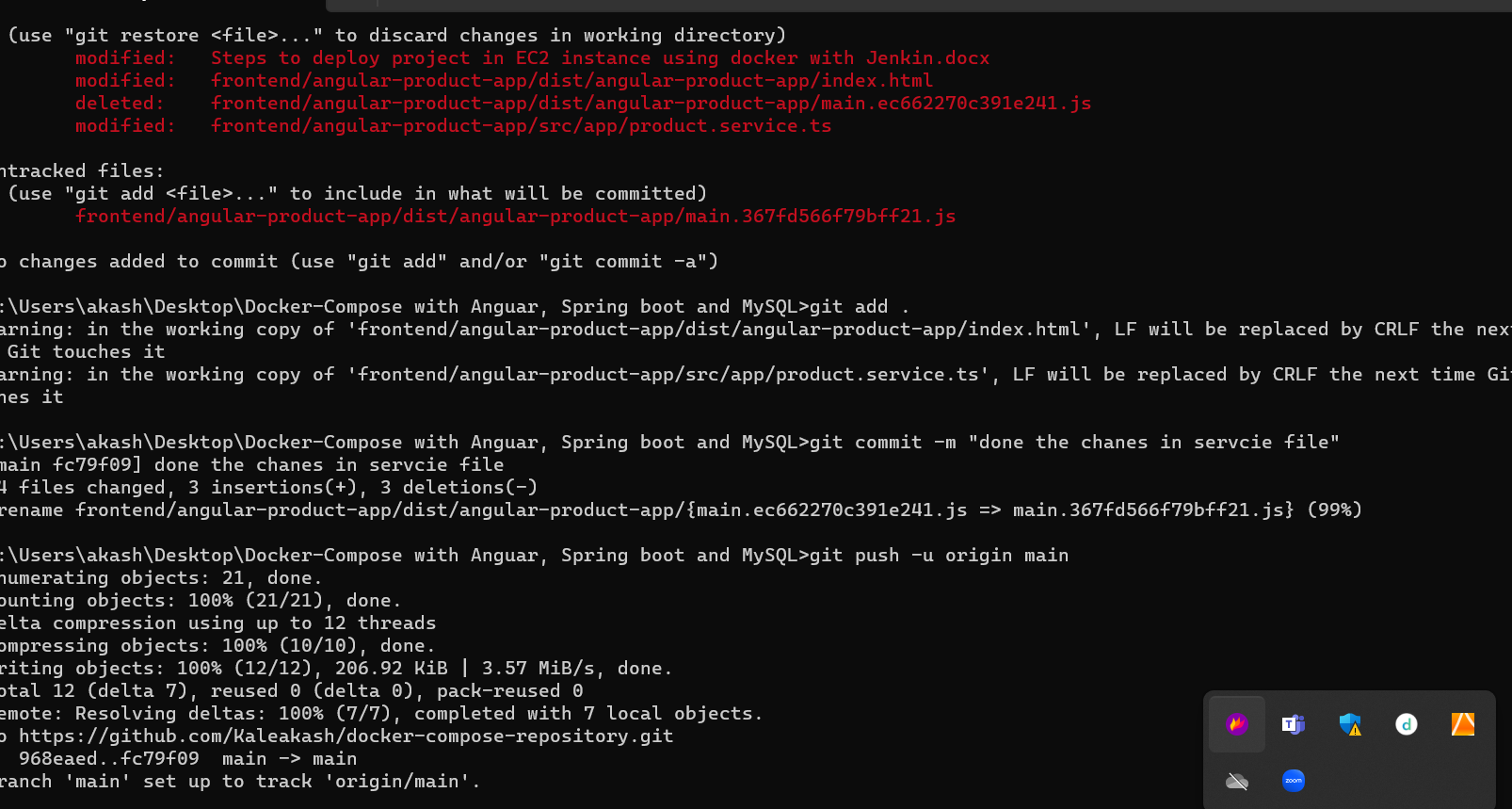
And we need to push this code once again in git hub repository.



This come to main folder to push the code in git hub account.

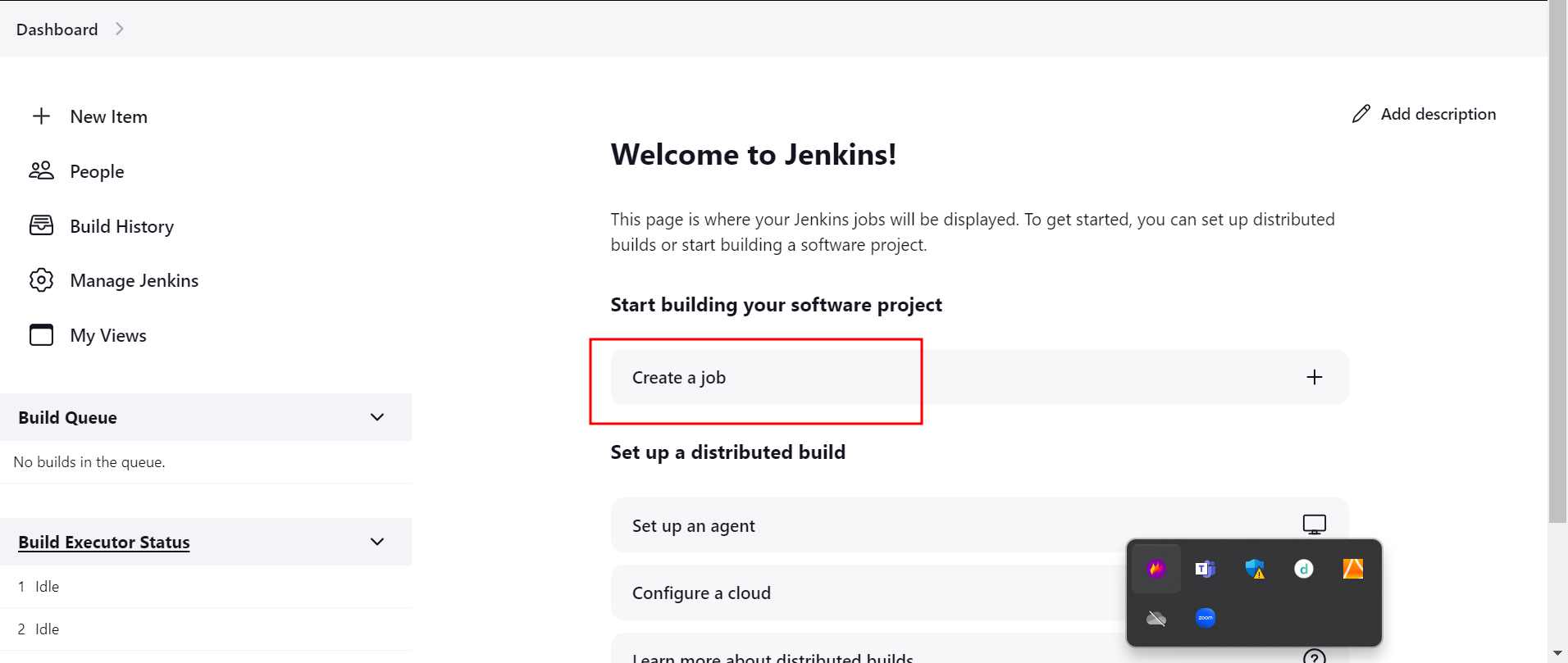


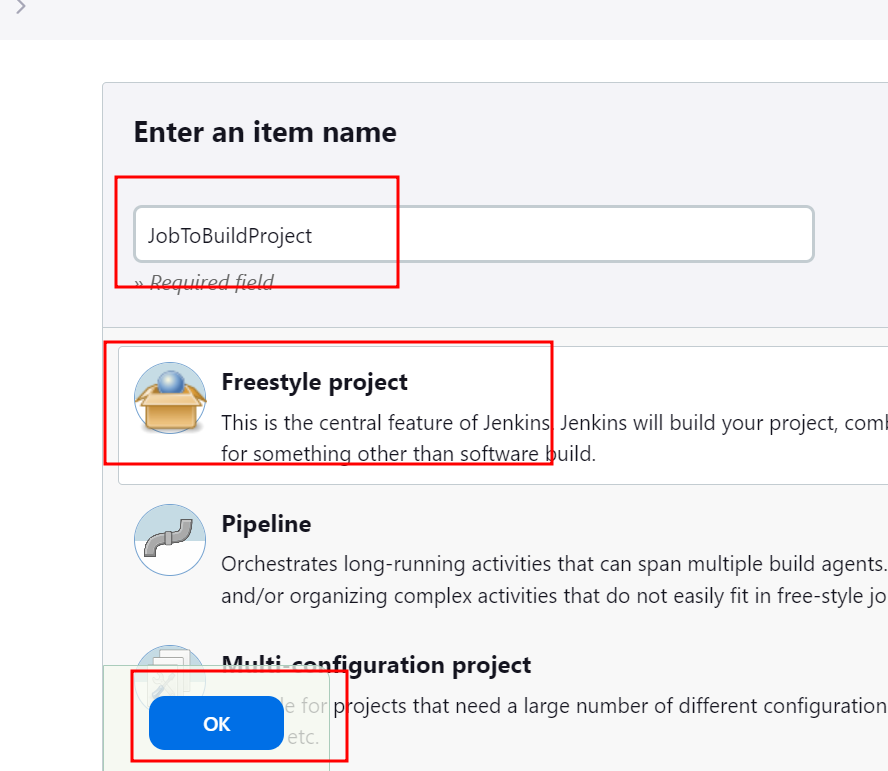


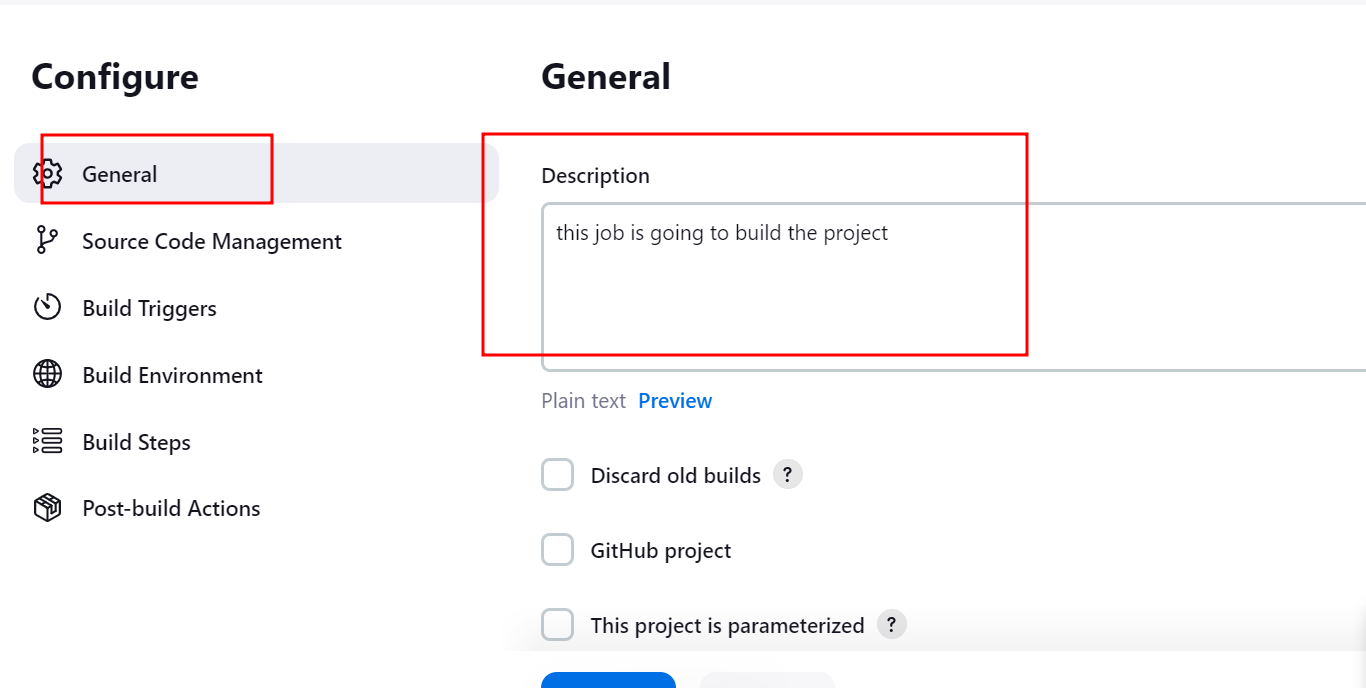


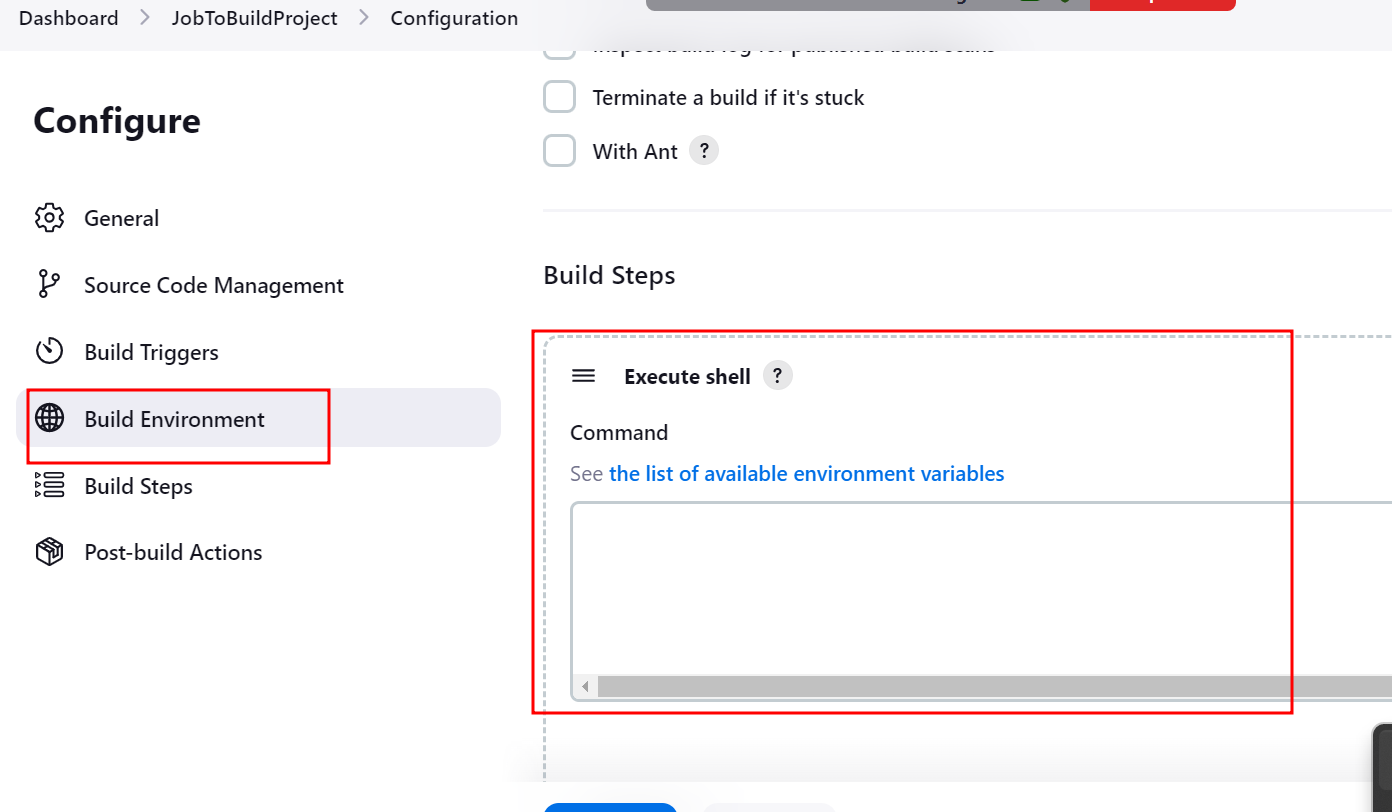
Now in Jenkin dashboard we will create job.

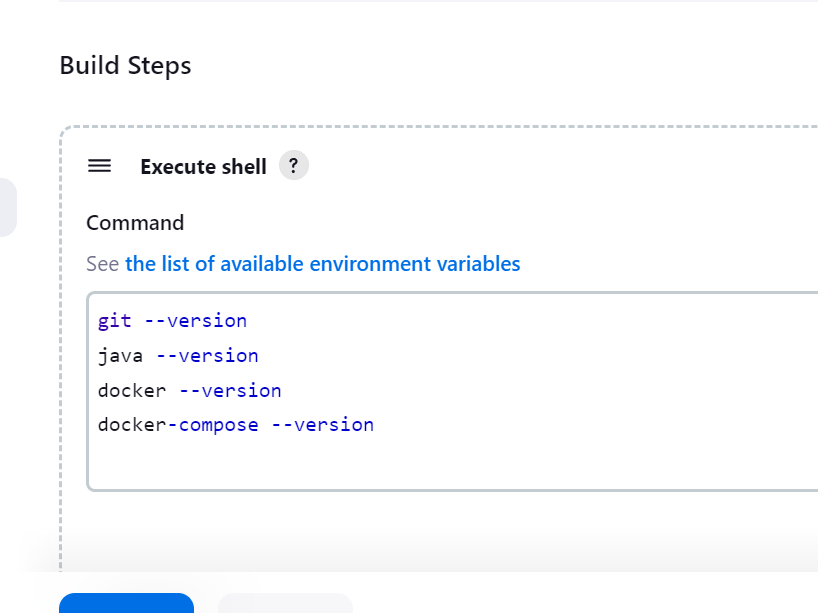
We will create job to check java –version, git version, docker version and docker-compose version



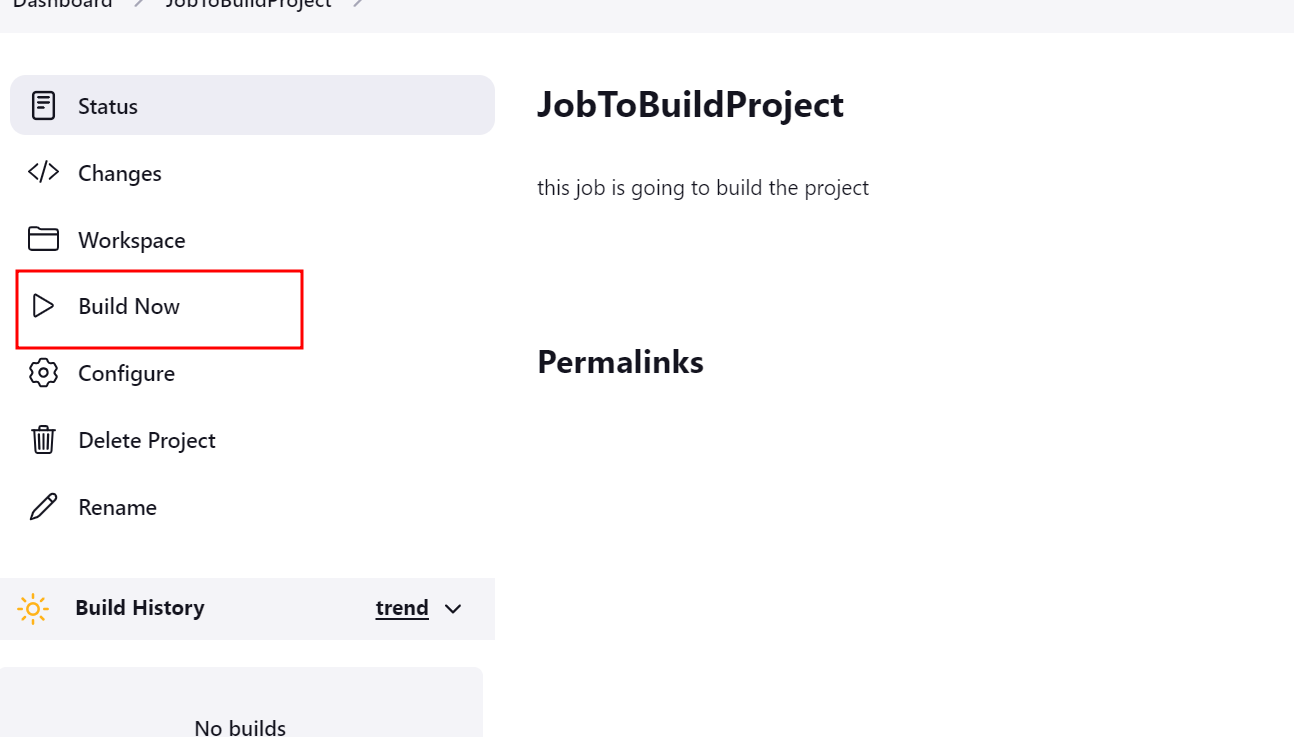




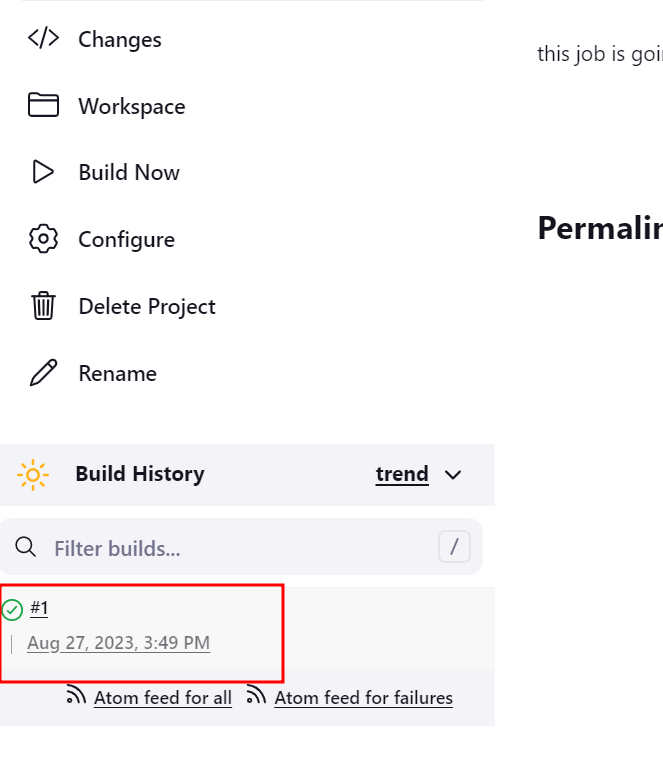


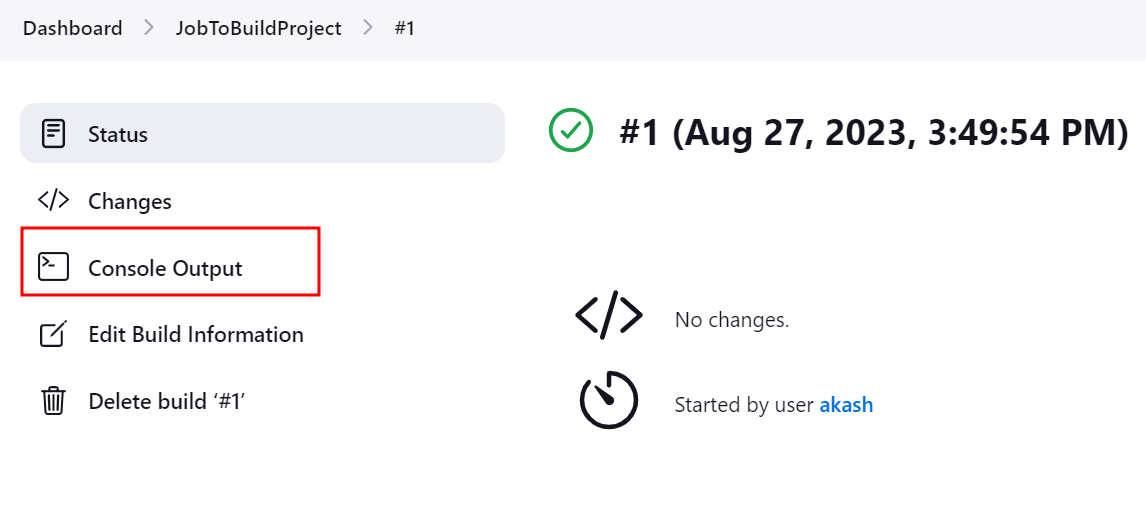


Click on apply and save



Then click on build button

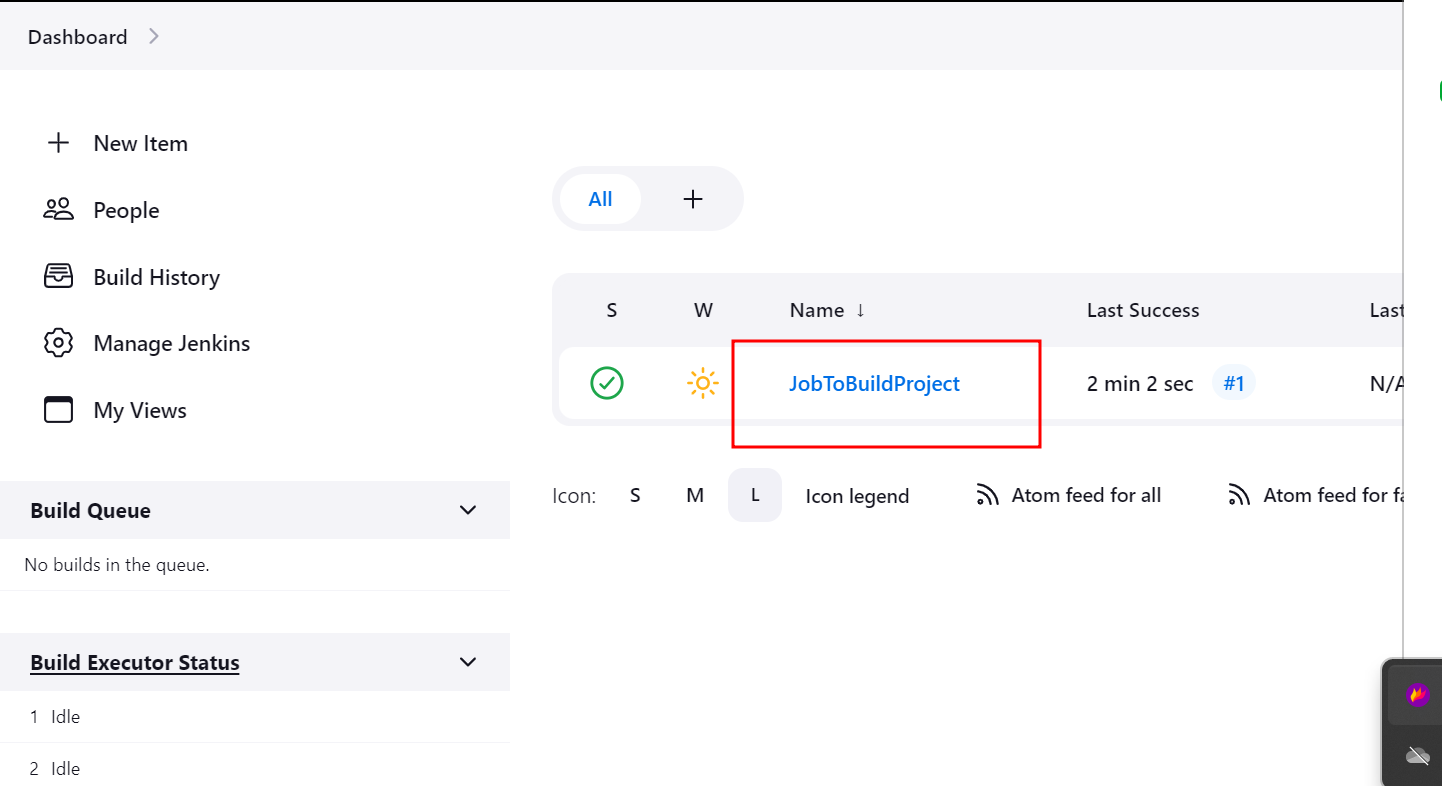




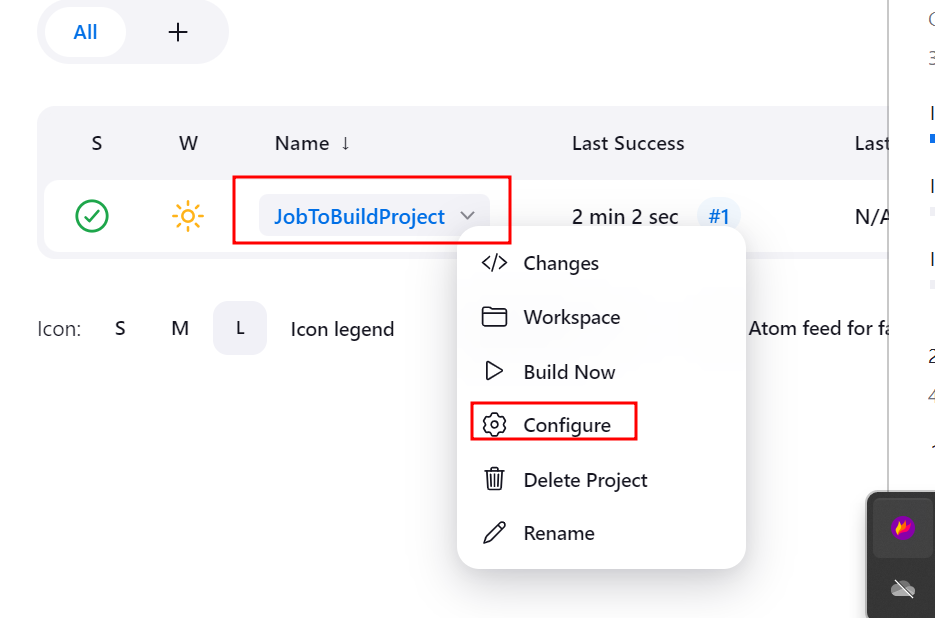


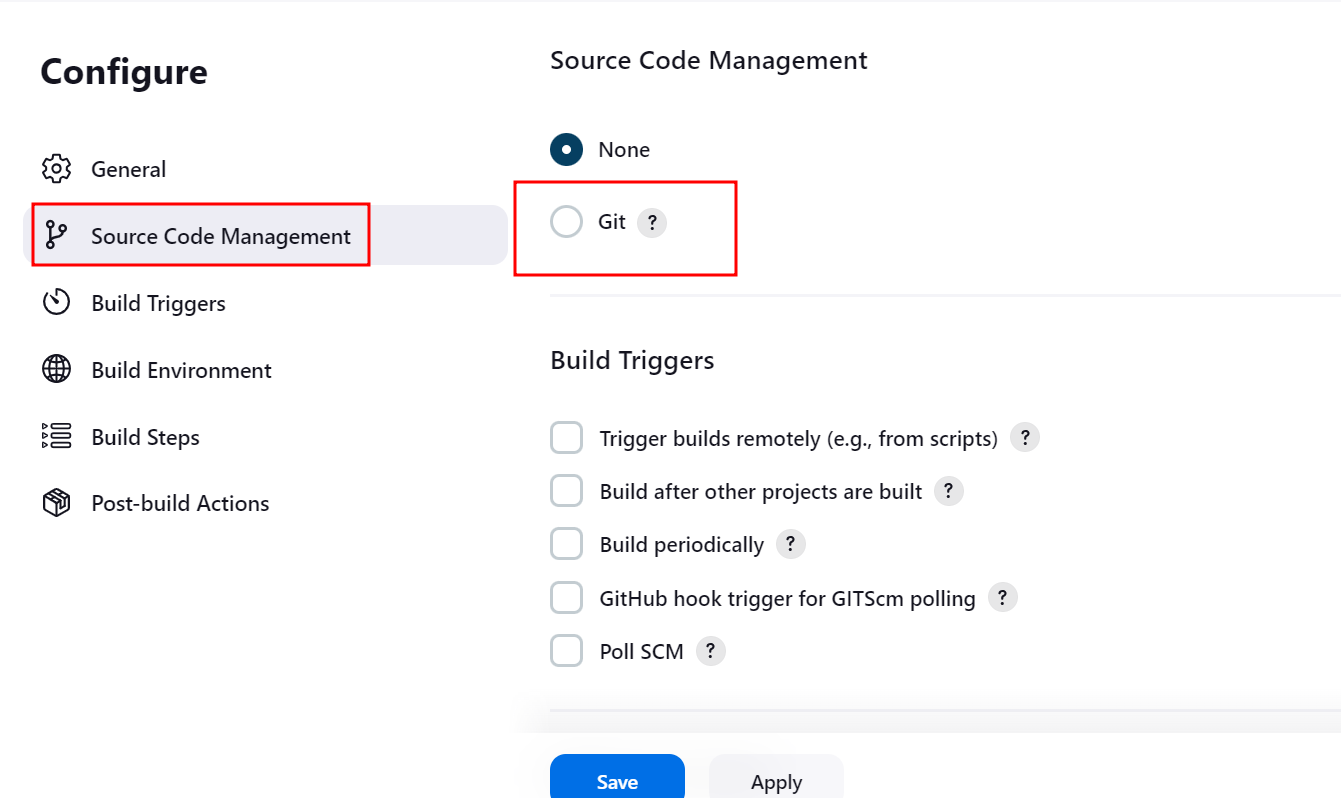
Now in same job we will pull the project from git

Please check on dashboard



Right click on job and select

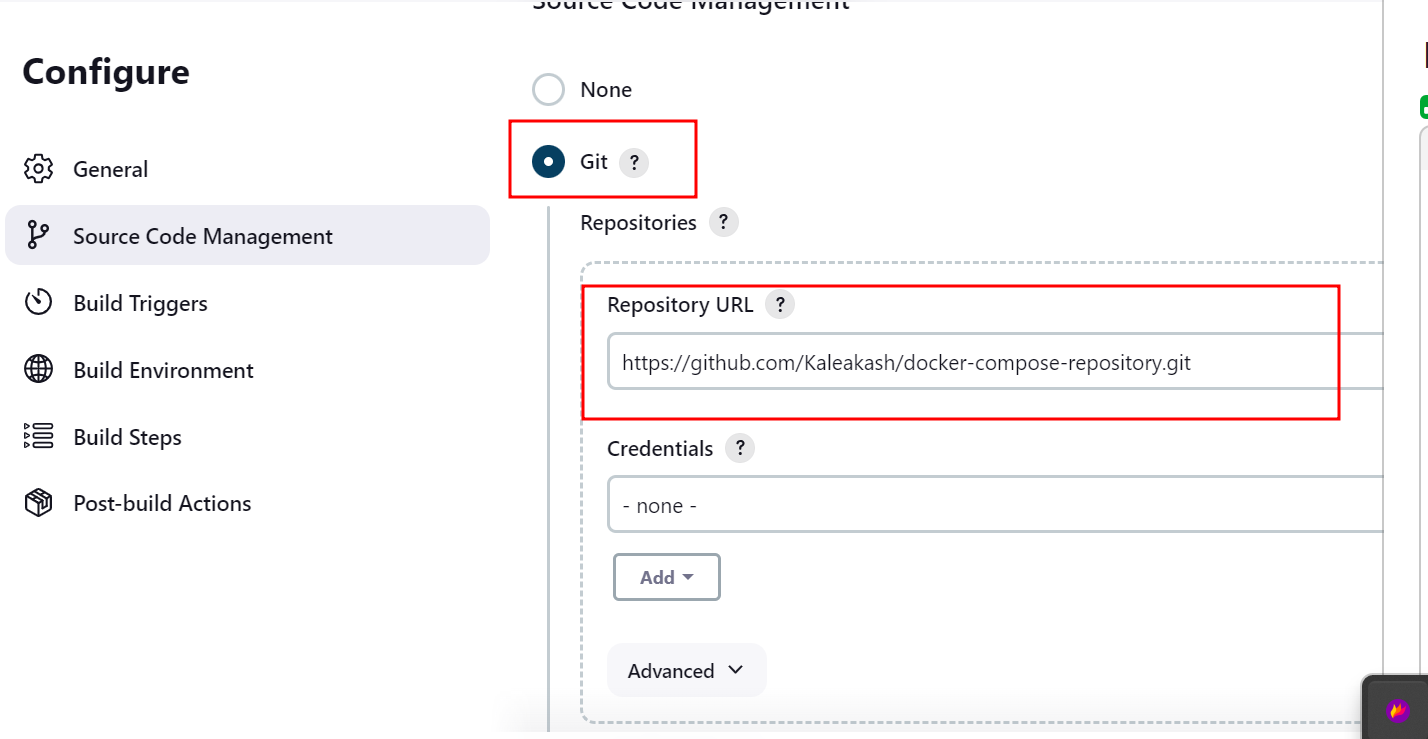




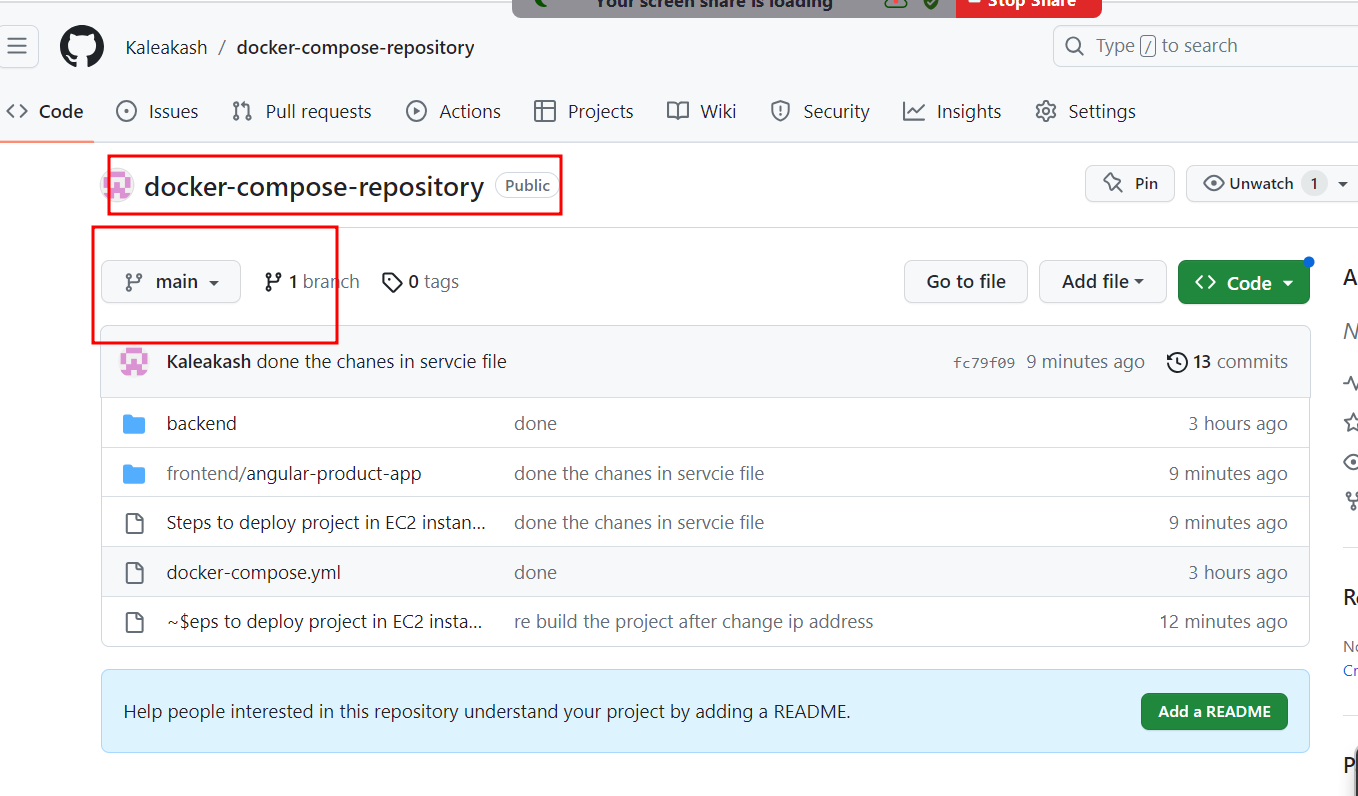
Please select git option

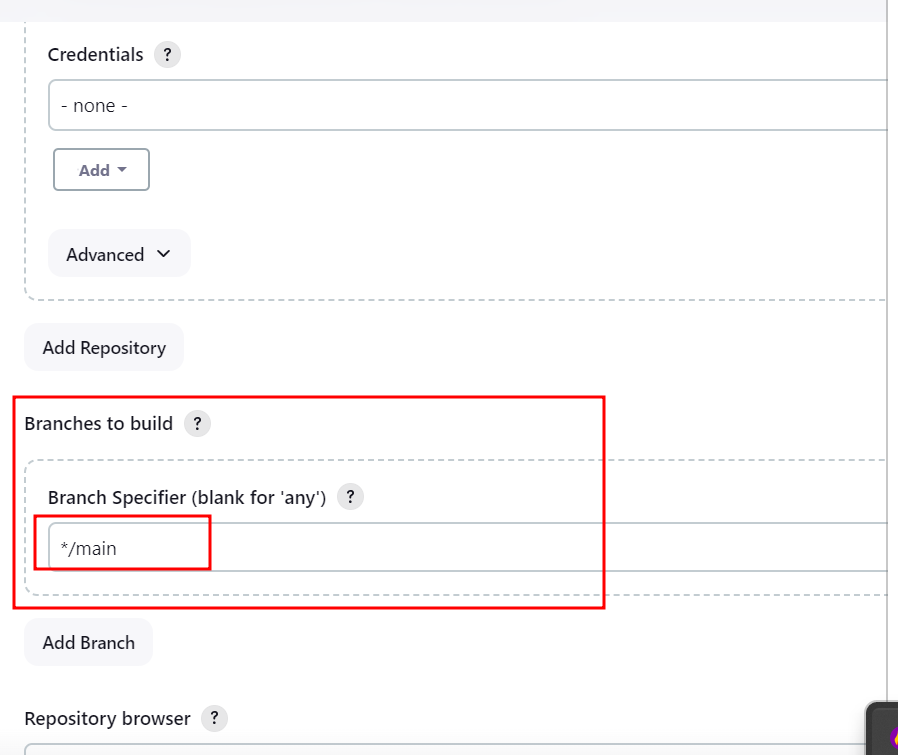
And write the repository as

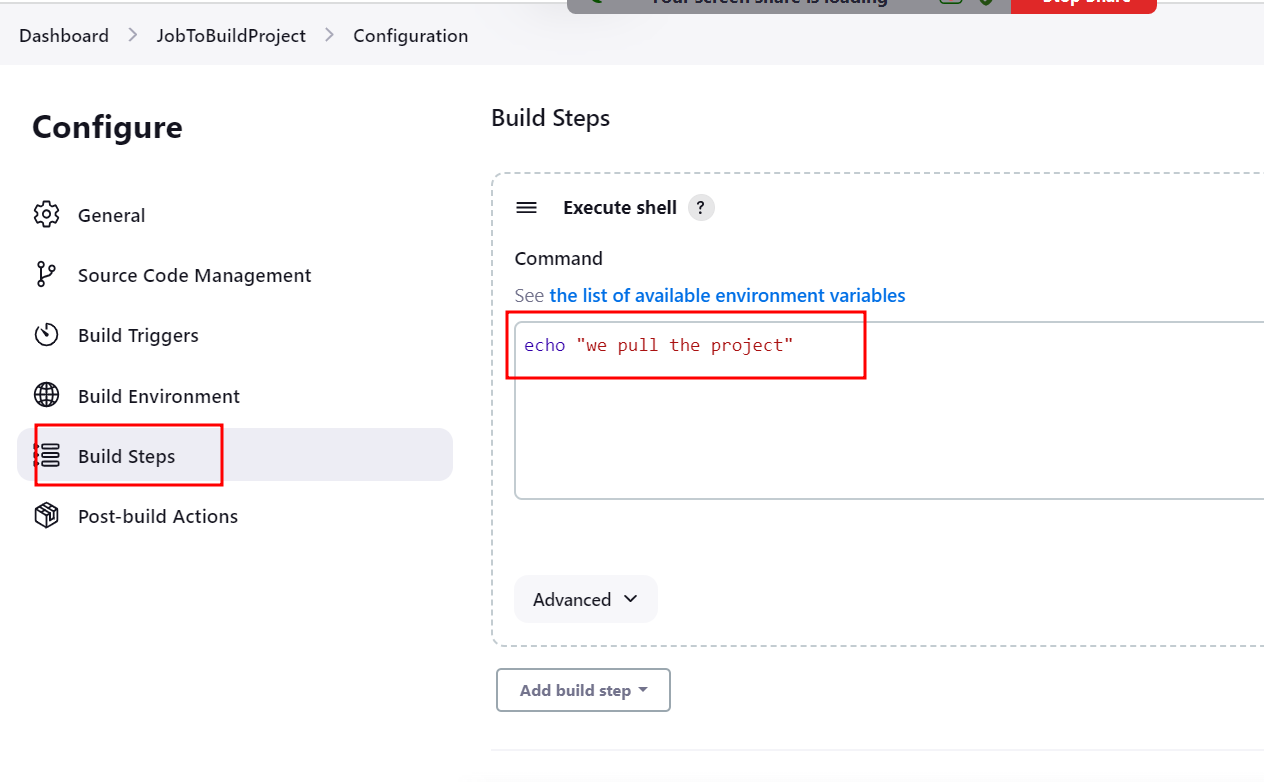
<https://github.com/Kaleakash/docker-compose-repository.git>



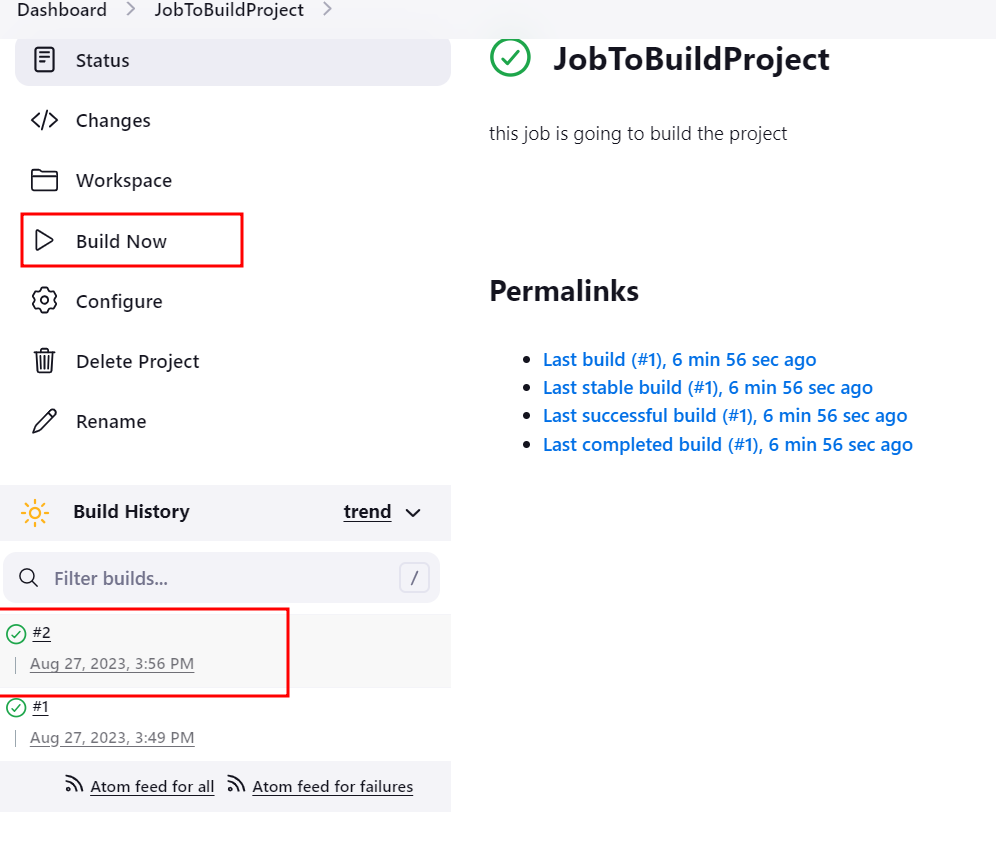
Then check the branch name must be same in git hub and in Jenkins

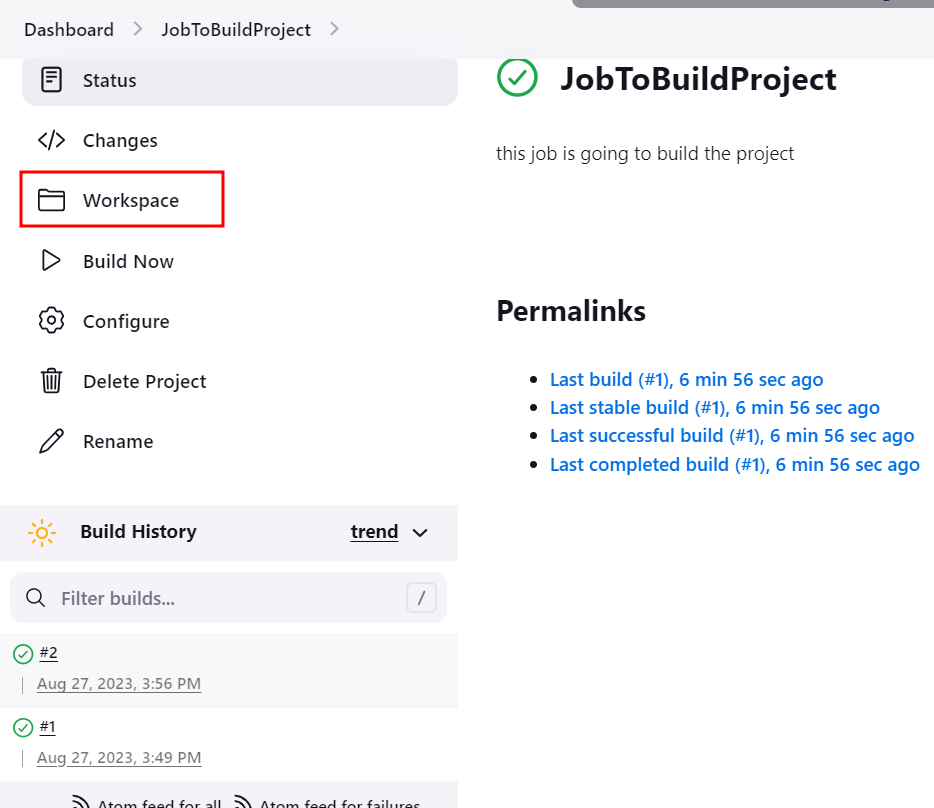




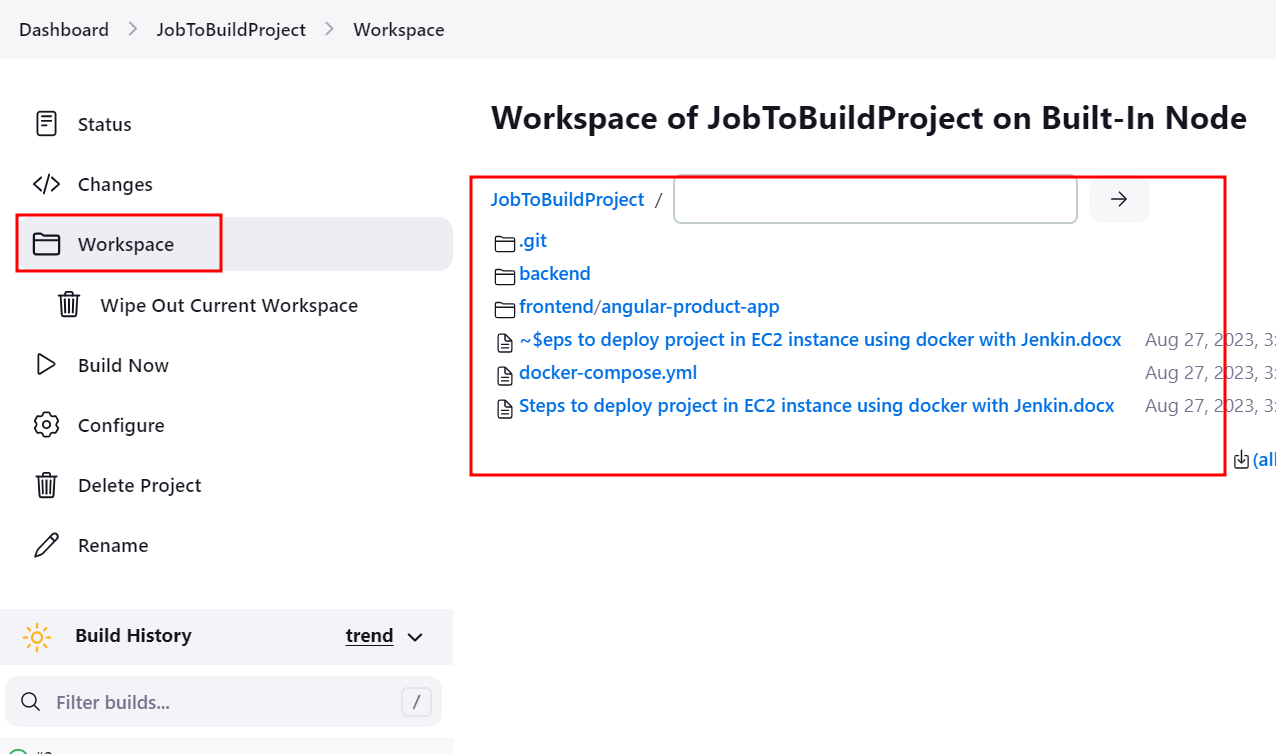


Now click on apply and save and build the project.

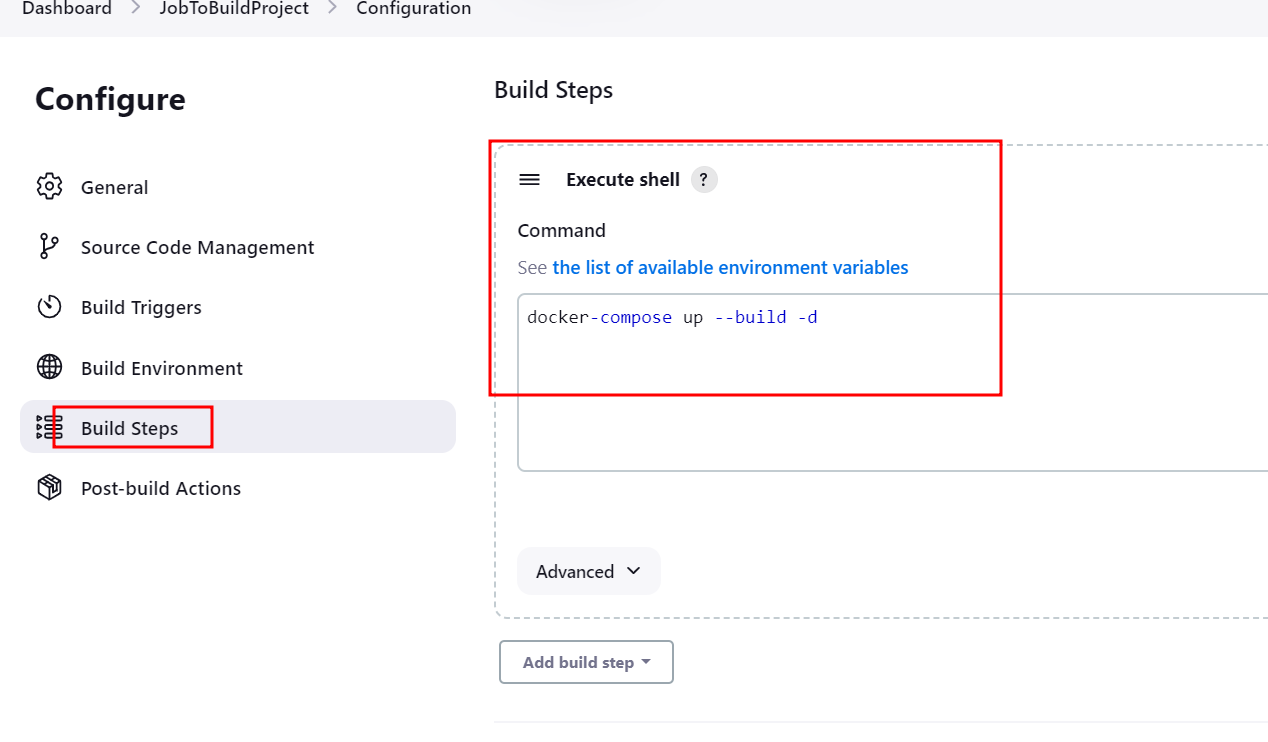




Please click on workspace to check project pull from git to Jenkin dashboard.



In we will run the docker-compose file to run three containers.



Here we are running docker-compose file to create images and run the container.

Save and apply and click build options.

