Course:Devops NAME:S.ANJANEYULU

Module:docker and docker hub EMAILID:ANJISINGAM@103GMAIL.COM Topic:sonar,nexus,tomcat,docker image Assignment no.4

Trainer name:Mr.Madhukar date of submission:12/01/202

BATCH NUMBER –116

1.sonarqube do code quality analysis

create and launch two instances

1.sonarqube 2.jenkins

1.sonarqube

sudo -i

apt update -y

apt install maven -y

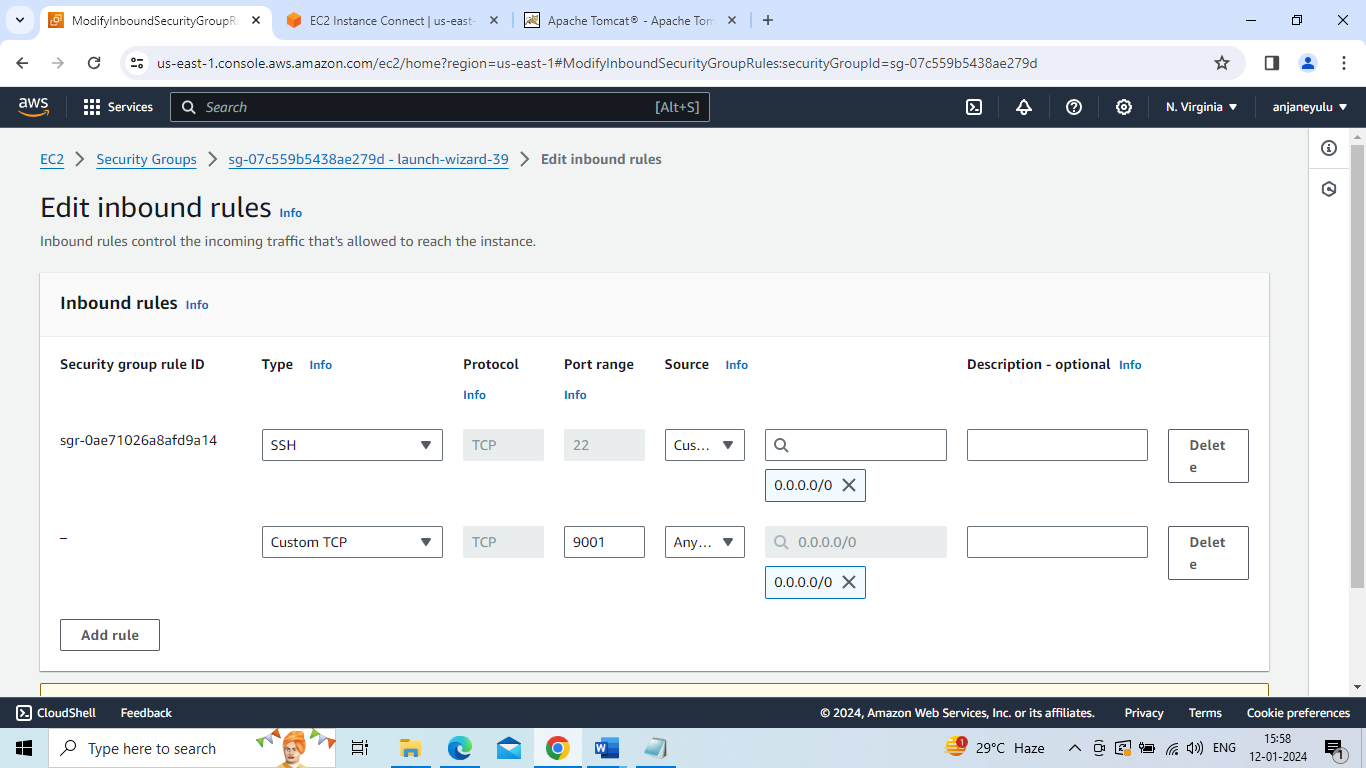
apt install openjdk-17-jdk -y

apt install docker.io -y

docker run --name sonar -d -p 9001:9000 sonarqube:latest

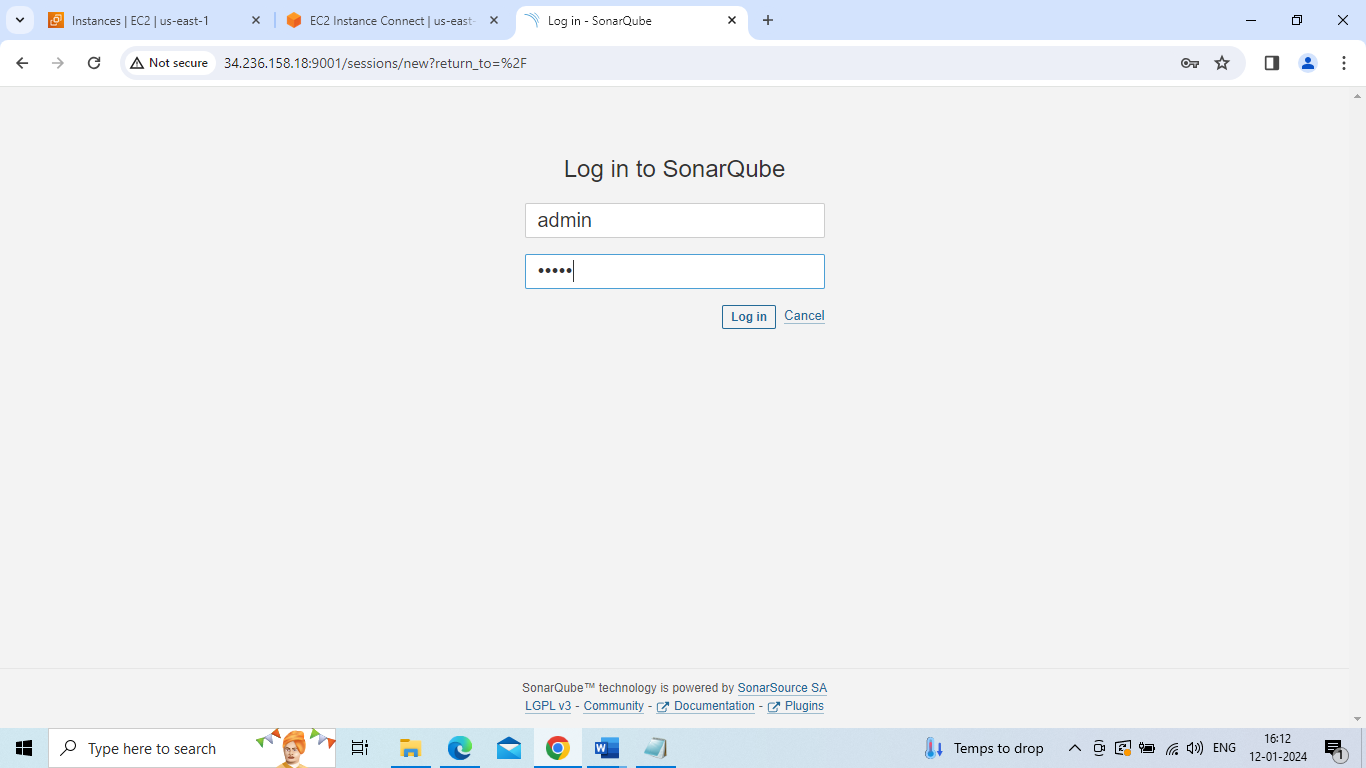
go to aws account-->security-->actions-->edit inbound rules --> add rule --> custom tcp,port number-9001,0.0.0.0/0

-->save rule



copy and paste pubile IP of sonarqube and port number --9001 in new page or browser

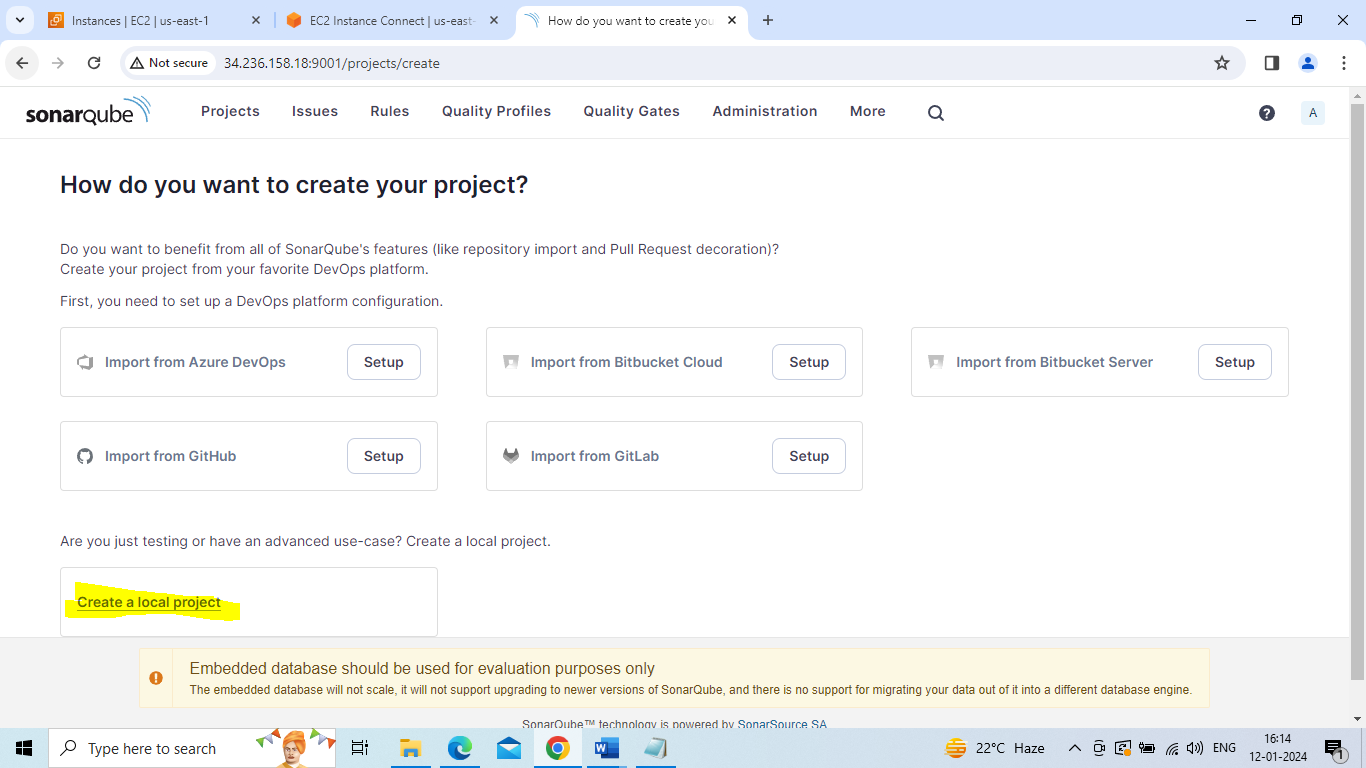
login to sonarqube using-->username-admin,password-admin



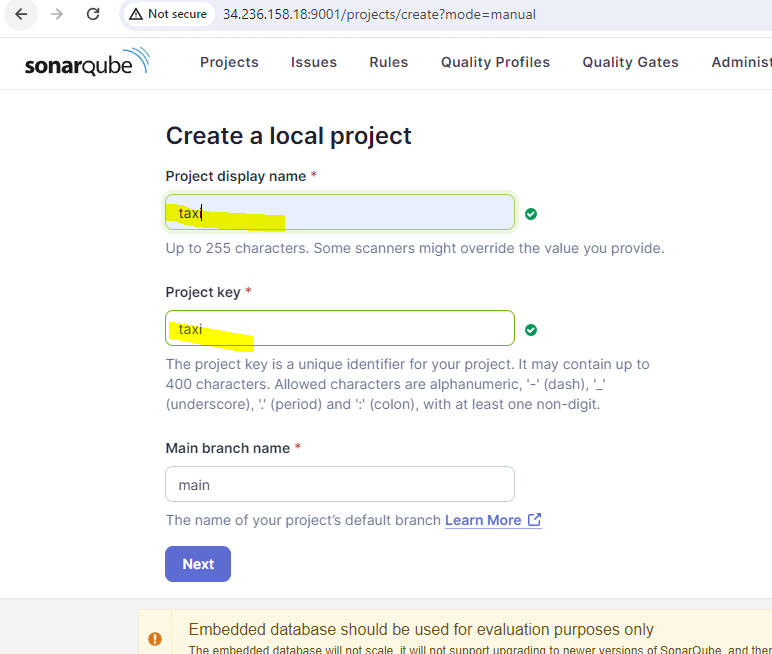
create local project with username-taxi-->define-->previous version -->create-->generate-->maven-->copy that data -->

paste in notepad and make a single line

Create local project

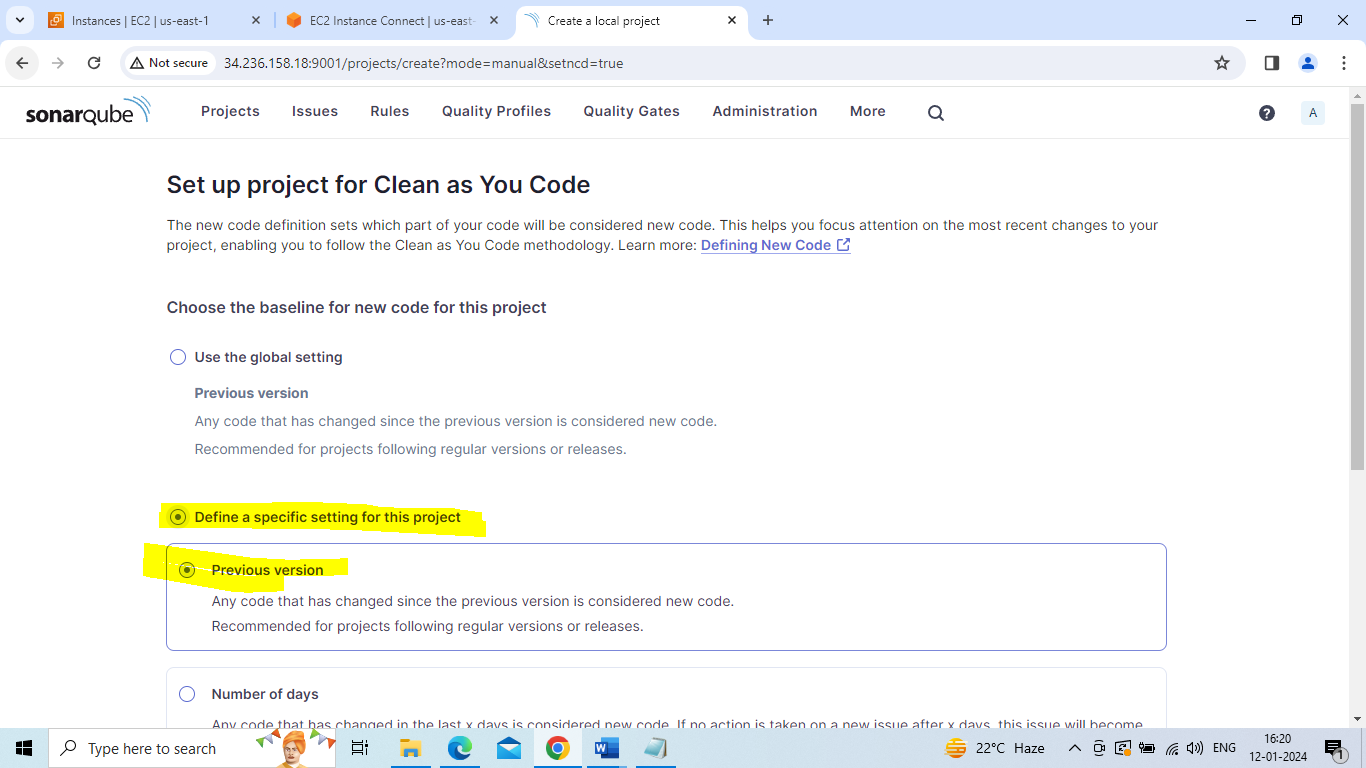


Project name –taxi



**Define a specific setting for this project**

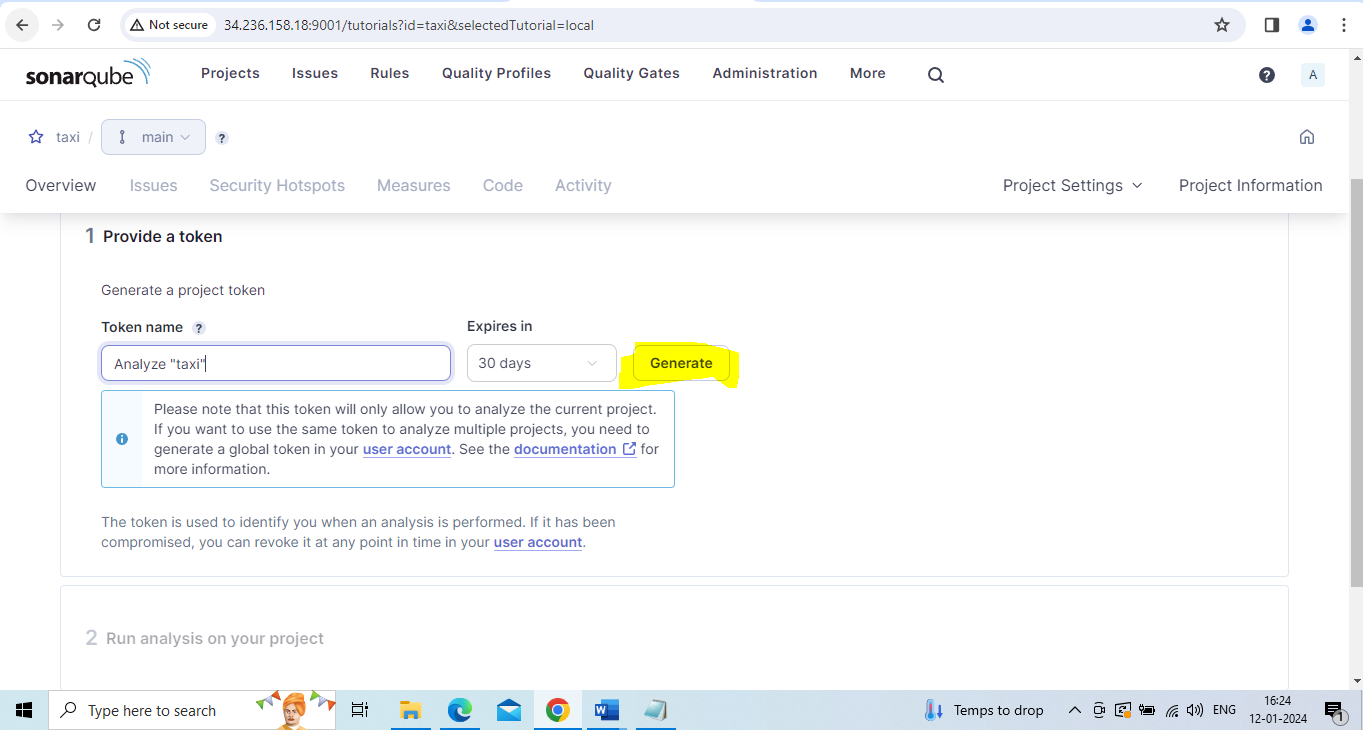
**Previous version**



Create project

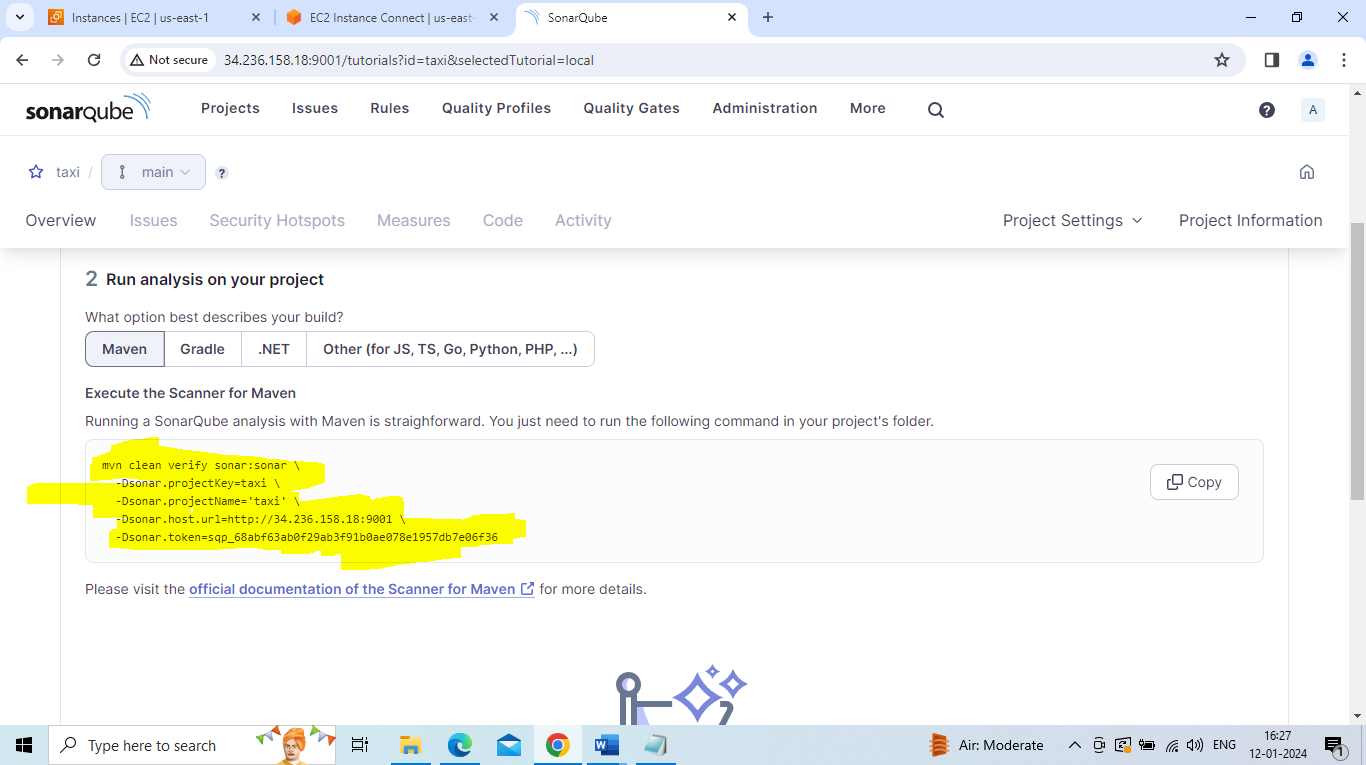
Locally

Generate



Continue

Select maven



clean verify sonar:sonar -Dsonar.projectKey=taxi -Dsonar.projectName='taxi

-Dsonar.host.url=http://34.236.158.18:9001

-Dsonar.token=sqp\_68abf63ab0f29ab3f91b0ae078e1957db7e06f36

2.jenkins

sudo -i

apt update -y

apt install maven -y

apt install openjdk-11-jdk -y

sudo wget -O /usr/share/keyrings/jenkins-keyring.asc

https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key

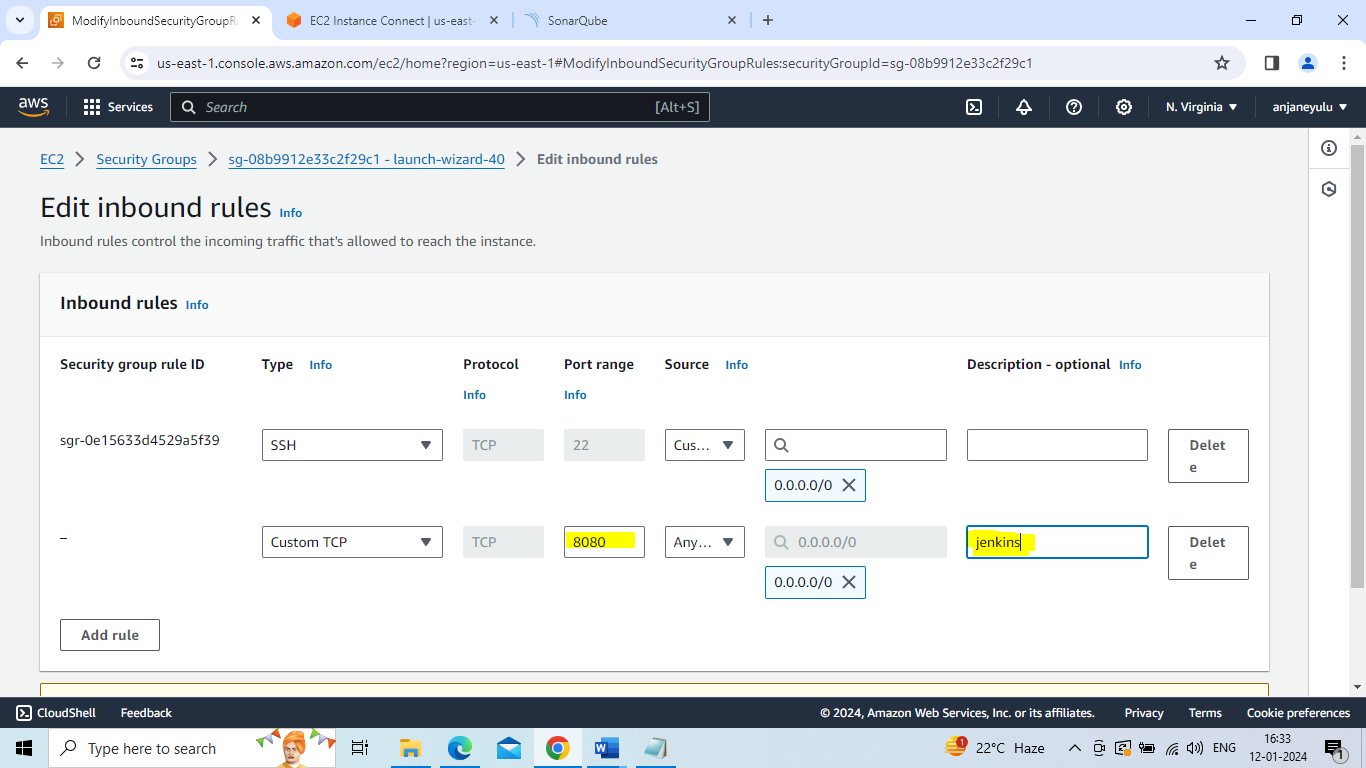
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian-stable binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null

sudo apt-get update

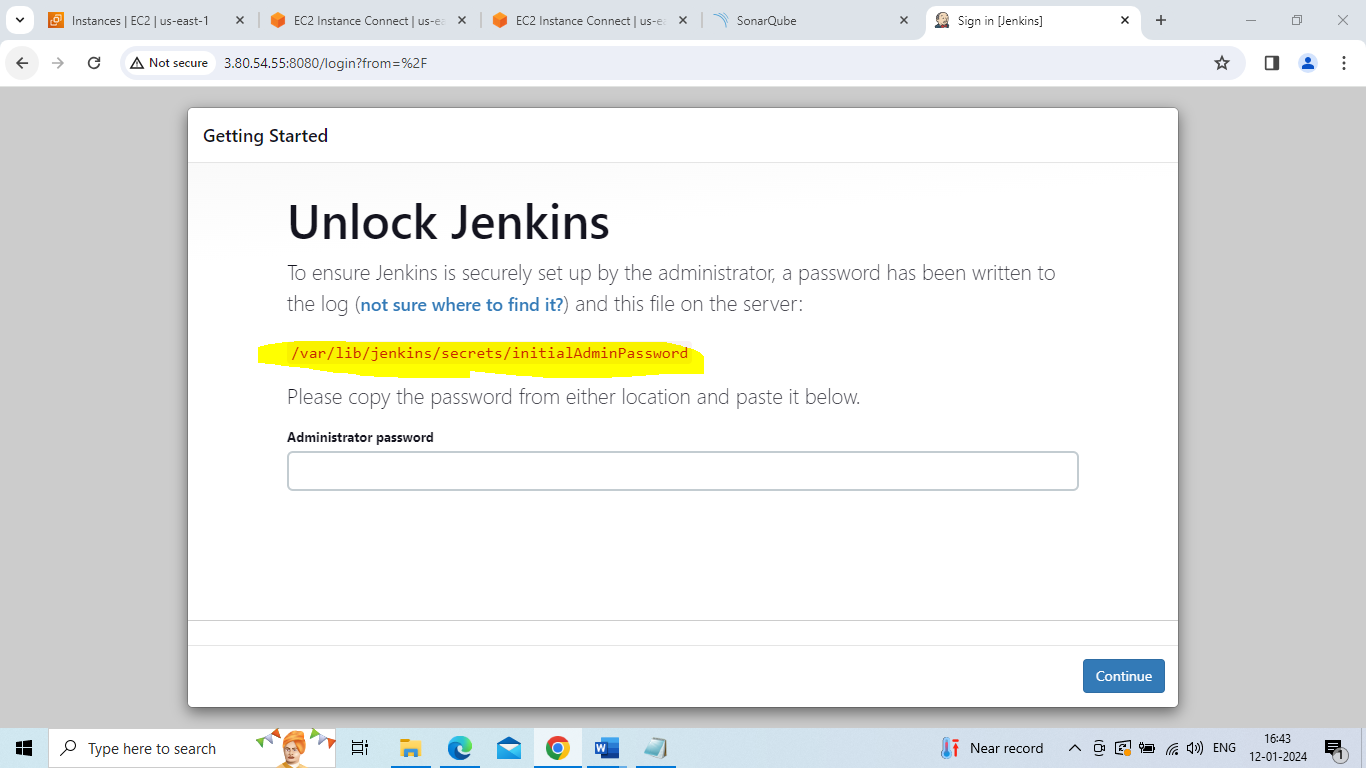
sudo apt-get install jenkins

go to aws account-->security-->actions-->edit inbound rules --> add rule --> custom tcp,port number-9001,0.0.0.0/0

-->save rule



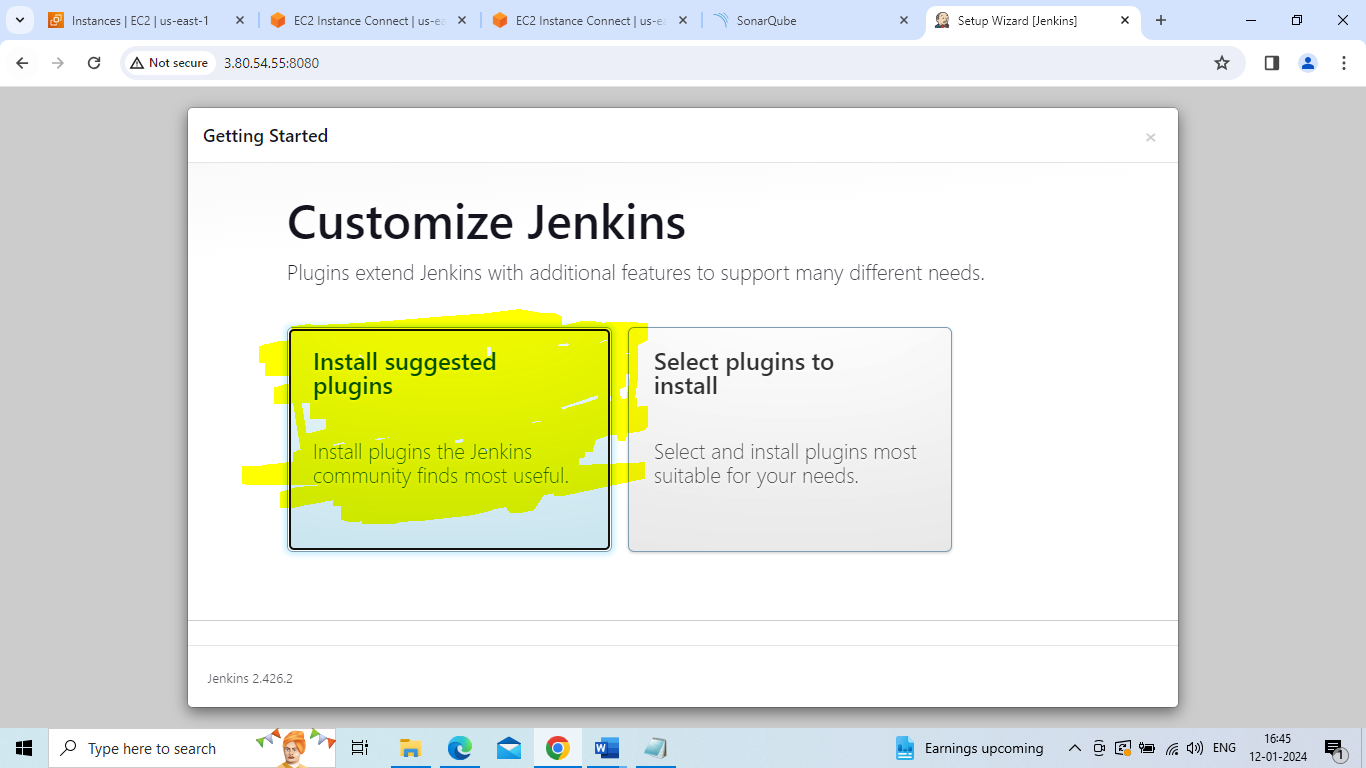
copy and paste pubile IP of jenkins and port number --8080 in new page or browser



Cat /var/lib/jenkins/secrets/initialAdminPassword

login to jenkins with initial password

Install suggested plugins



Login with your details

Start using jenkins

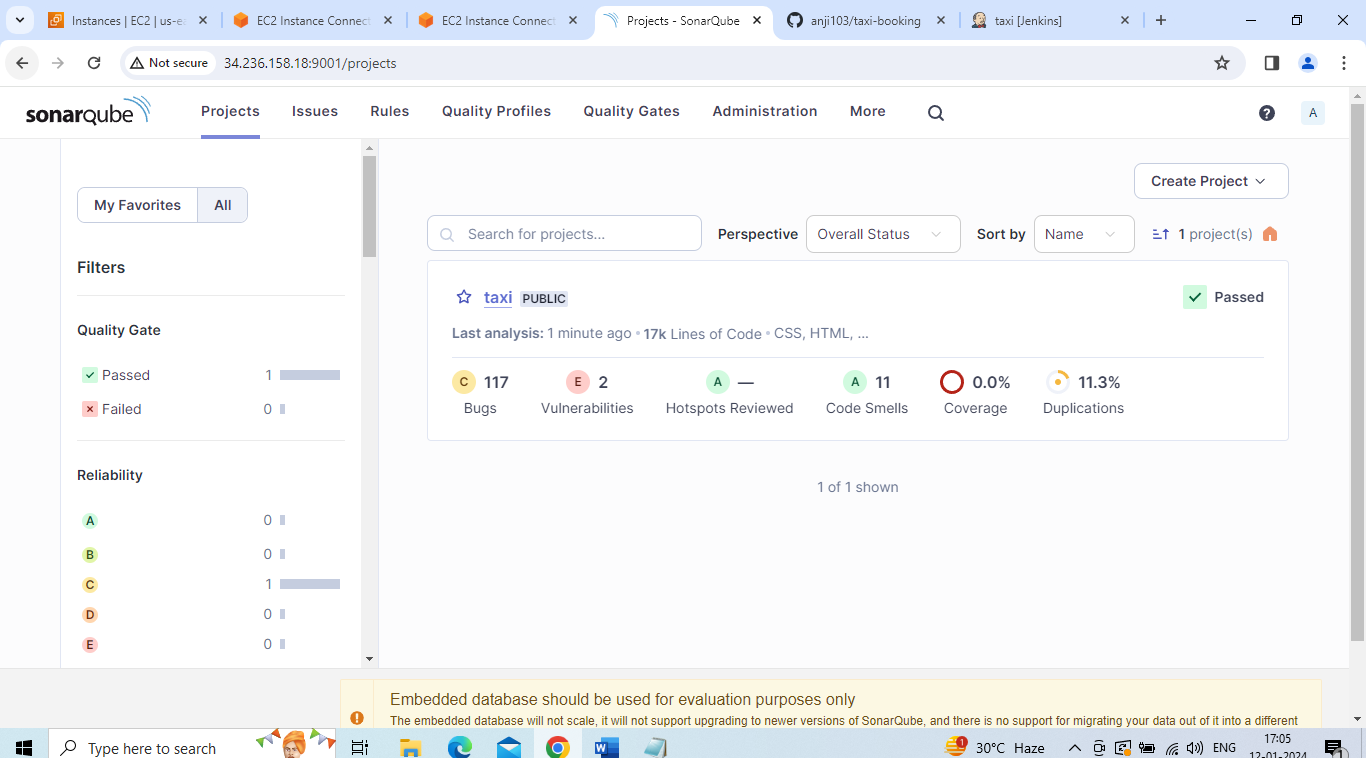
create one job with name—sonarqube(freestyle)

paste github link of taxi-booking in git repository

build step-->invoke top level maven target-->goals-->package

build step-->invoke top level maven target-->goals-->paste the notepad data-->apply and save

build now and after then auto-matically code quality done of taxi-booking in sonarqube



Deploy in tomcat in dockerjenkin

create and launch two instances -

1.tomcat 2.jenkins

1.tomcat

sudo -i(convert to root user)

apt update -y

apt install maven -y

apt install openjdk-17-jdk -y

go to google search --> tomcat 9 dowmload --->tar.zp(copy this link)

Cd /opt

wget paste https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.85/bin/apache-tomcat-9.0.85.tar.gz

Tar –xvzf apache-tomcat-9.0.85.tar.gz

Ls

apache-tomcat-9.0.85 apache-tomcat-9.0.85.tar.gz

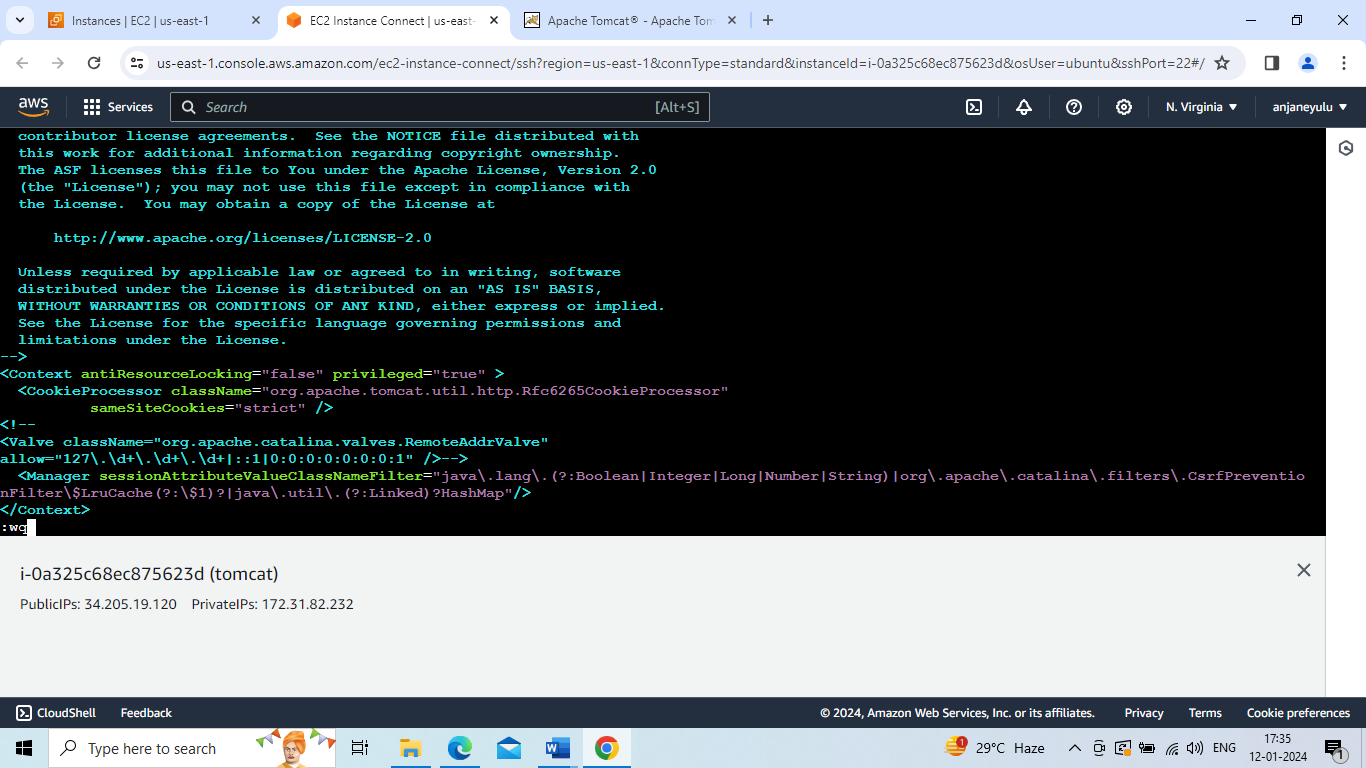
Mv apache-tomcat-9.0.85 tomcat

cd /opt/tomcat/webapps/host-manager/META-INF --->vi context.xml

<!--

vlave

allow >-->



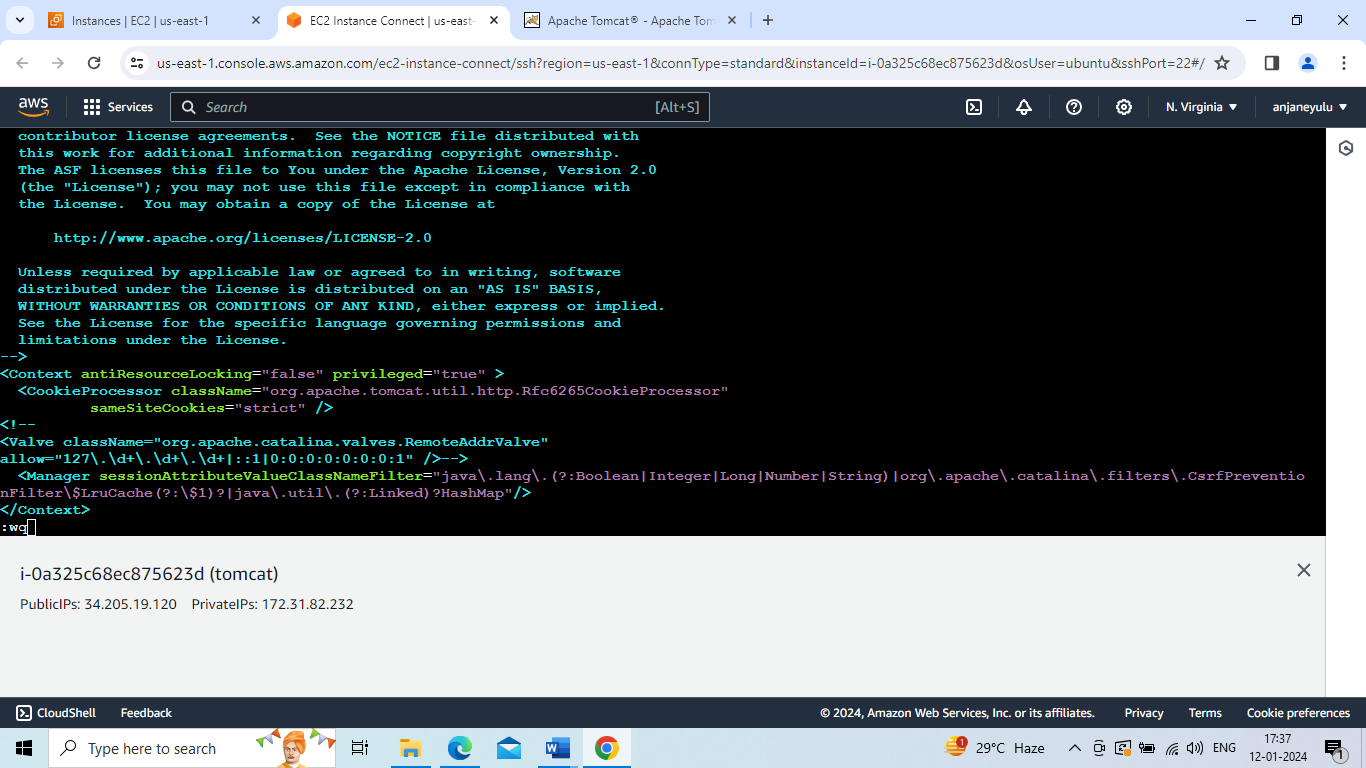
cd

cd /opt/tomcat/webapps/manager/META-INF --->vi context.xml

<!--

vlave

allow >-->



cd

cd /opt/tomcat/conf --->vi tomcat-users.xml

<role rolename="manager-gui"/>

<role rolename="manager-script"/>

<role rolename="manager-jmx"/>

<role rolename="manager-status"/>

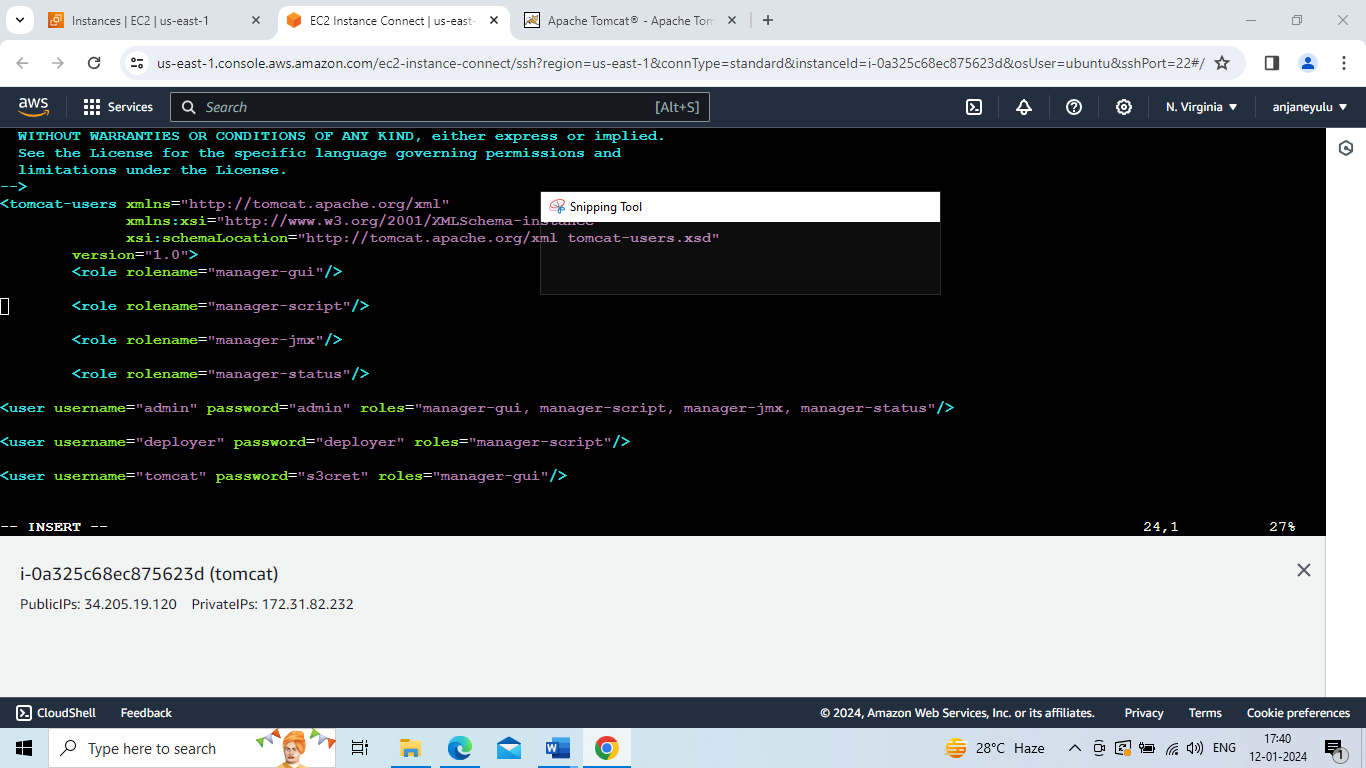
<user username="admin" password="admin" roles="manager-gui, manager-script, manager-jmx, manager-status"/>

<user username="deployer" password="deployer" roles="manager-script"/>

<user username="tomcat" password="s3cret" roles="manager-gui"/>

Go to aws account --> security--> actions--> edit inbound rules--> add rule --> port number : 8080, custom tcp,

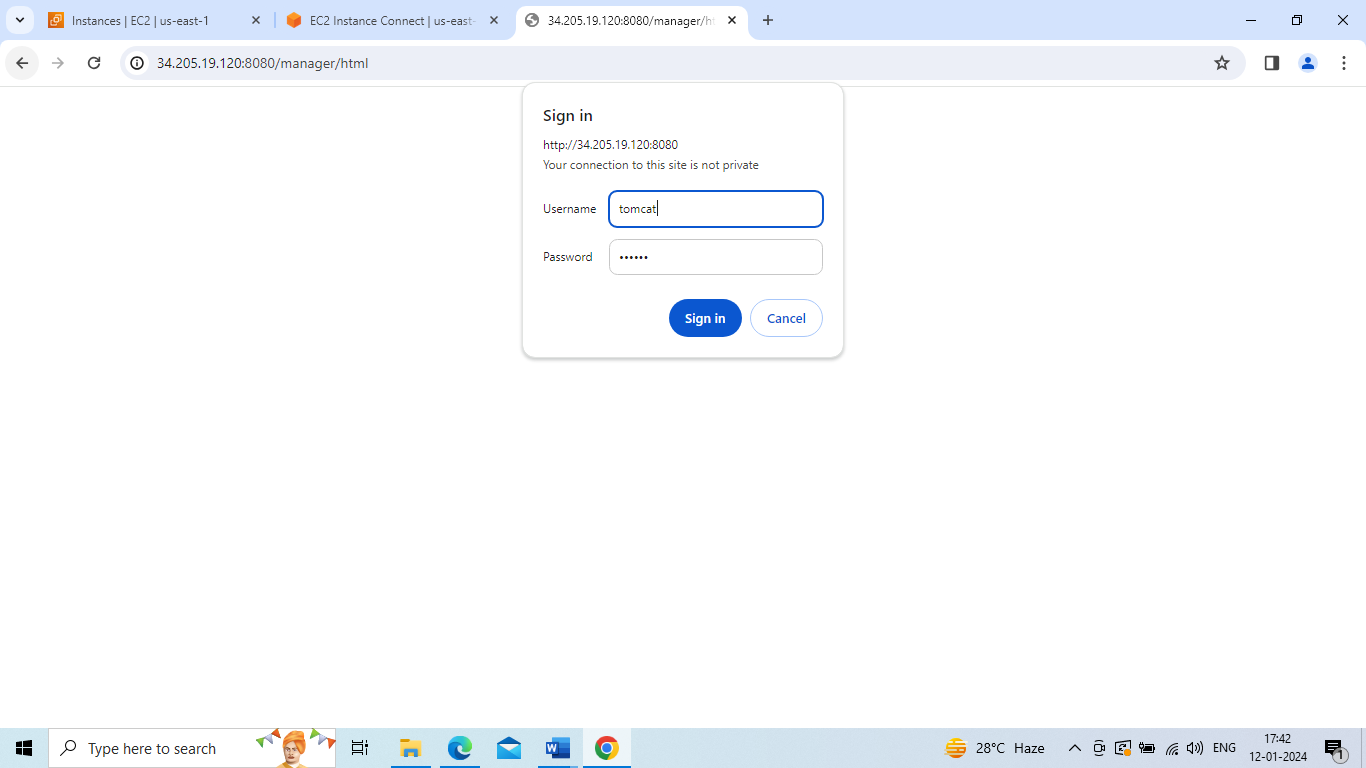
0.0.0.0/0 --> save rule

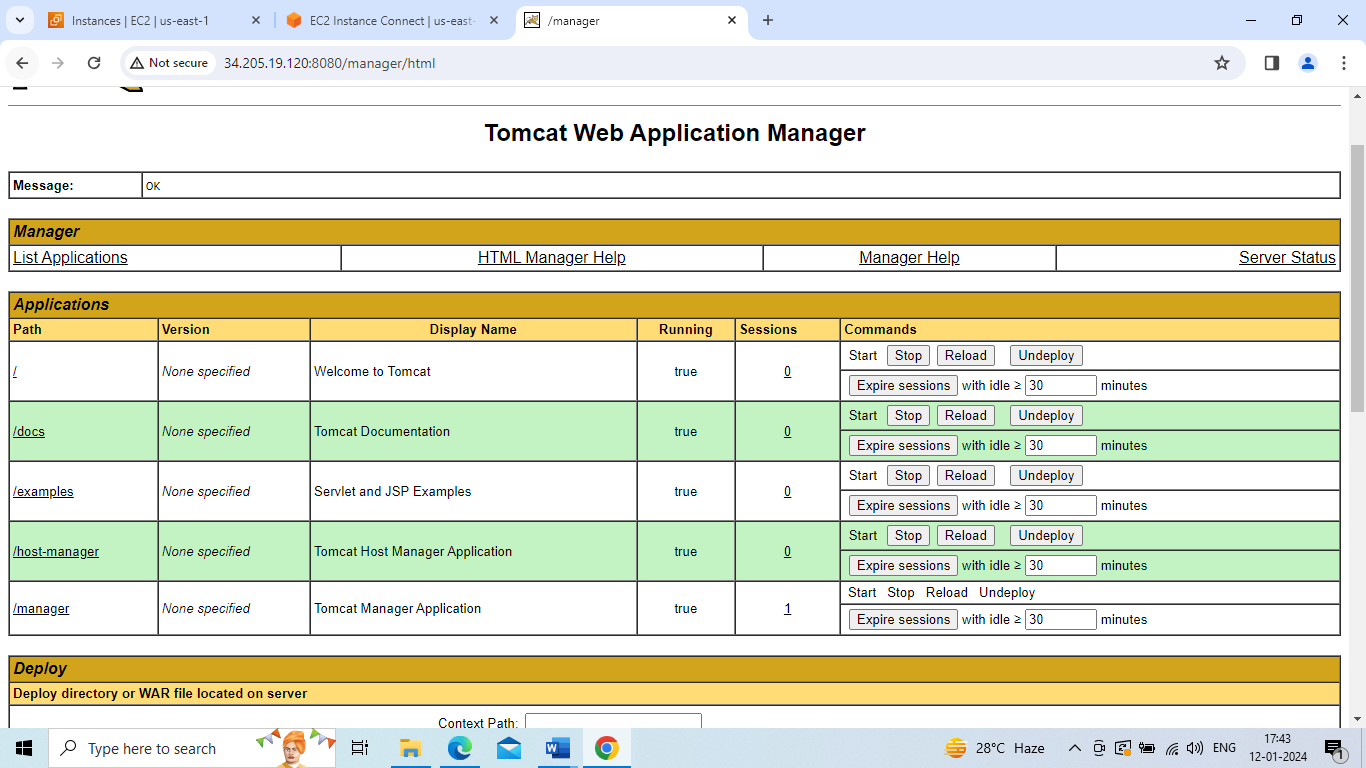


cd

cd /opt/tomcat/bin --> ./shutdown.sh --> ./startup.sh

Login to tomcat





2.jenkins

sudo -i(convert to root user)

apt update -y

apt install maven -y

apt install openjdk-17-jdk -y

sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \

https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key

echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \

https://pkg.jenkins.io/debian-stable binary/ | sudo tee \

/etc/apt/sources.list.d/jenkins.list > /dev/null

sudo apt-get update

sudo apt-get install jenkins

go aws instance jenkins--> security-->actions--> edit inbound rules--> add rule --> port number : 8080, custom tcp,

0.0.0.0/0 --> save rule

copy and paste pubile IP of jenkins and port number 8080 --> press enter --> install and login to jenkins

manage jenkins -->plugin --> deploy to container -->install it

crate a new job with deploytaxibookingtotomcat --> freestyle -->ok

scm-->git --> paste link from github taxi-booking

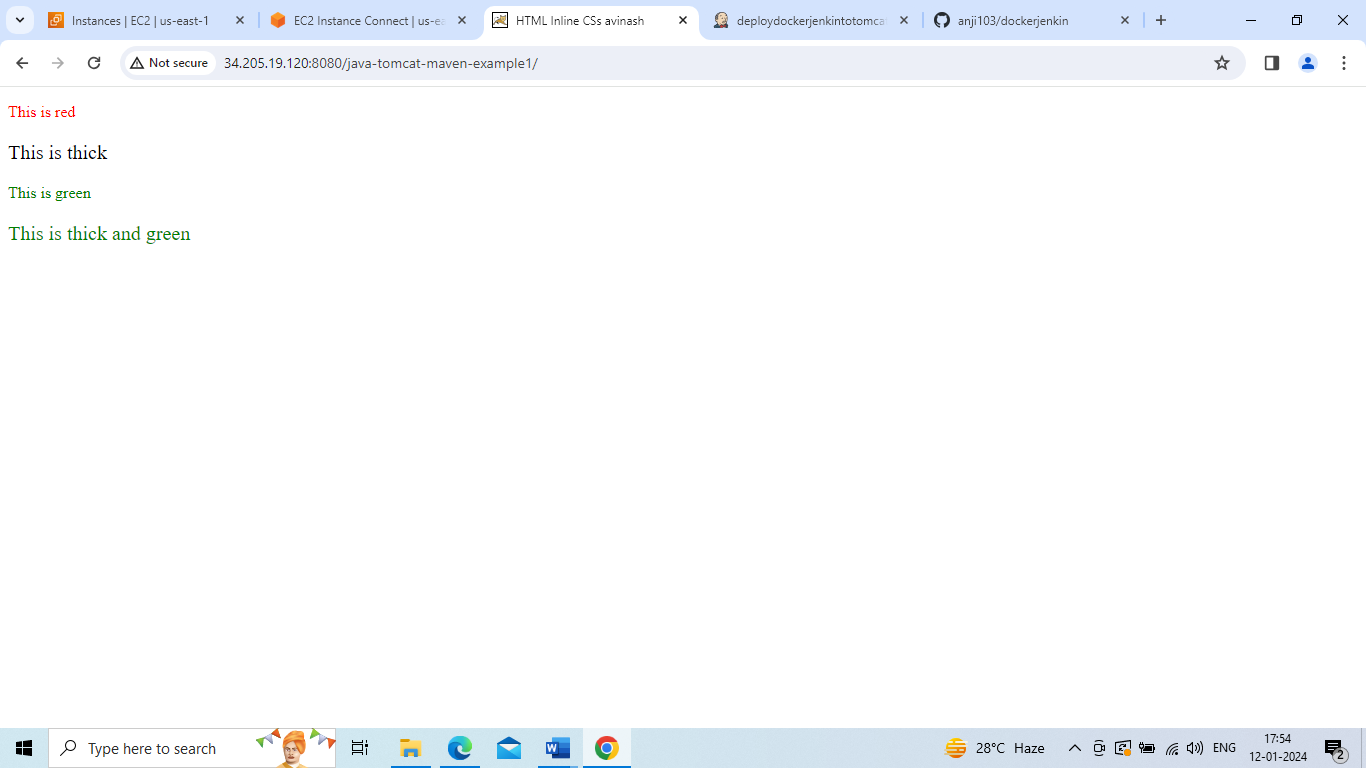
branch -master(in what you have branch in github-taxi-booking that branch name only)

build step -->invoke maven target --> goals --> package

post build step --> deploy to container --> war/ear file -->\*\*/\*war--> tomcat 9x container -->tomcat url --> apply and save

build now

After completing build SUCCESS refresh tomcat web page then automatically dockerjenkin deploy success



Build image and push image to docker hub

sudo -i(convert to root user)

apt update -y

apt install maven -y

apt install openjdk-17-jdk –y

Apt install docker.io -y

Vi Dockerfile

FROM nginx:latest

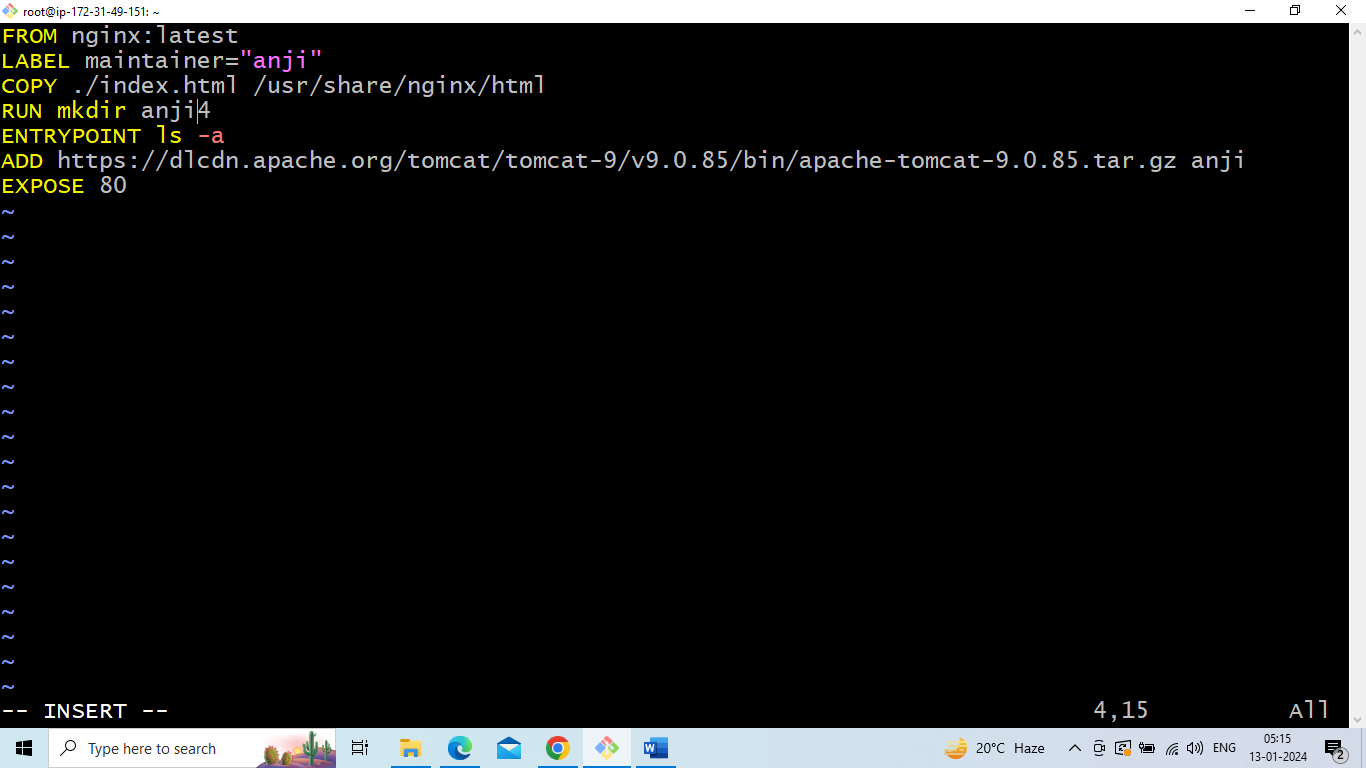
LABEL maintainer=”anji4”

COPY ./index.html /usr/share/nginx/html

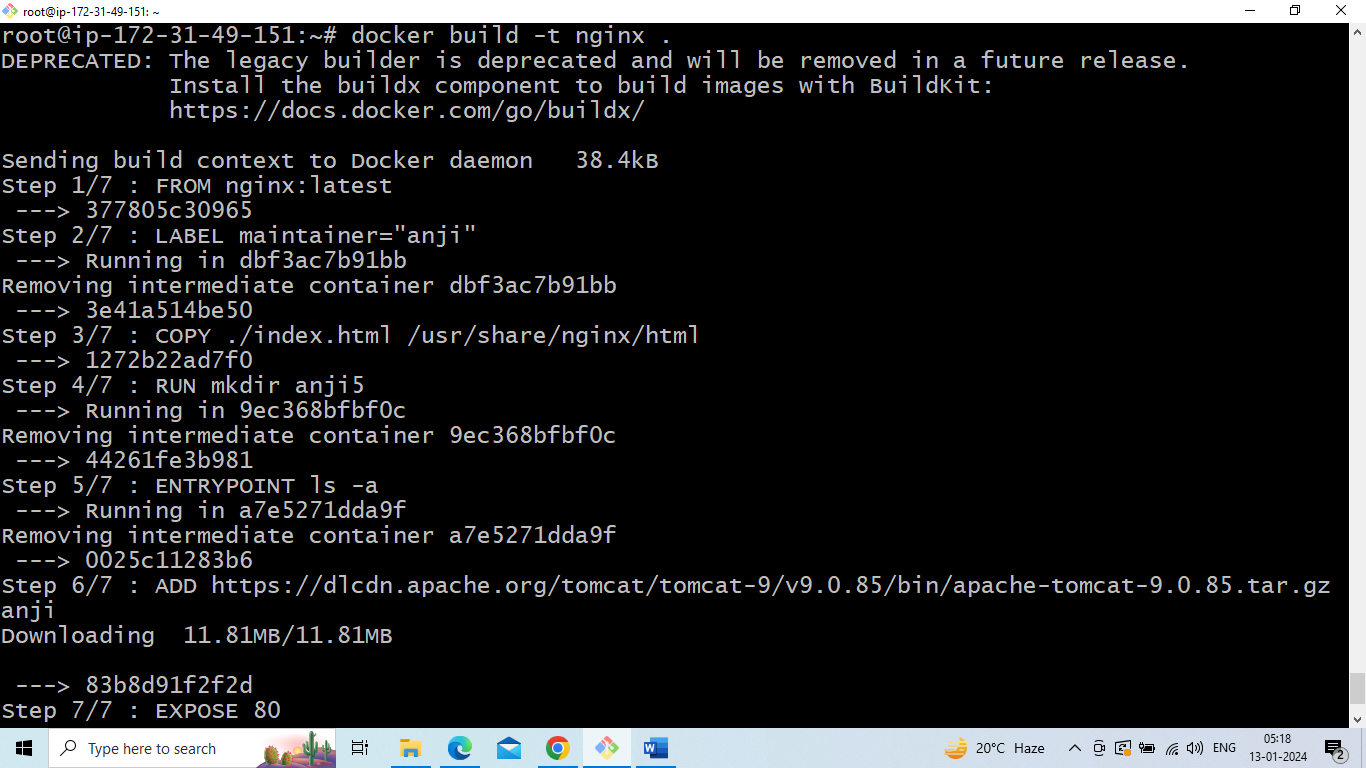
RUN mkdir anji

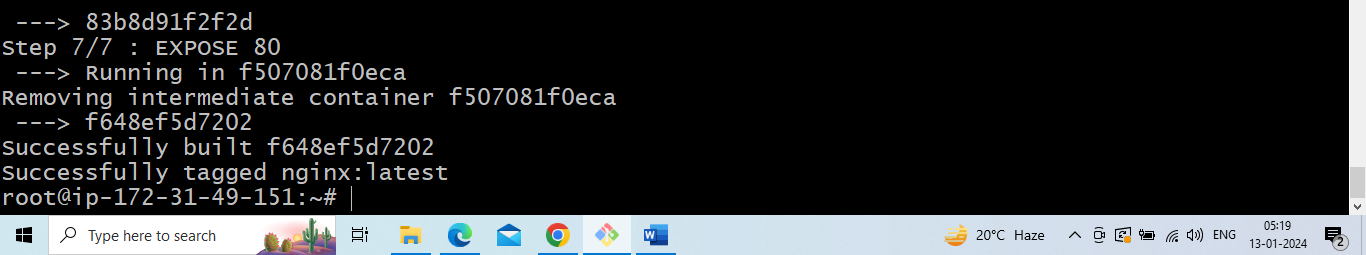
CMD ls –a

EXPOSE 80

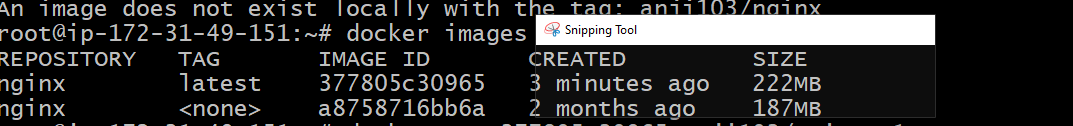


Docker build –t nginx .





Docker images



Syntax for docker push

Docker tag <image-id> dockerhubname/imagename:tagname

Docker push dockerhubname/imagename:tagnmae

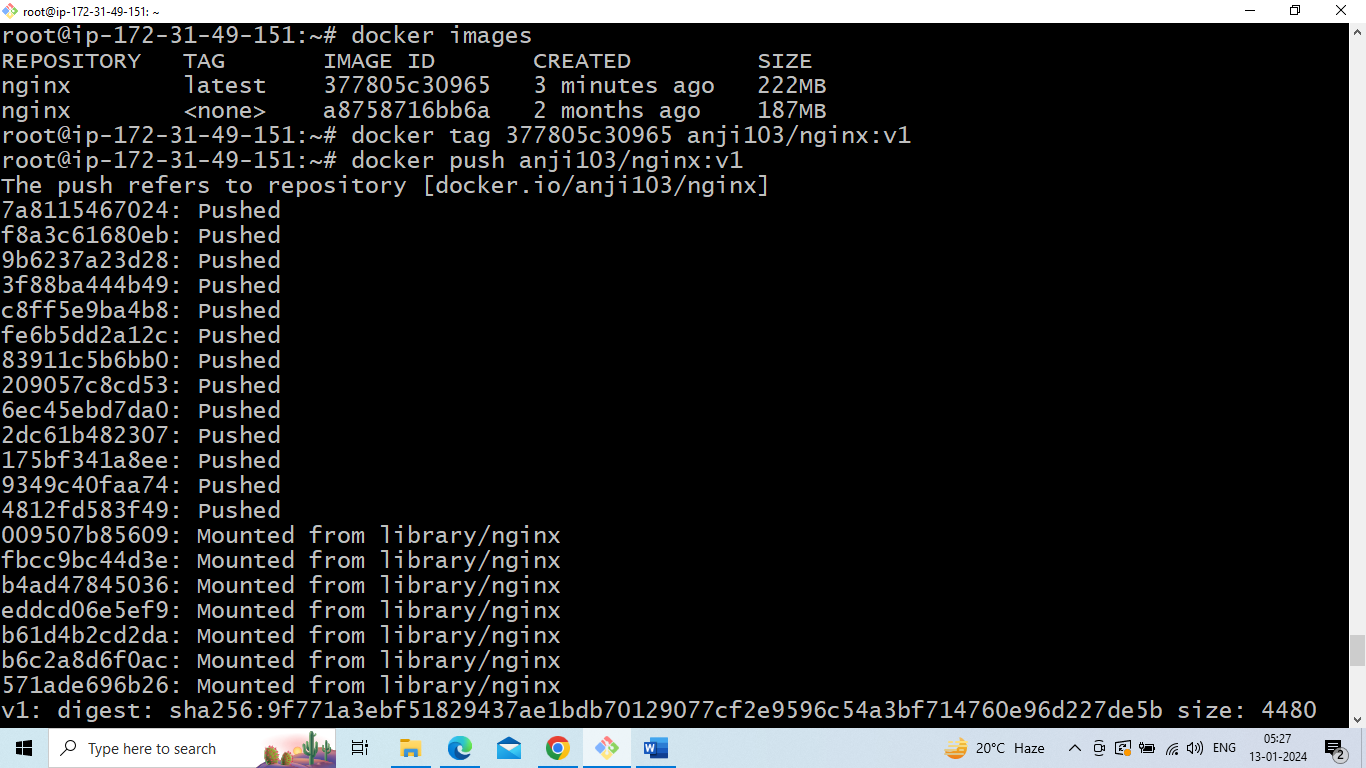
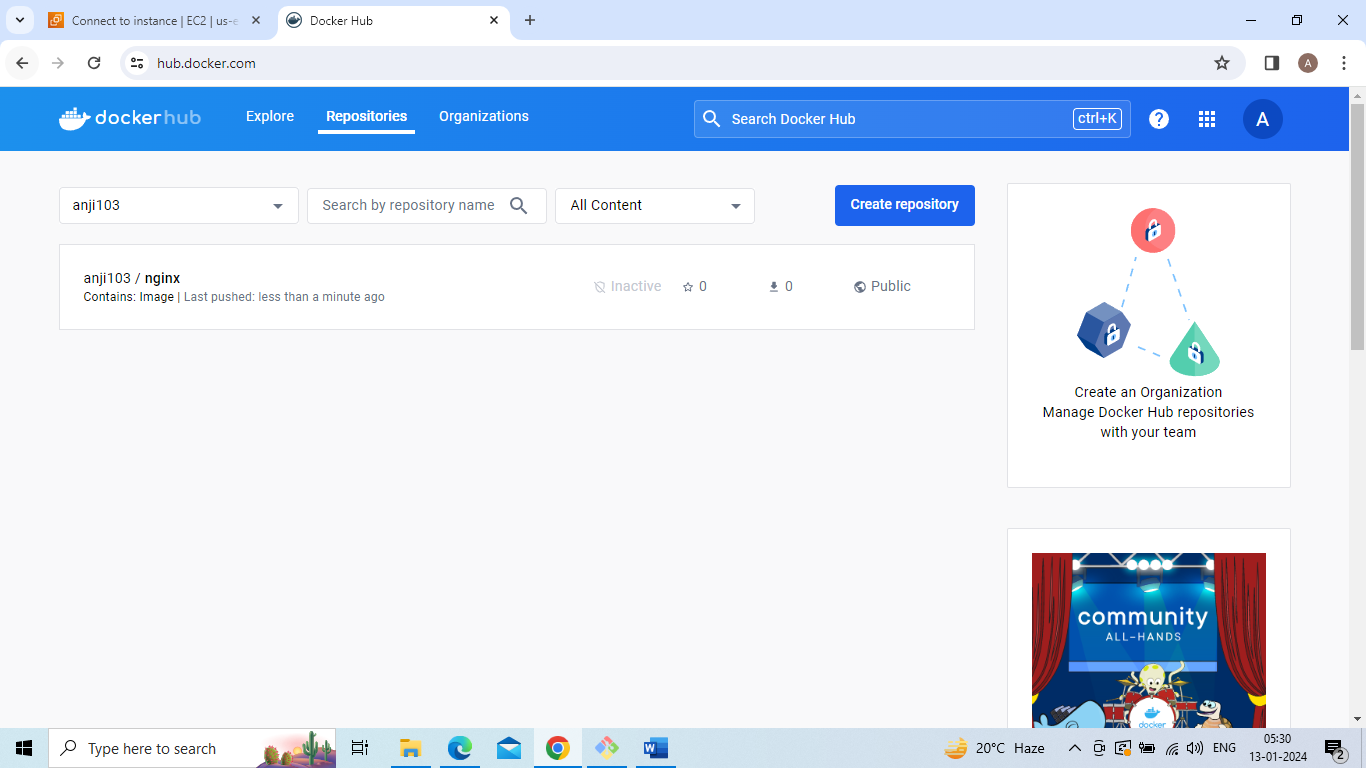


Image push to dockerhub is completed go to dockerhub and check whether the image is push or not



Nexus artifact

sudo -i(convert to root user)

apt update -y

apt install maven -y

apt install openjdk-17-jdk -y

go to google search --> tomcat 9 dowmload --->tar.zp(copy this link)

Cd /opt

Wget [https://download.sonatype.com/nexus/3/nexus-3.0.2-unix.tar.gz](https://download.sonatype.com/nexus/3/nexus-3.64.0-03-unix.tar.gz)

Ls

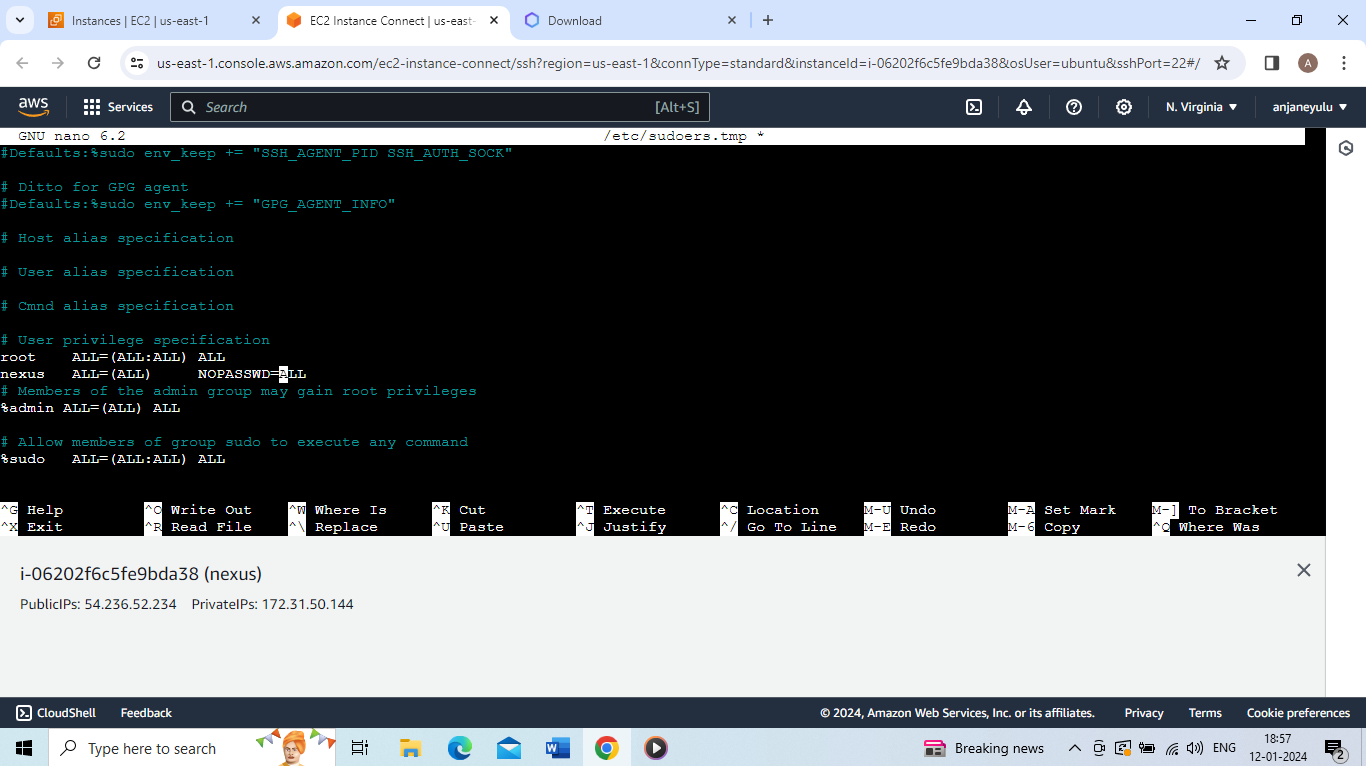
tar -xvzf nexus-3.64.0-03-unix.tar.gz

mv nexus-3.64.0-03 nexus

Adduser nexus

Visudo

Nexus ALL=(ALL) NOPASSWD:ALL

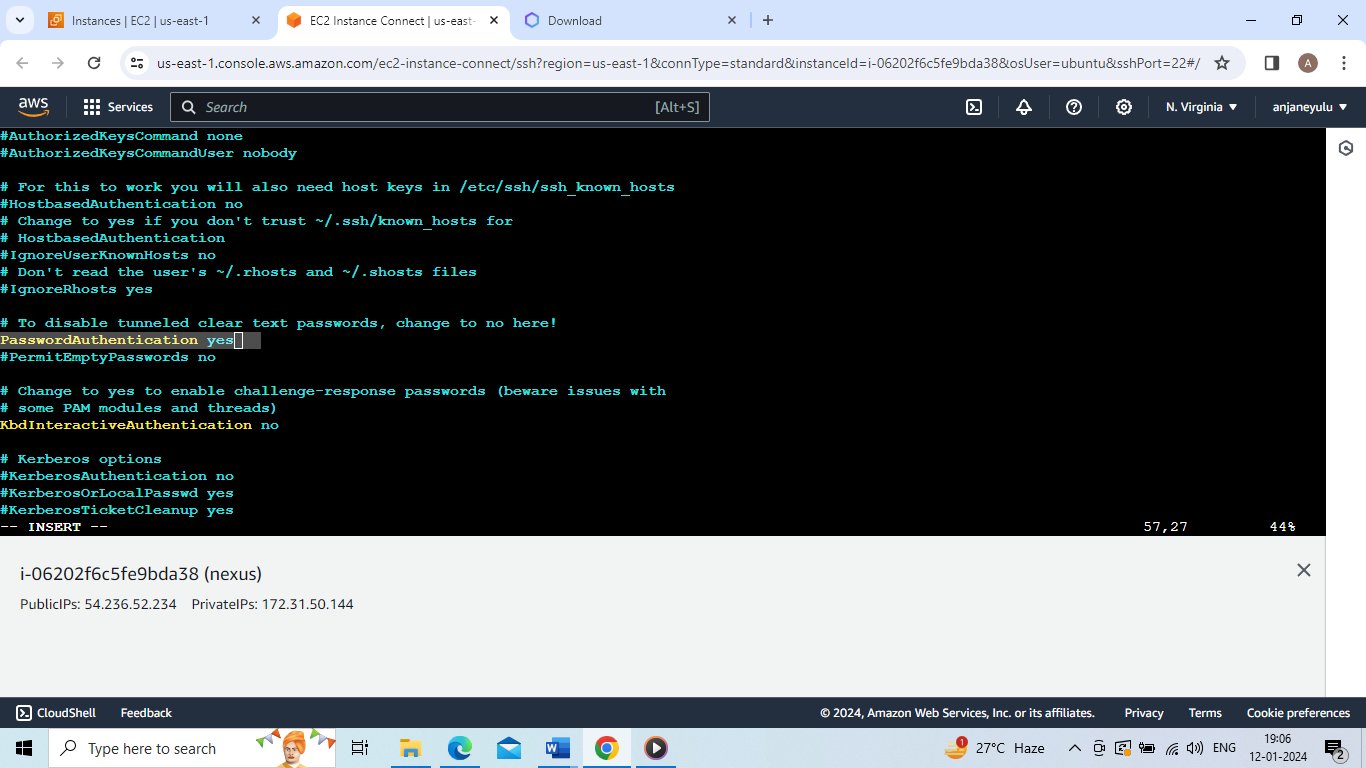


Cd /etc/ssh

Ls

Vi ssh\_config

PasswordAuthentication yes



Systemctl restart ssh

Cd

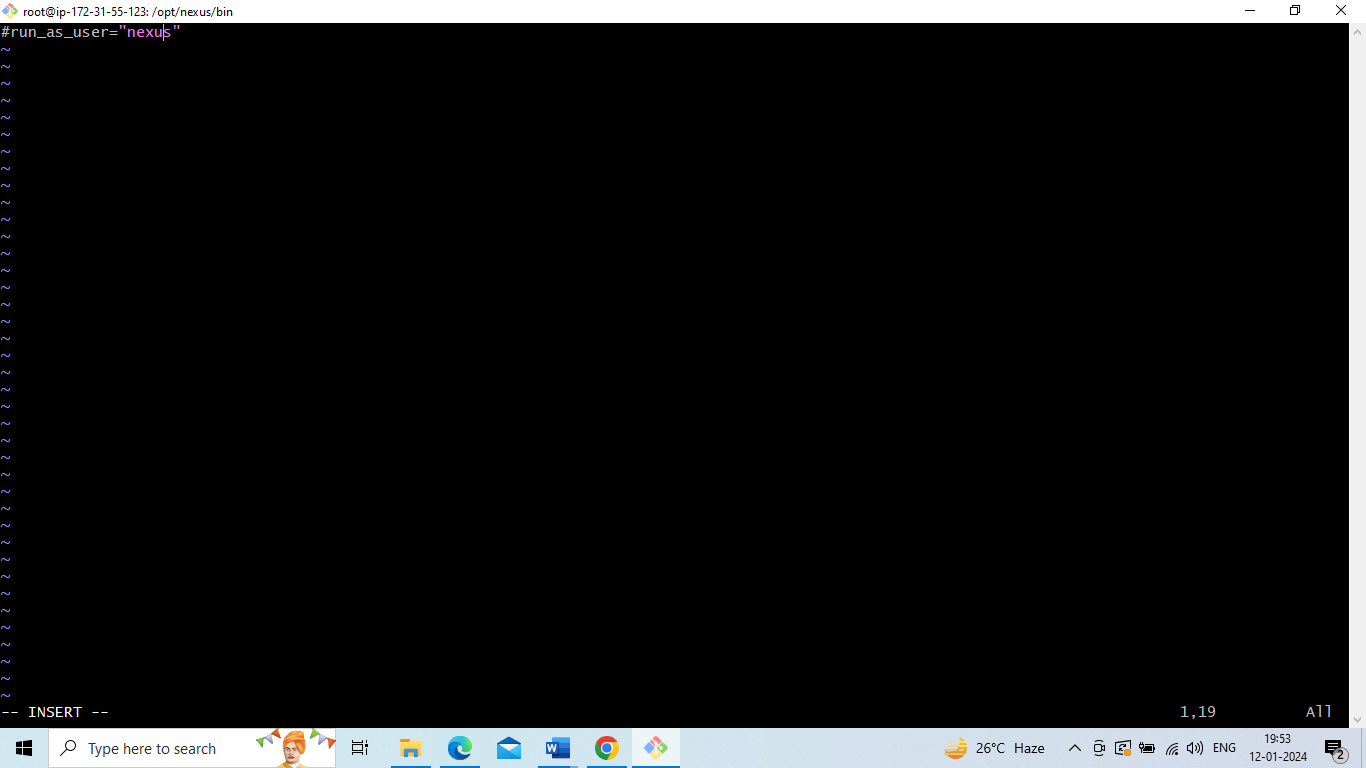
Adduser nexus

Chown –R nexus:nexus /opt/nexus

cd /opt/nexus/bin

Vi nexus.rc

#run\_as\_user="nexus"



su nexus

Ls

cd /opt/nexus/bin

./nexus start

./nexus status

ln /opt/nexus/bin/nexus /etc/inti.d/nexus