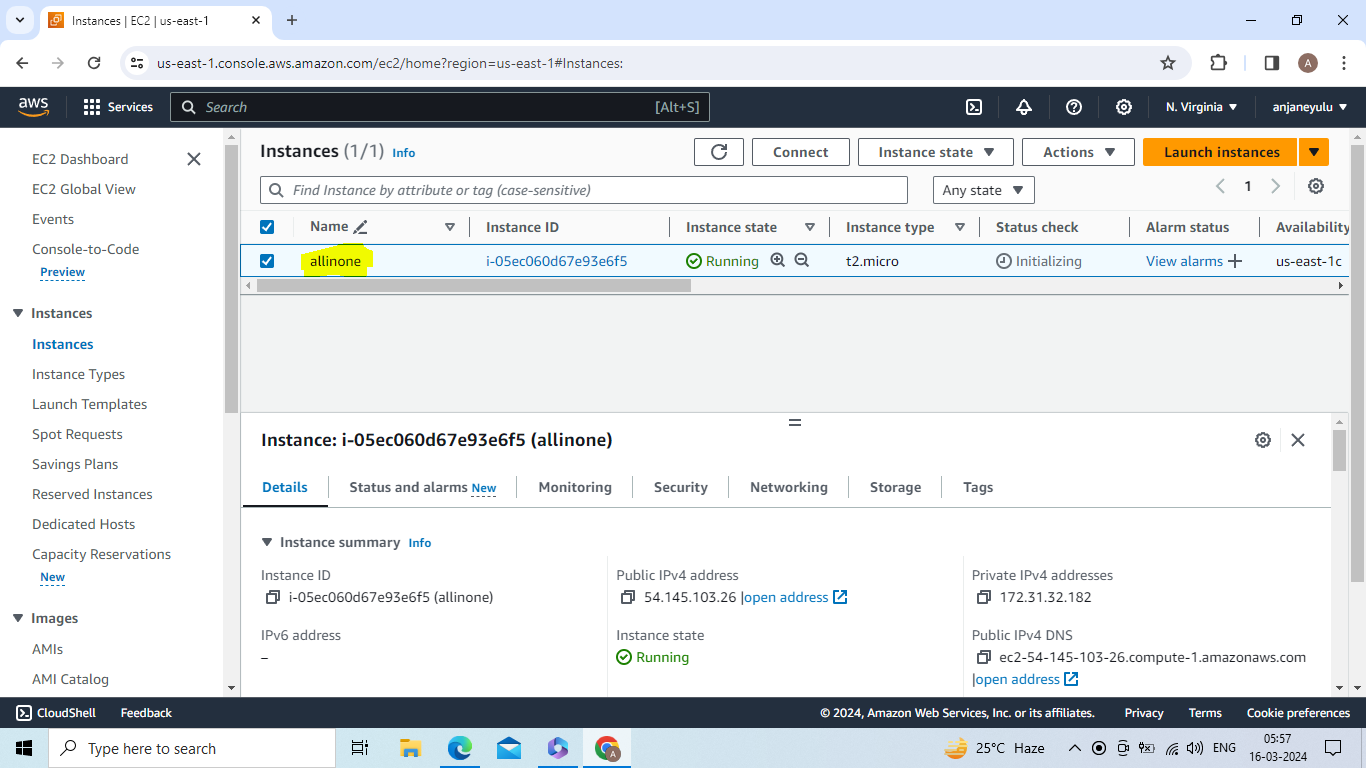
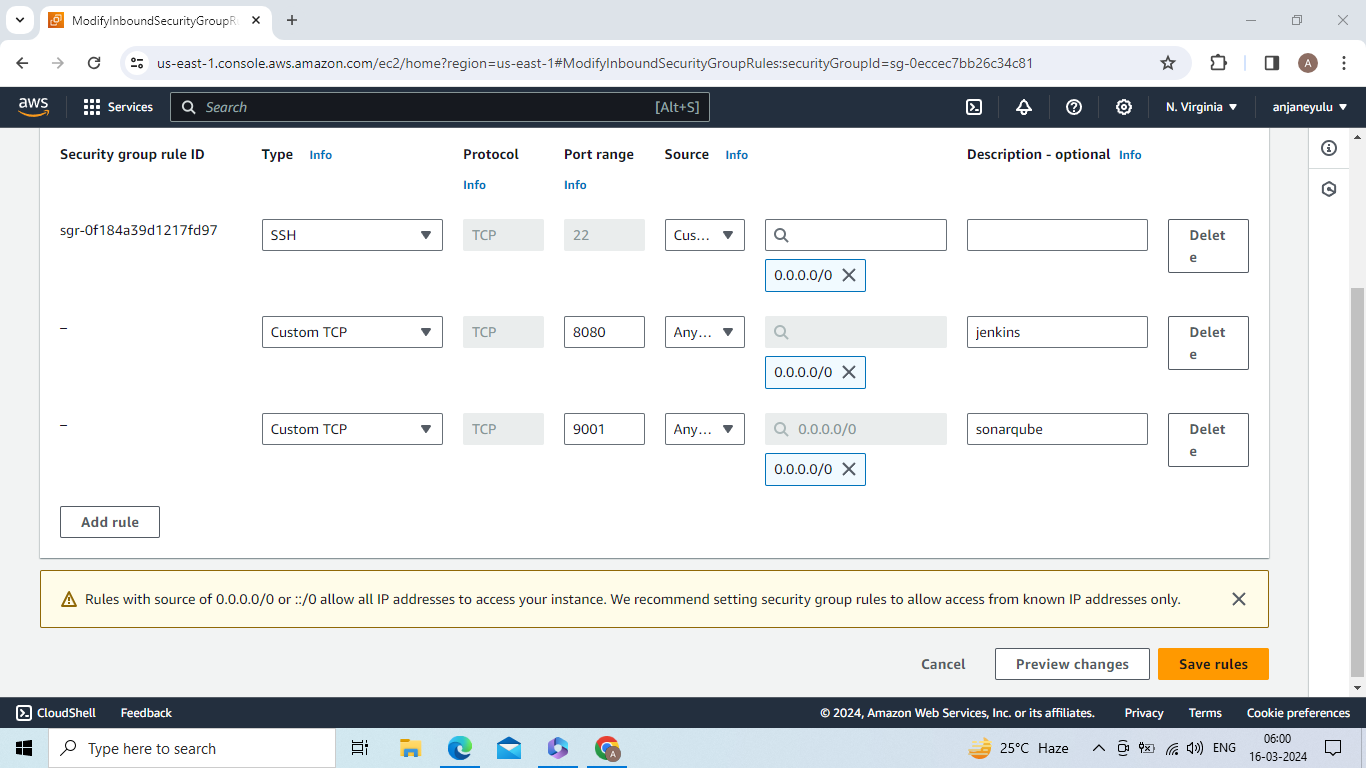
DEVSECOPS PROJECT

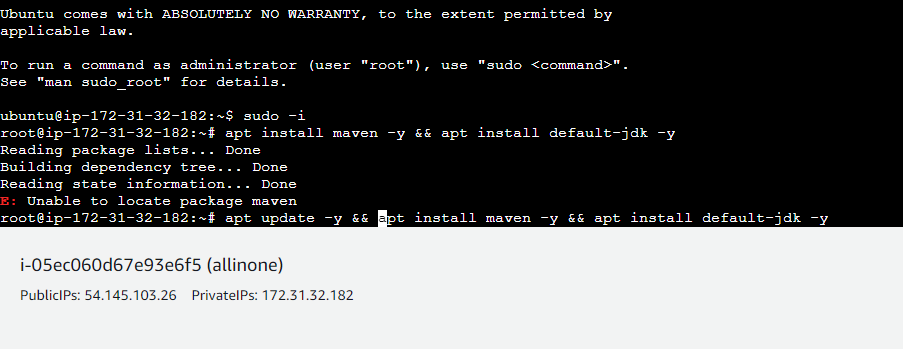
Create and launch one instance



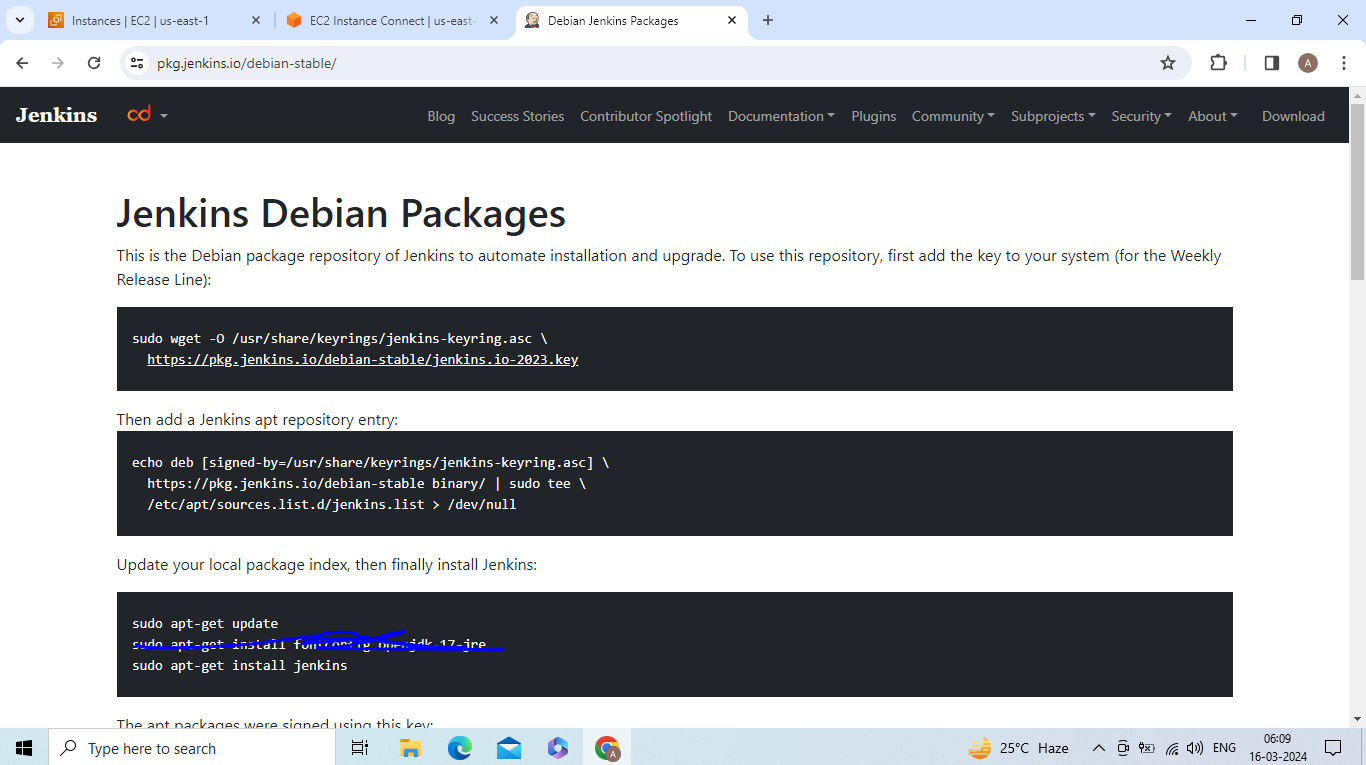
Go to security -->security group -->actions--> edit inbound rules -->8080(jenkins) , 9001 (sonarqube) -->save rules



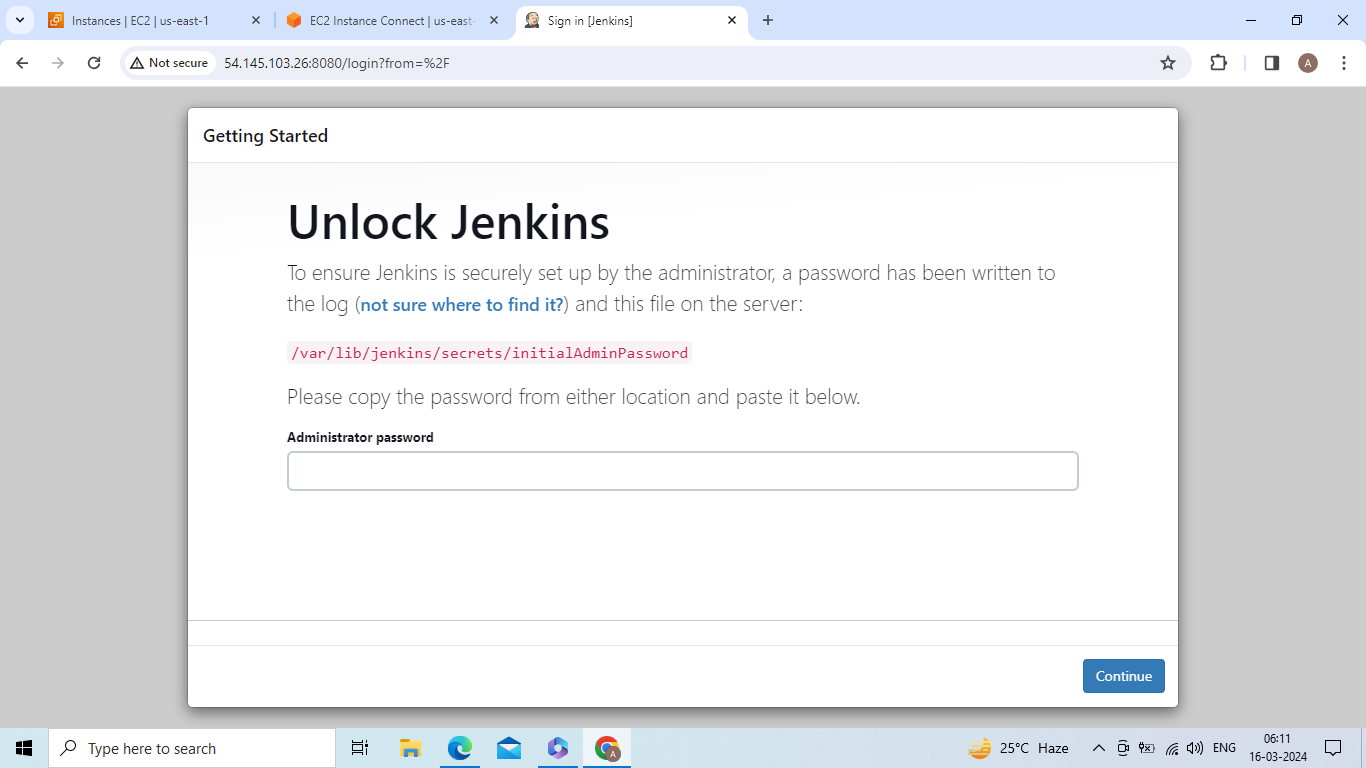
apt update -y && apt install maven -y && apt install default-jdk -y



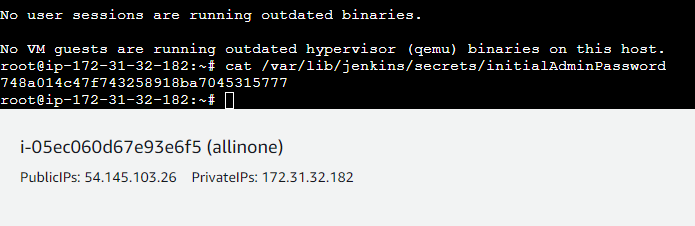
Install jenkins by using below commands



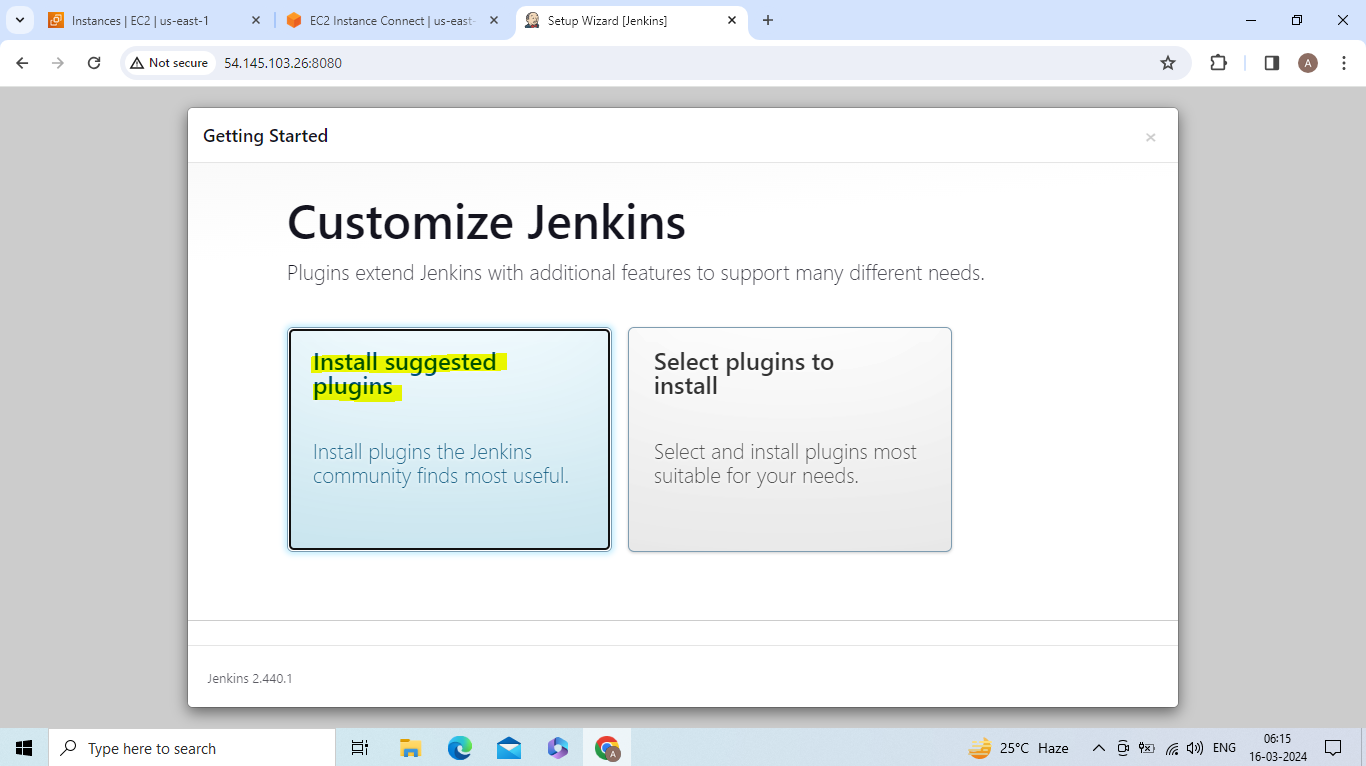
Copy public IP and port number 8080 and paste it in new page or browser then u get unlock jenkins page



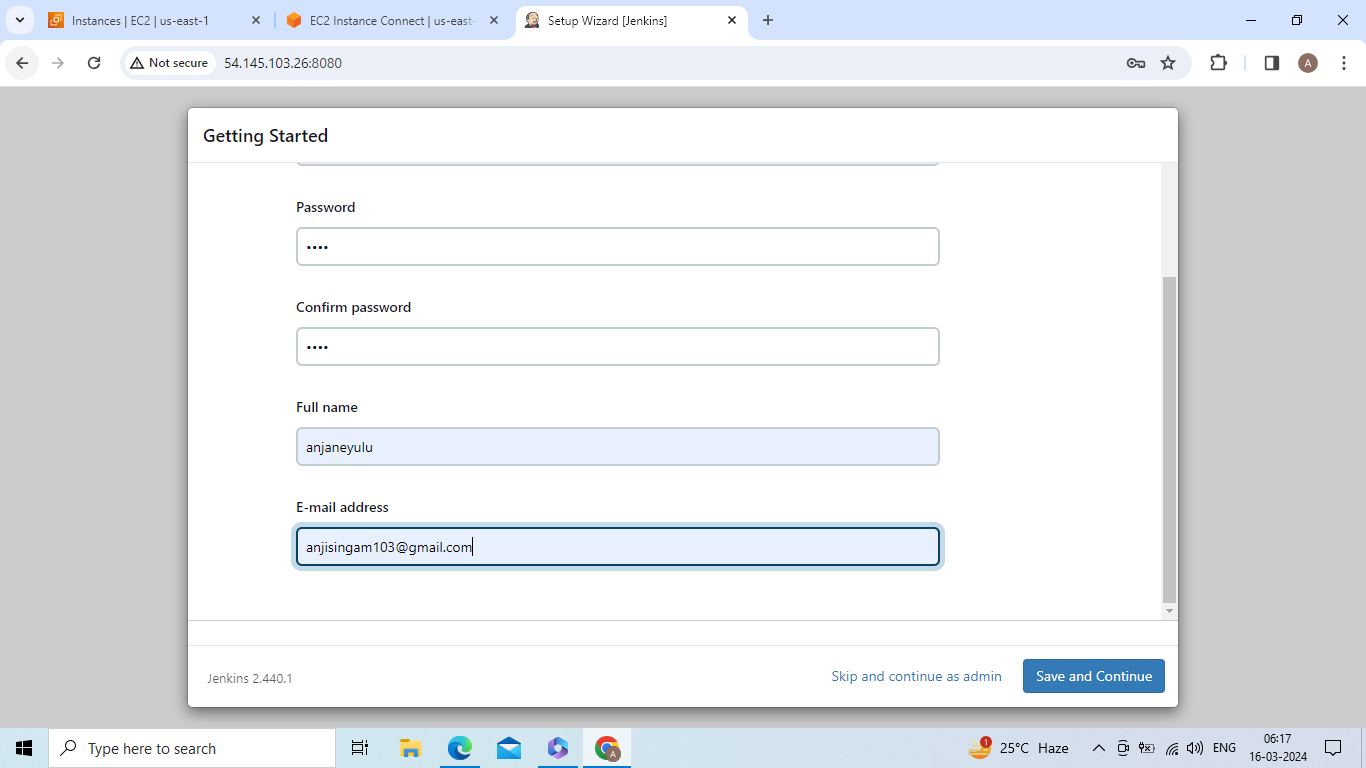
For **Administrator password copy above path paste it in server by using cat command**



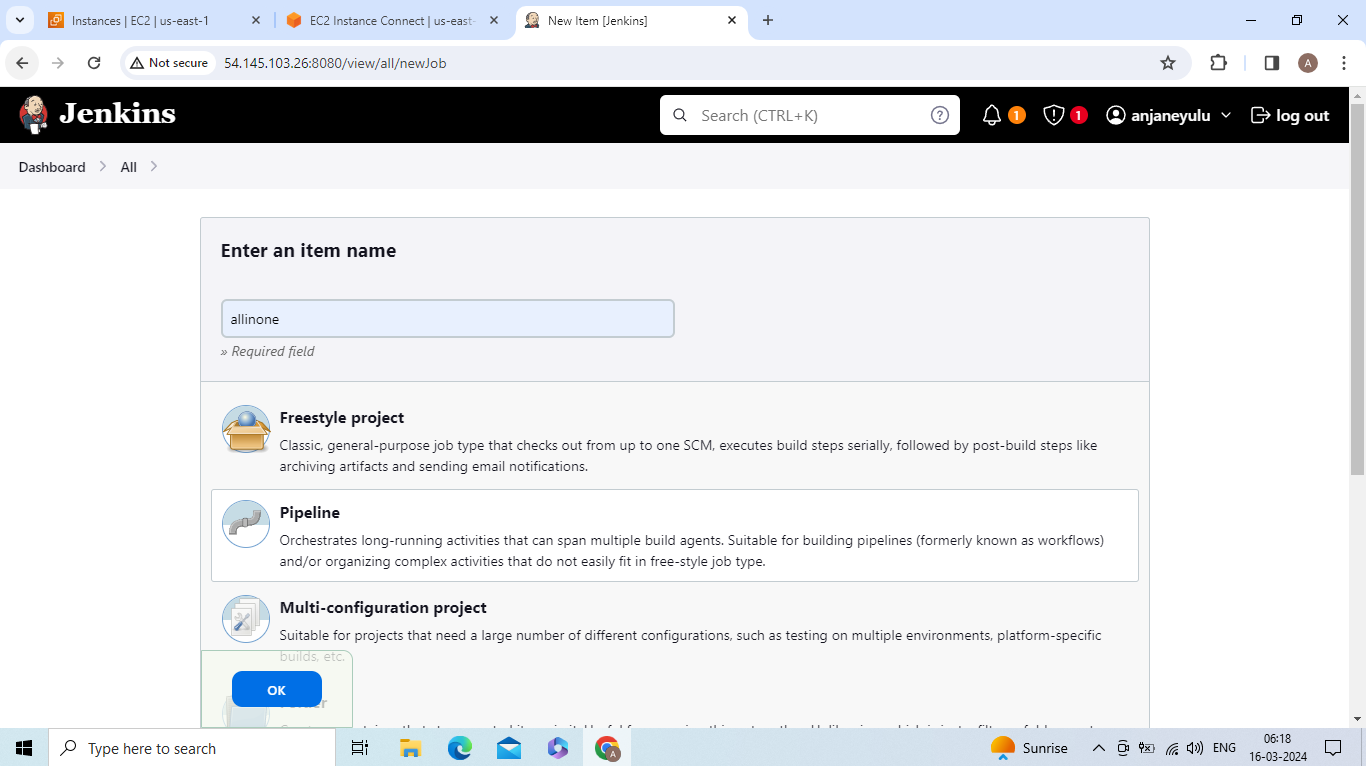
Copy password and paste it in unlock jenkins page



Login to jenkins by giving username, password,email,full name

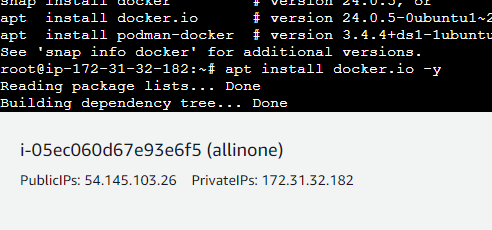


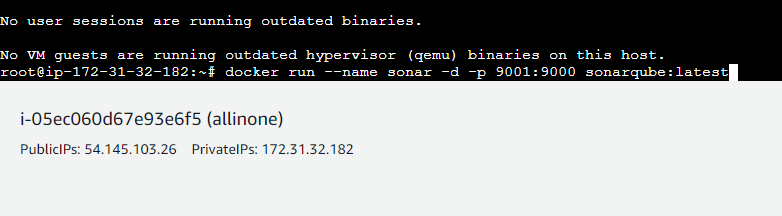
Create one new job with pipeline



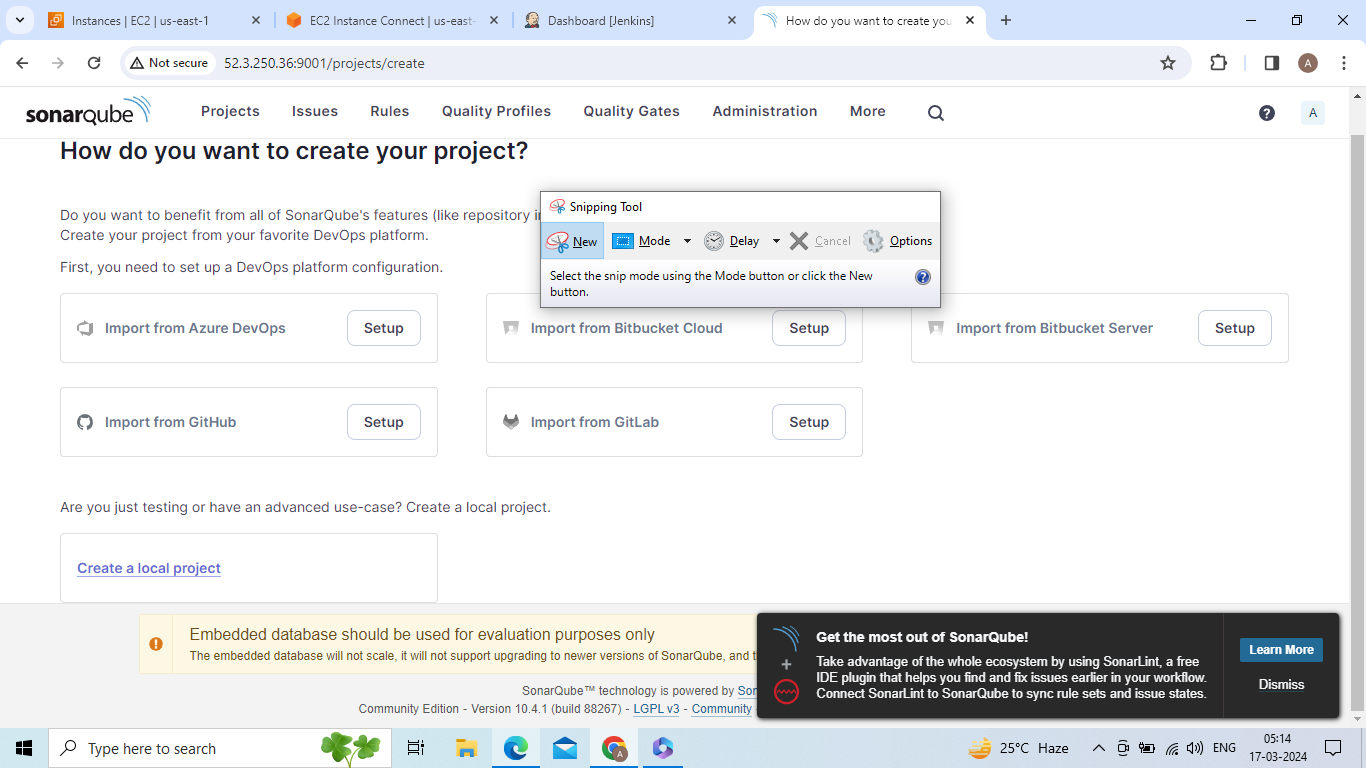
Go to manage jenkins --> plugins -->available plugins --> Maven Integration , Config File Provider , SonarQube Scanner , Sonar Quality Gates , Quality Gates , OWASP Dependency-Check , Eclipse Temurin installer --> install plugins

Go to server --> install docker ,sonarqube using docker

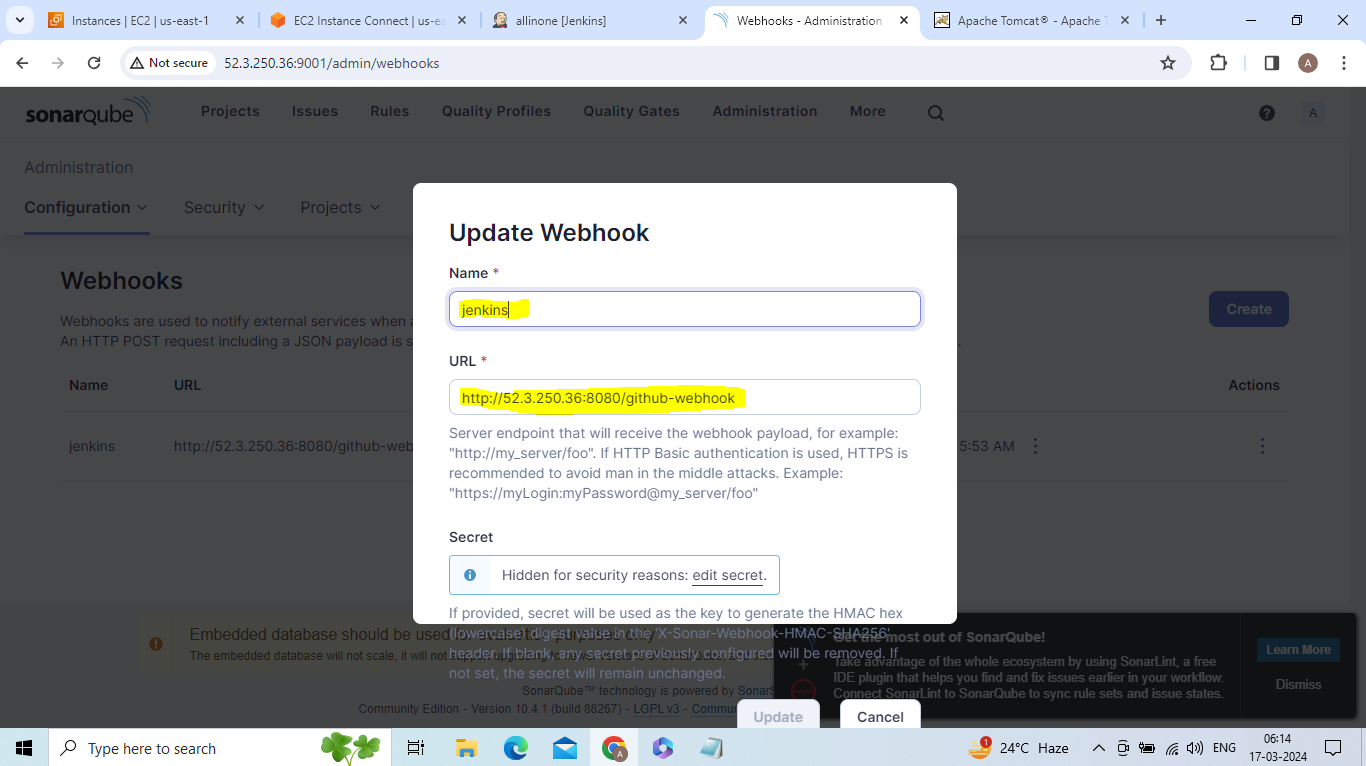




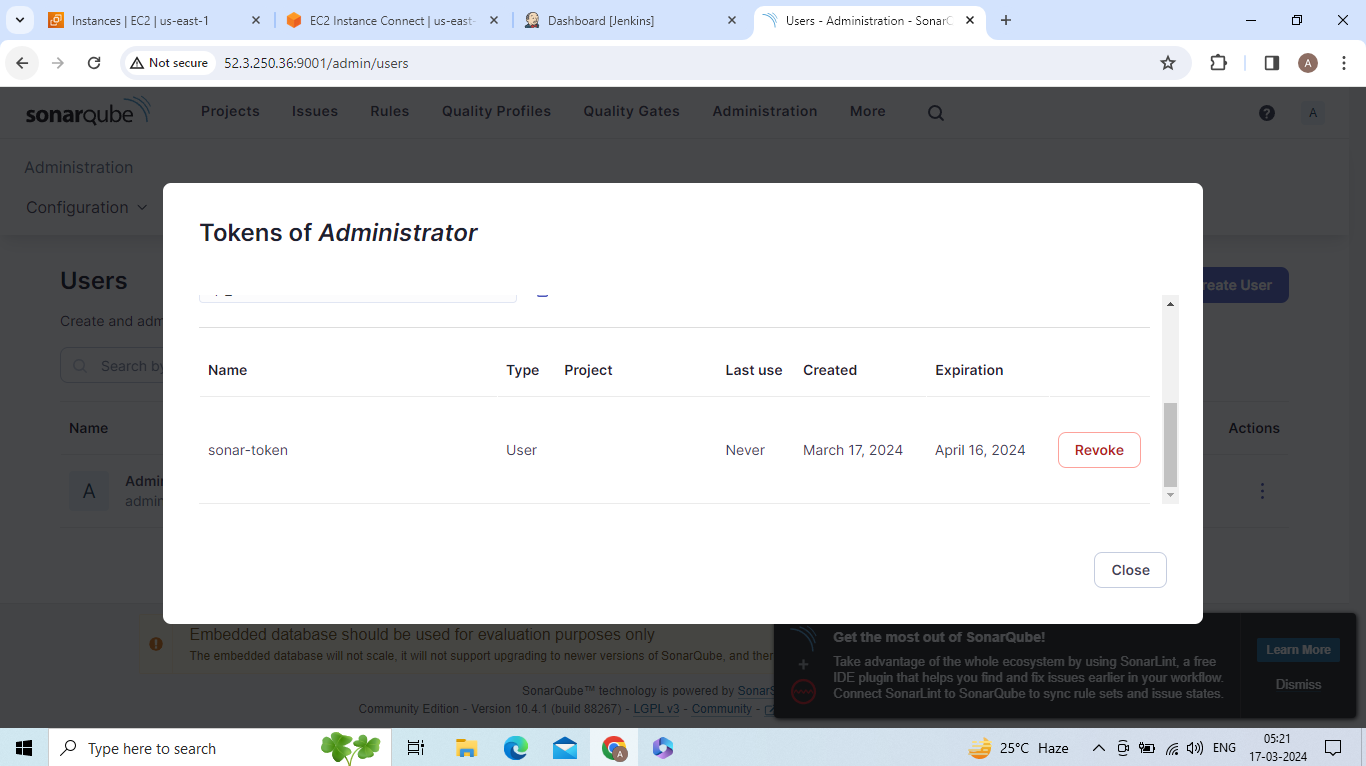
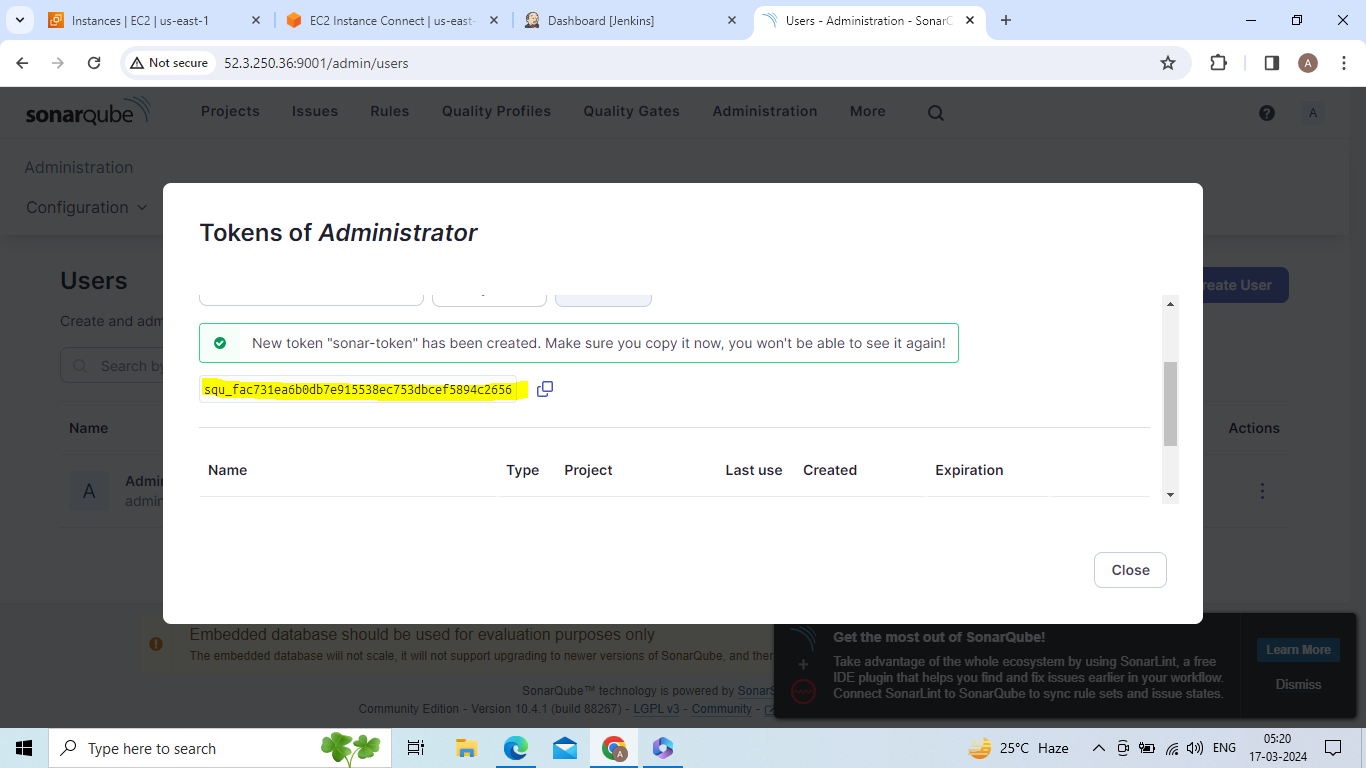
Copy public IP ,port number 9001 and paste it in new page or browser



Go adimminstration -->configuration-->webhook -->name – jenkins ,URL – jenkins URL

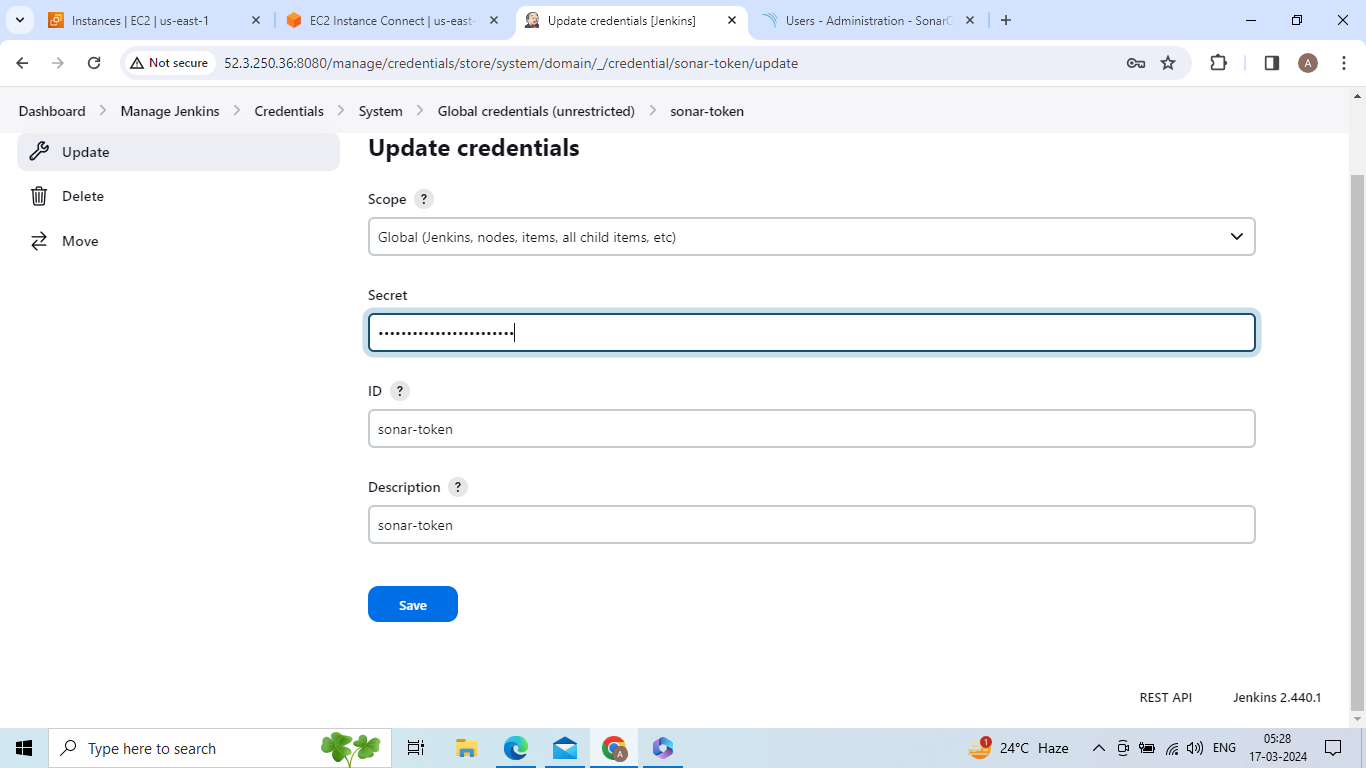


Go to adminstration -->security -->users -->token -->username – sonar-token ,generate token and copy token ID

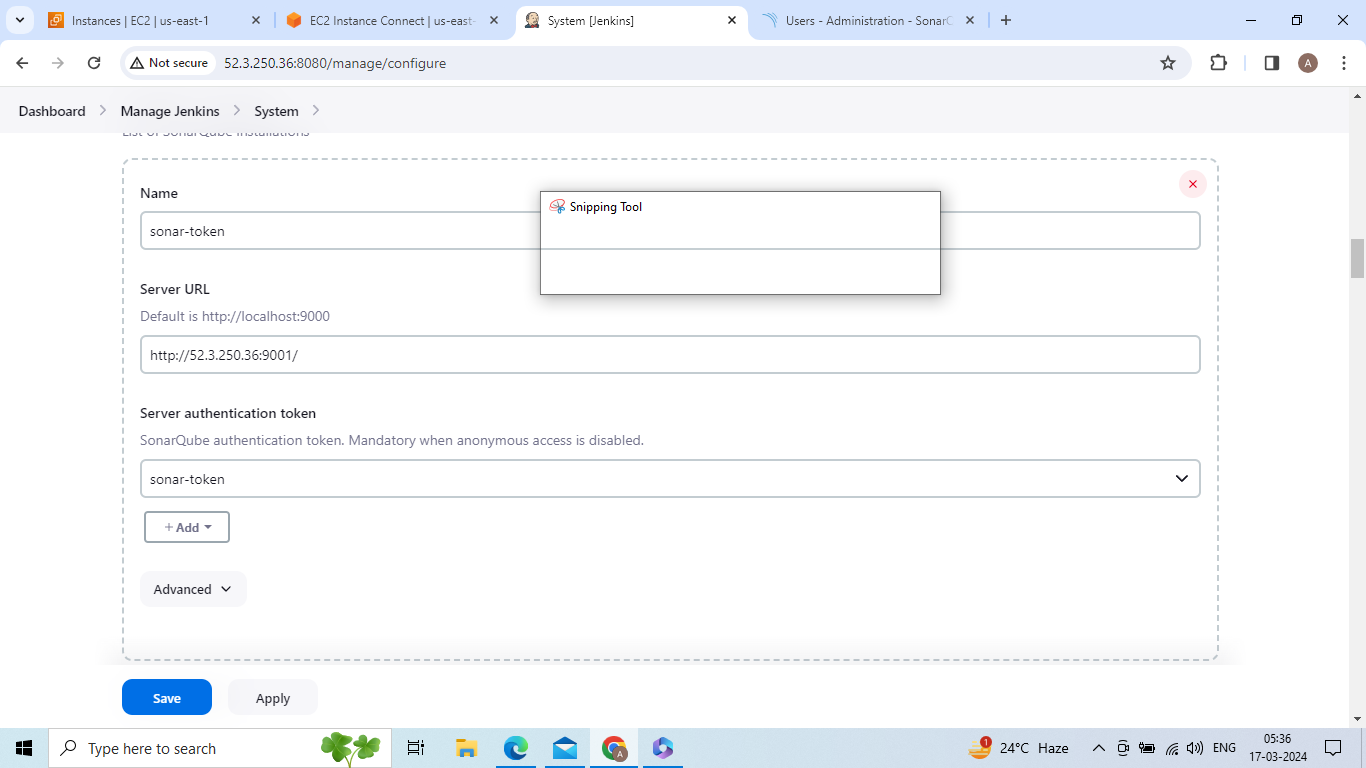


Go to manage jenkins --> credentials --> system --> global credentials -->add credentials

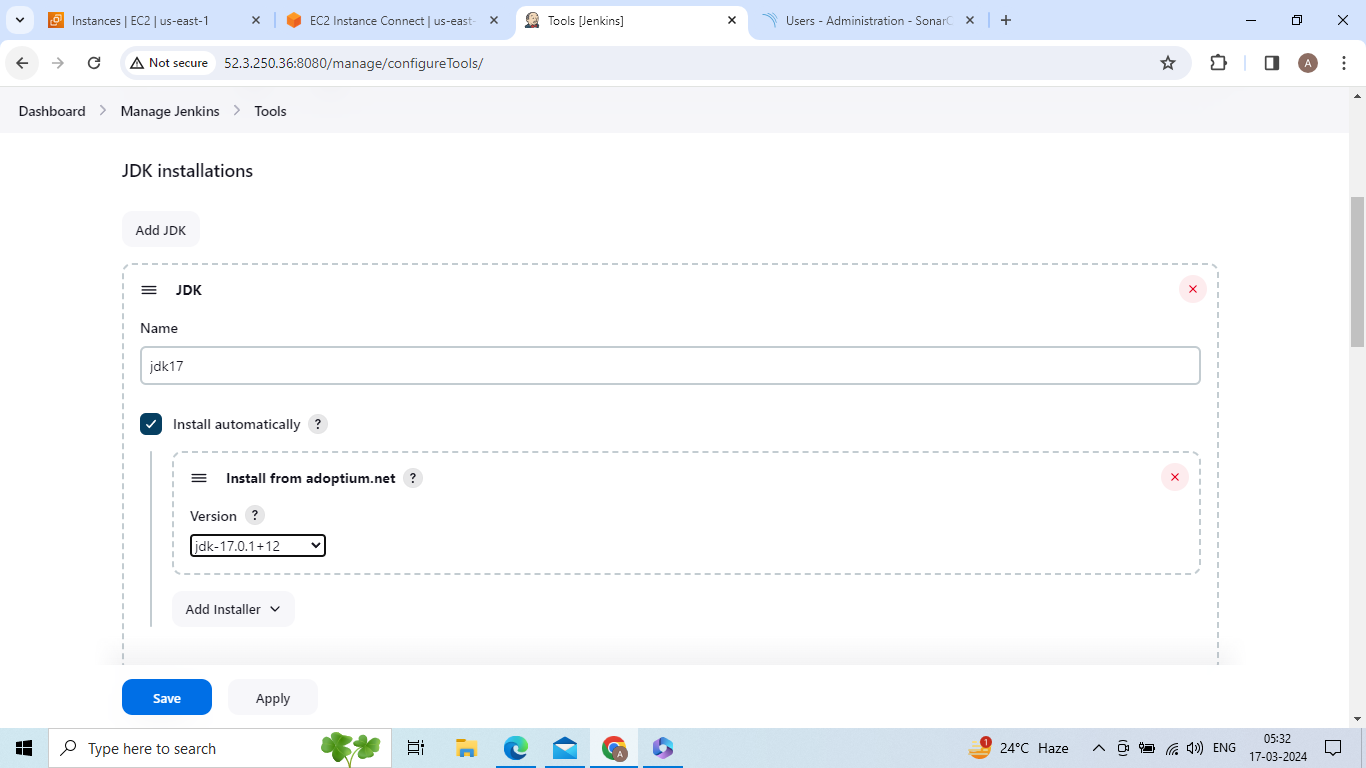
-->select secret text -->paste token ID(what you copied token above in sonarqube)



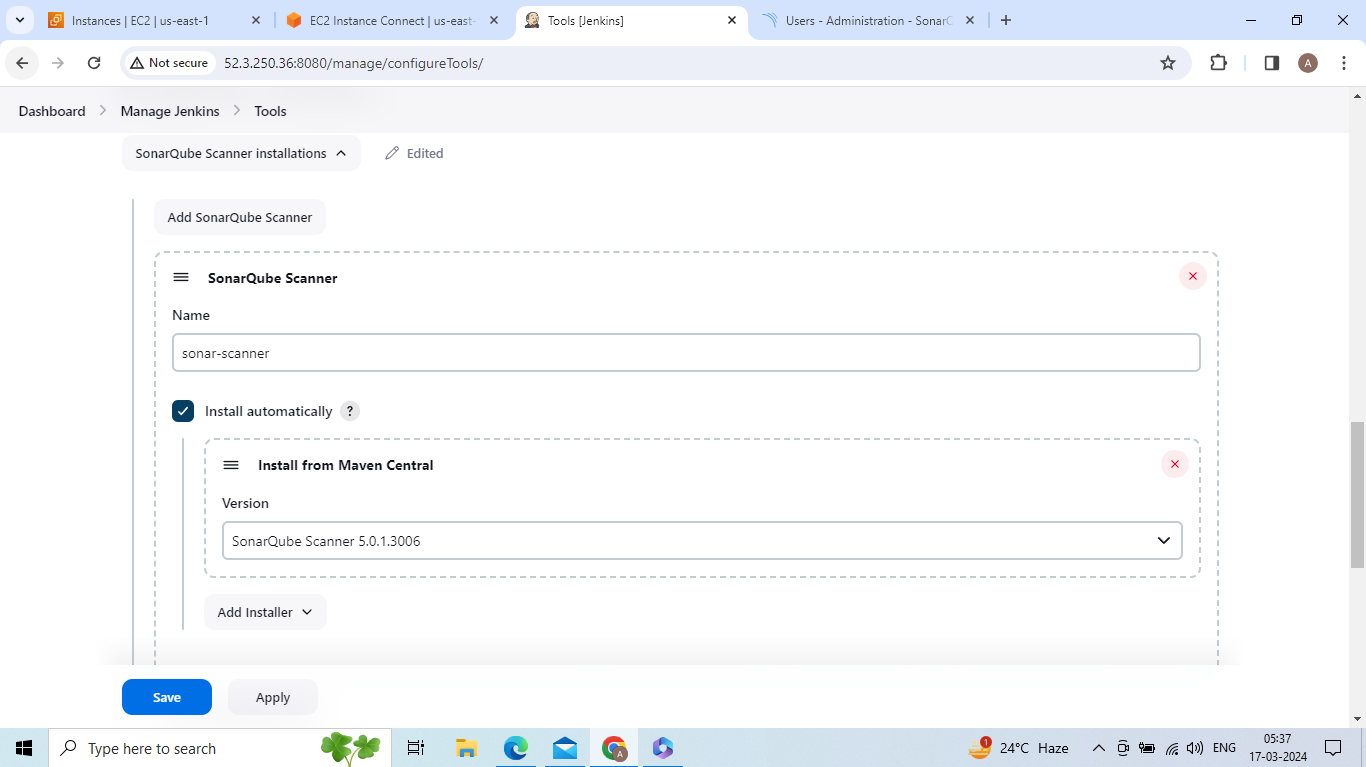
Go to manage jenkins --> system --> sonrqube installations



Go to manage jenkins --> tools -->jdk installions

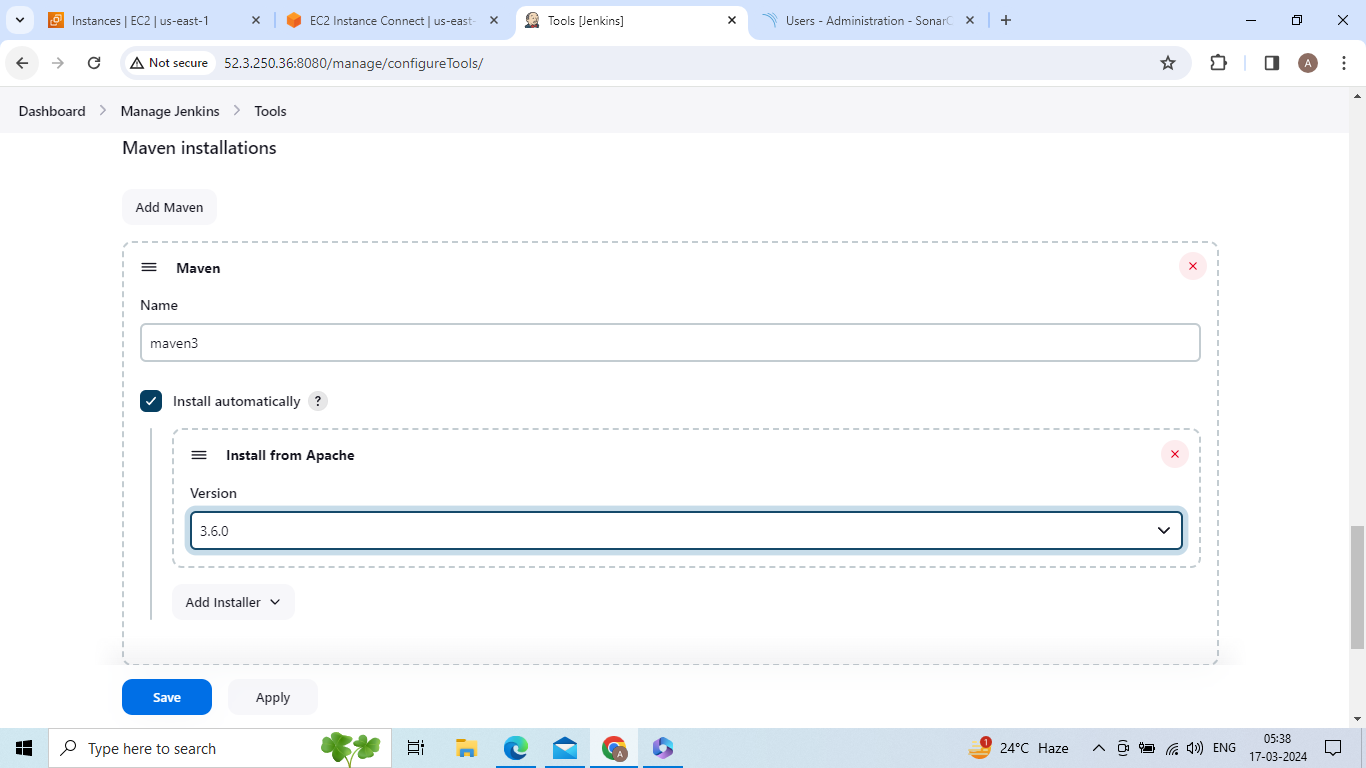


Go to manage jenkins --> tools --> sonar scanner installtions

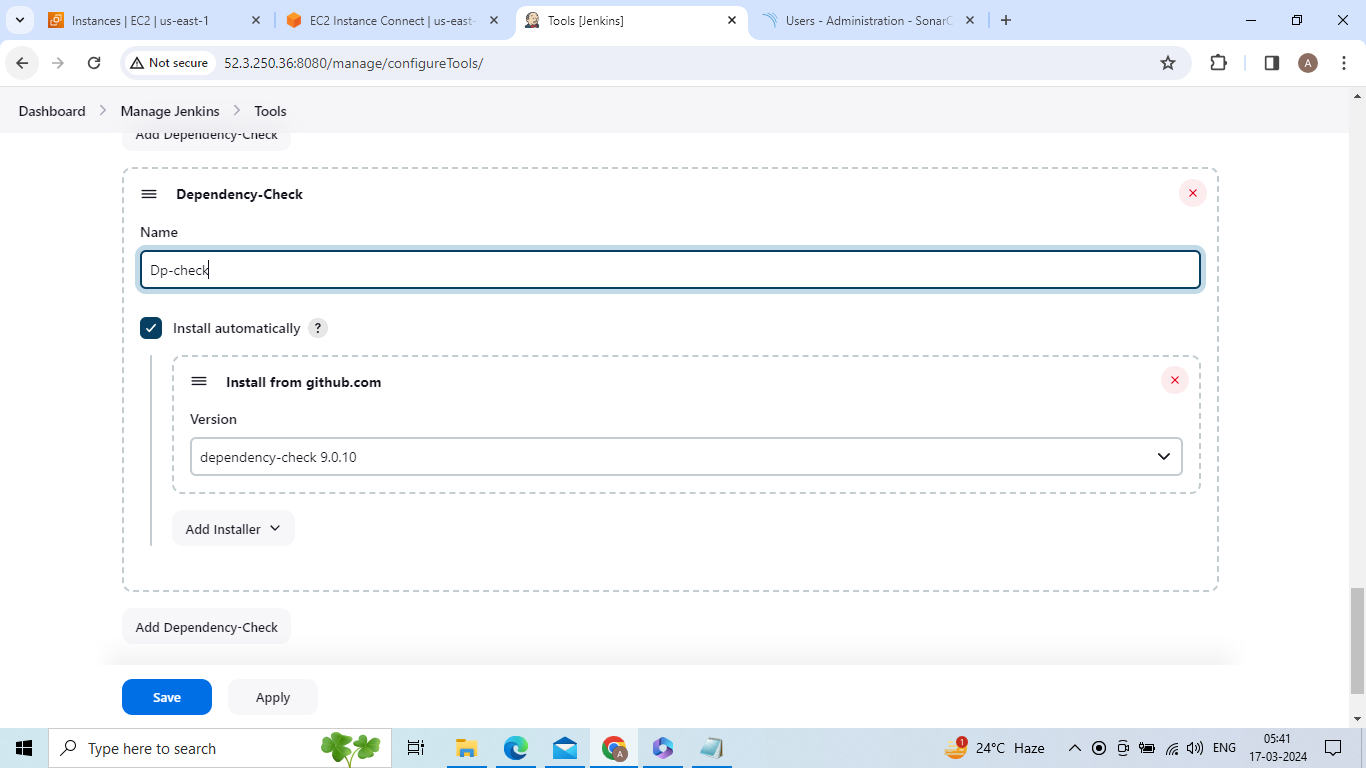


Go to manage jenkins --> tools --> sonar scanner installtions

Go to manage jenkins --> tools --> maven installtions



Go to manage jenkins --> tools --> dependency check



Go to allinone job --> write pipeline script

pipeline{

agent any

tools {

jdk 'jdk17'

maven 'maven3'

}

environment {

SCANNER\_HOME=tool 'sonar-scanner'

}

stages{

stage ('clean Workspace'){

steps{

cleanWs()

}

}

stage ('checkout scm') {

steps {

checkout scmGit(branches: [[name: '\*/master']], extensions: [], userRemoteConfigs: [[url: 'https://github.com/anji103/taxi-booking.git']])

}

}

stage ('maven compile') {

steps {

sh 'mvn clean compile'

}

}

stage ('maven Test') {

steps {

sh 'mvn test'

}

}

stage("Sonarqube Analysis "){

steps{

withSonarQubeEnv('sonar-token') {

sh ''' $SCANNER\_HOME/bin/sonar-scanner -Dsonar.projectName=Petshop \

-Dsonar.java.binaries=. \

-Dsonar.projectKey=Petshop '''

}

}

}

stage("OWASP Dependency Check"){

steps{

dependencyCheck additionalArguments: '--scan ./ --format XML ', odcInstallation: 'Dp-check'

dependencyCheckPublisher pattern: '\*\*/dependency-check-report.xml'

}

}

}

}

Save and apply

Build now

