TASK

Jenkins CI/CD Pipeline with GitHub + Maven + Nexus + SonarQube + Tomcat

Here we will complete our setup in 6 steps

1.1) In AWS Portal create a new instance as,

Name  : Jenkins-server

AMI    : Ubuntu

Instance type  : t2.medium

Key pair : create

Allow: HTTP & HTTPS

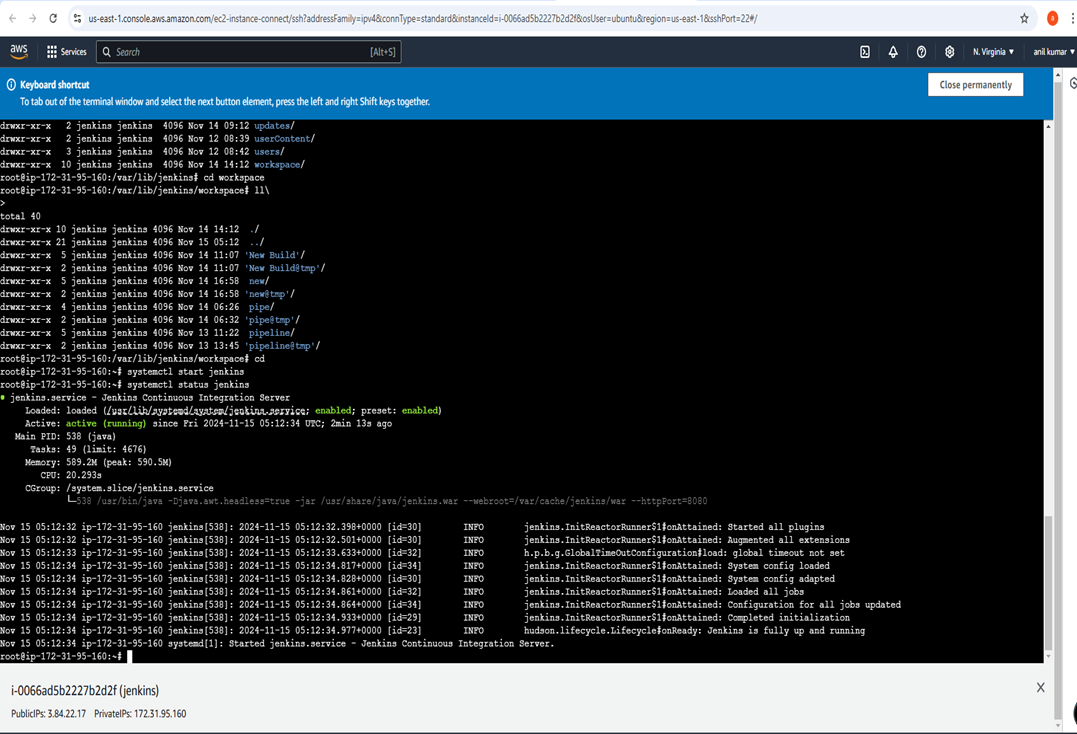
Allow: port number : 8080 (Jenkins default port number)

Allow : SSH 22

1.2) Connect to Ubuntu VM

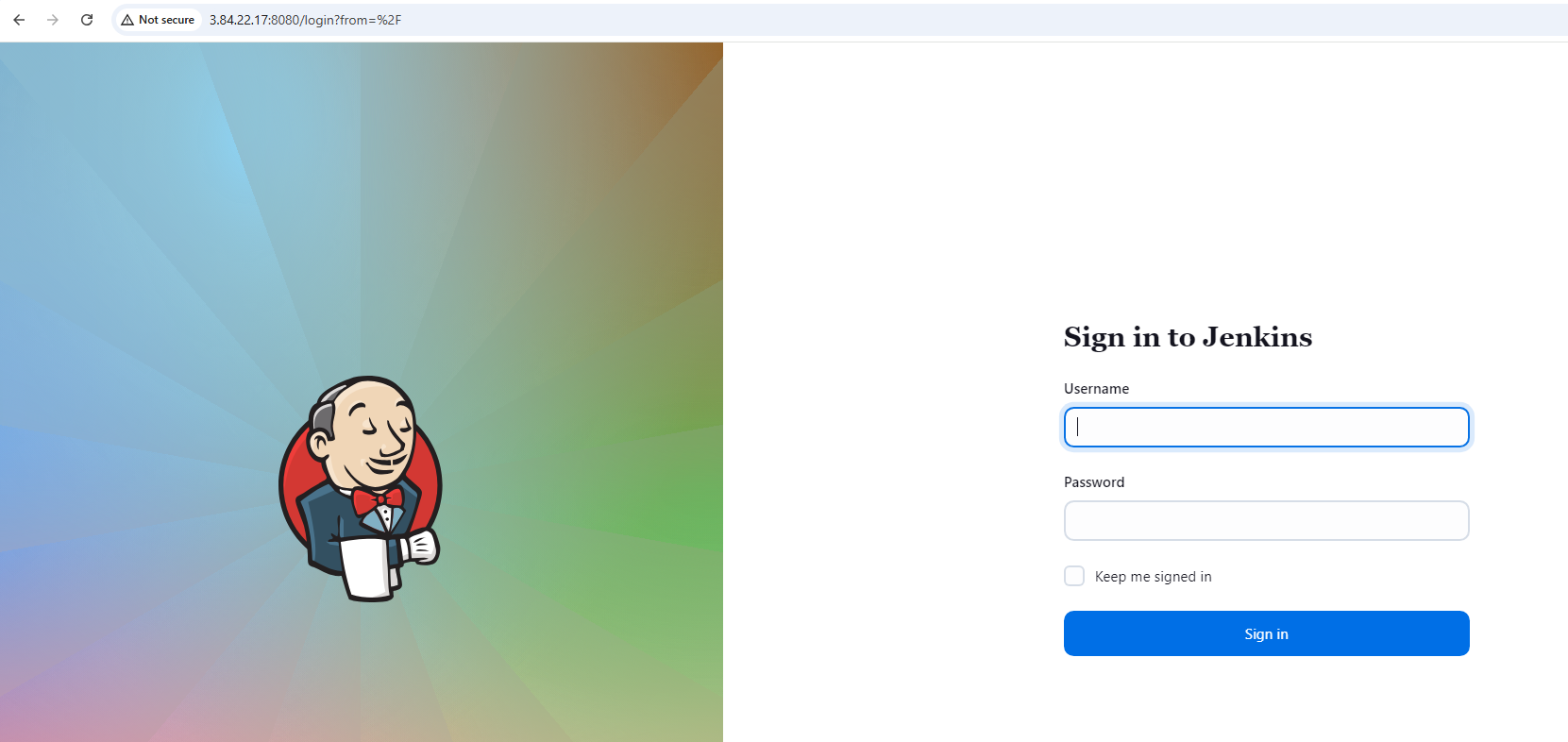
1.3) Now install Jenkins on the machine, by following this link

<https://www.jenkins.io/doc/book/installing/linux/#debianubuntu>

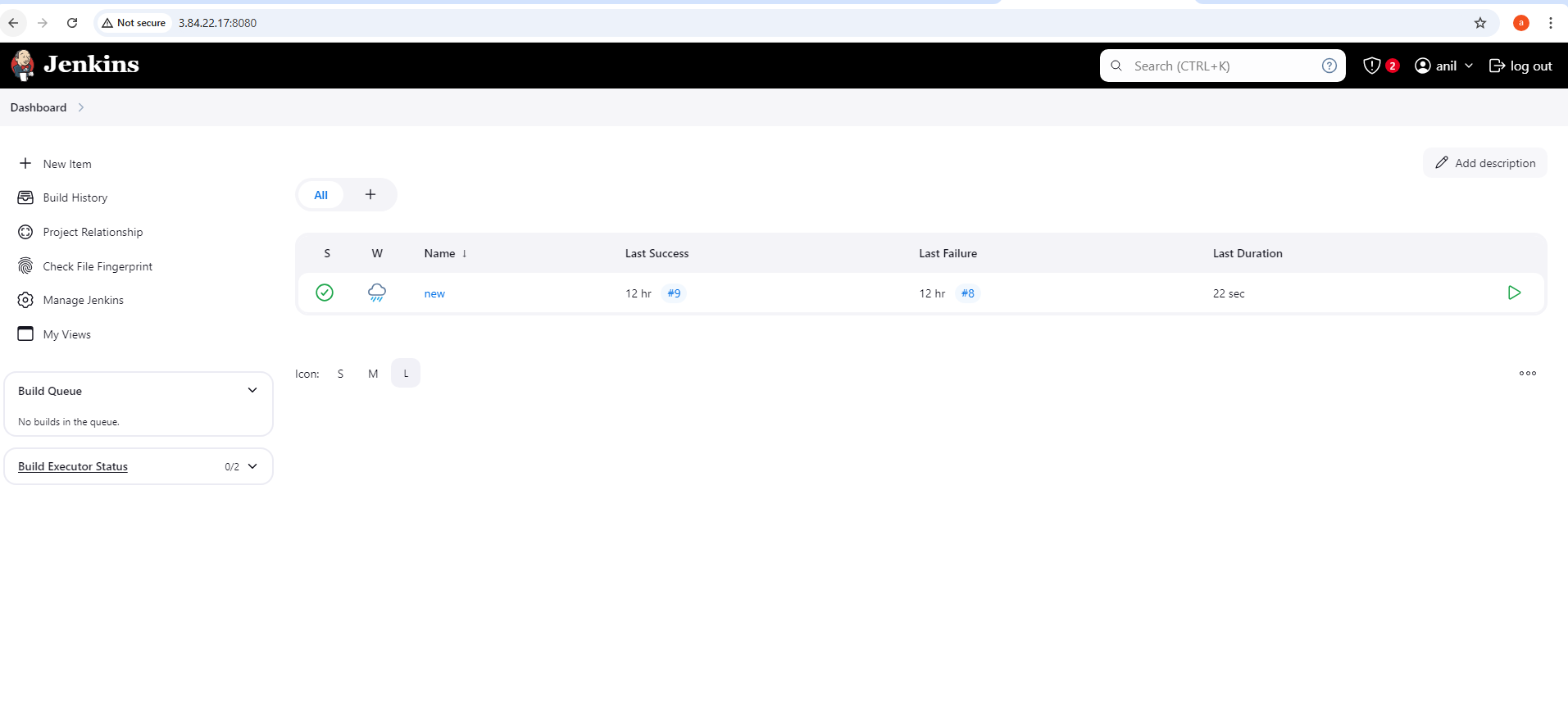


1.4) Now copy the public IP of the machine and paste it to the browser to access the Jenkins Portal.

VM Public IP : 8080



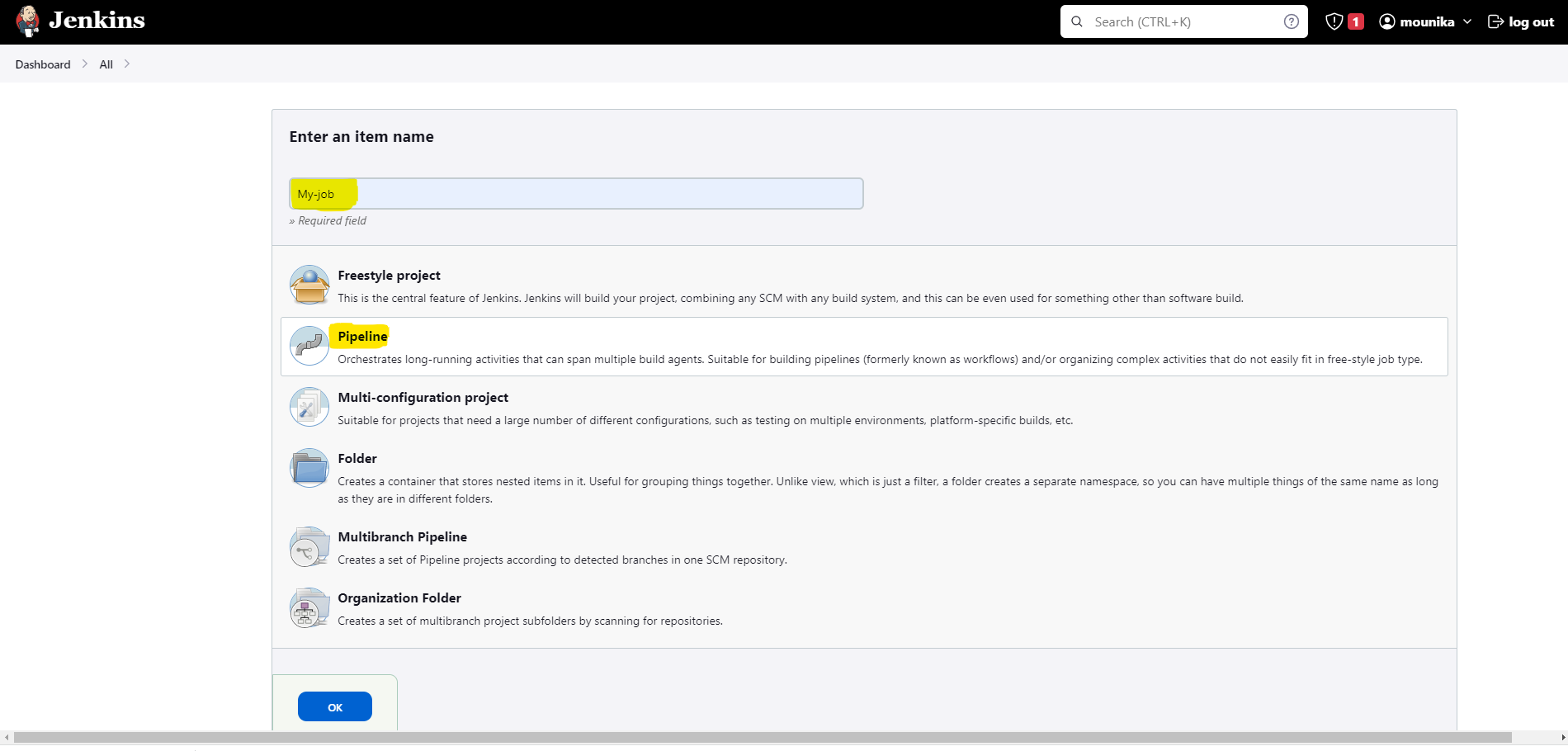
Create a CI/CD Pipeline,

From Jenkins Dashboard click on “New Item”

Create a pipeline for project

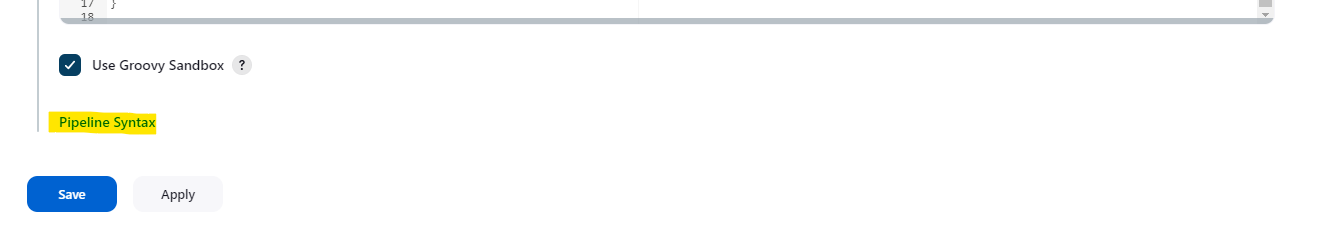
Enter an item name --> choose project name

Click on pipeline option --> ok



Step - 2. Source Code Clone from GitHub

2.1) Select pipeline syntax



2.2) Select GitHub Repository URL  and add credentials

2.3)  Add GitHub pipeline syntax same like below



Step - 3) Maven Build

3.1) Add Maven in Global Tool Configuration

3.2) Select Maven name & Maven version

3.3) Write pipeline syntax to build java project

stage('Maven Build'){

def mavenHome = tool name: "Maven-3.8.6", type:"maven"

def mavenCMD = "${mavenHome}/bin/mvn"

sh "${mavenCMD} clean package"

}

3.4) Maven build Success and target folder created

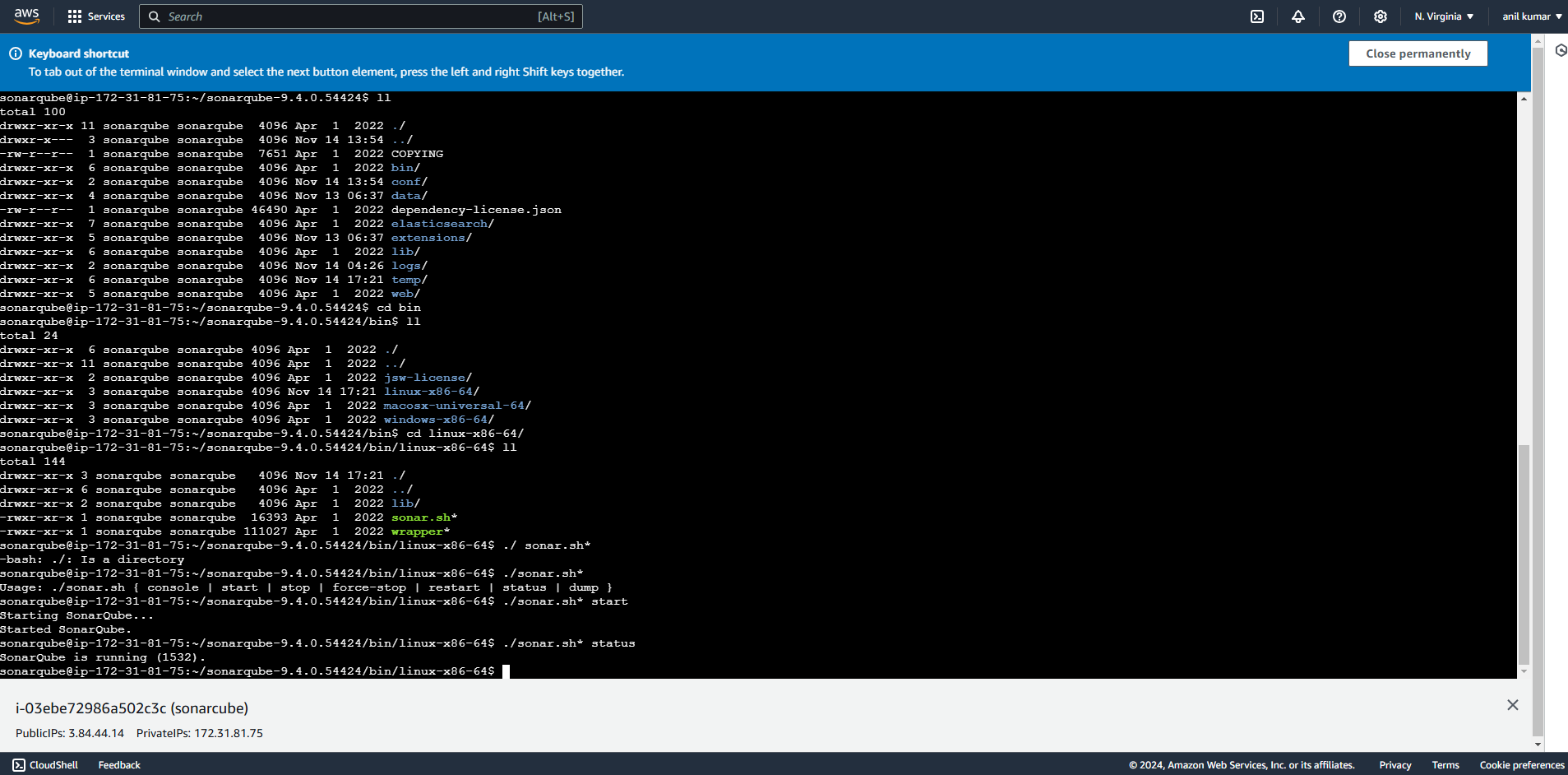
Step - 4) SonarQube Scanner

4.1) In AWS Portal, create a new instance as,

Name  : Sonar-Server

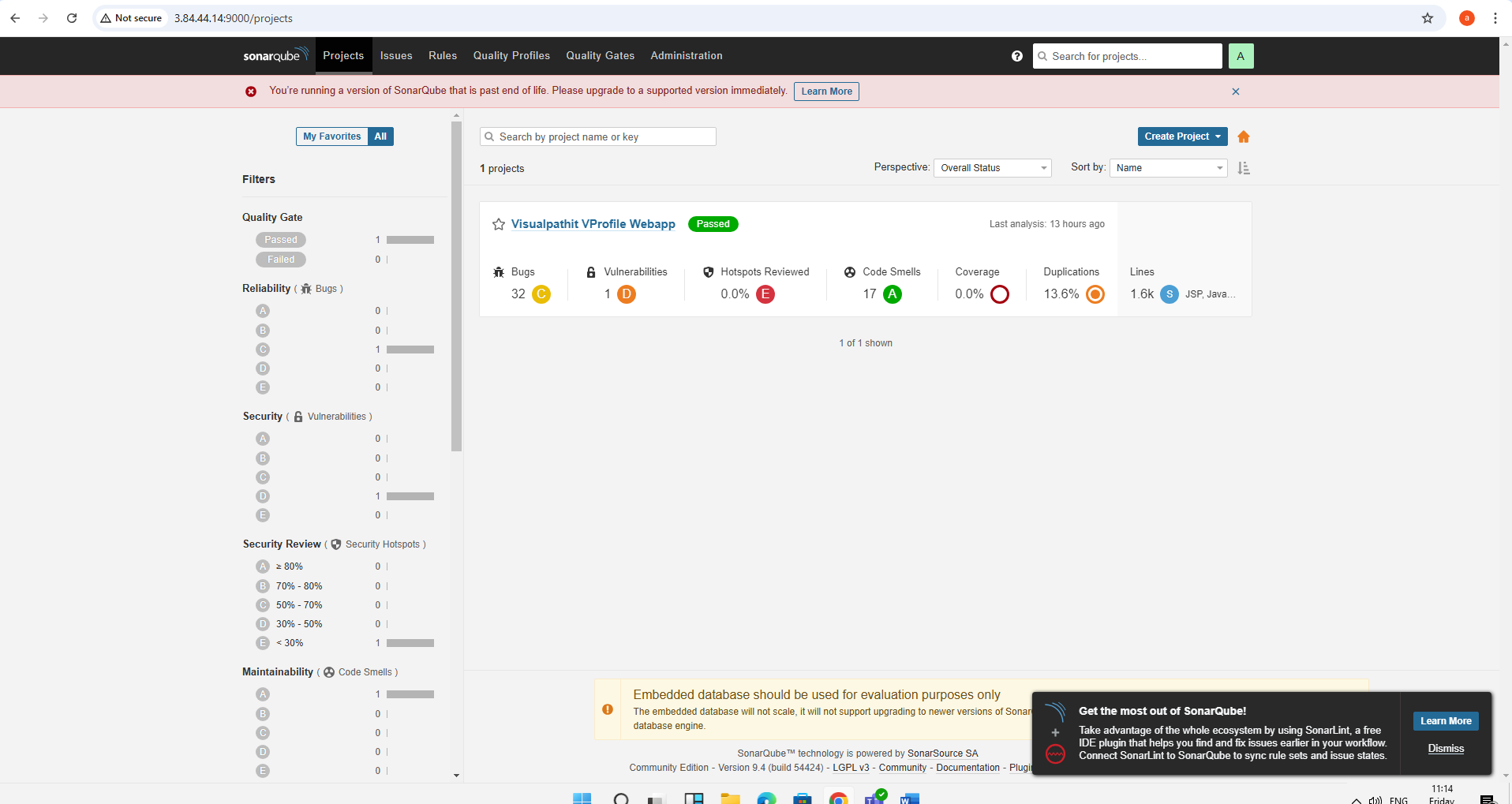
Allow: port number: 9000 (SonarQube default port number)

4.2) Install SonarQube software and check sonar running



4.3) Now copy the public IP of sonar-server machine and paste it to the browser to access the sonar server

Sonar-server public IP: 9000



* Configure Sonar syntax like below

stage('sonarscan'){

withSonarQubeEnv('sonarqube'){

def mavenHome = tool name: "Maven-3.8.6", type:"maven"

def mavenCMD = "${mavenHome}/bin/mvn"

sh "${mavenCMD} sonar:sonar"

* SonarQube project uploaded to Sonar portal

Step - 5) Nexus Artifactory

5.1) In AWS Portal, create a new instance as,

Allow: port number: 8081 (Nexus default port number)

5.3) Download nexus software

https://help.sonatype.com/repomanager3/product-information/download

5.4) Login to Nexus

5.5) Create Repository to store Artifacts

5.6) Download nexus Plugin from Manage Jenkins

5.7) Select pipeline syntax option

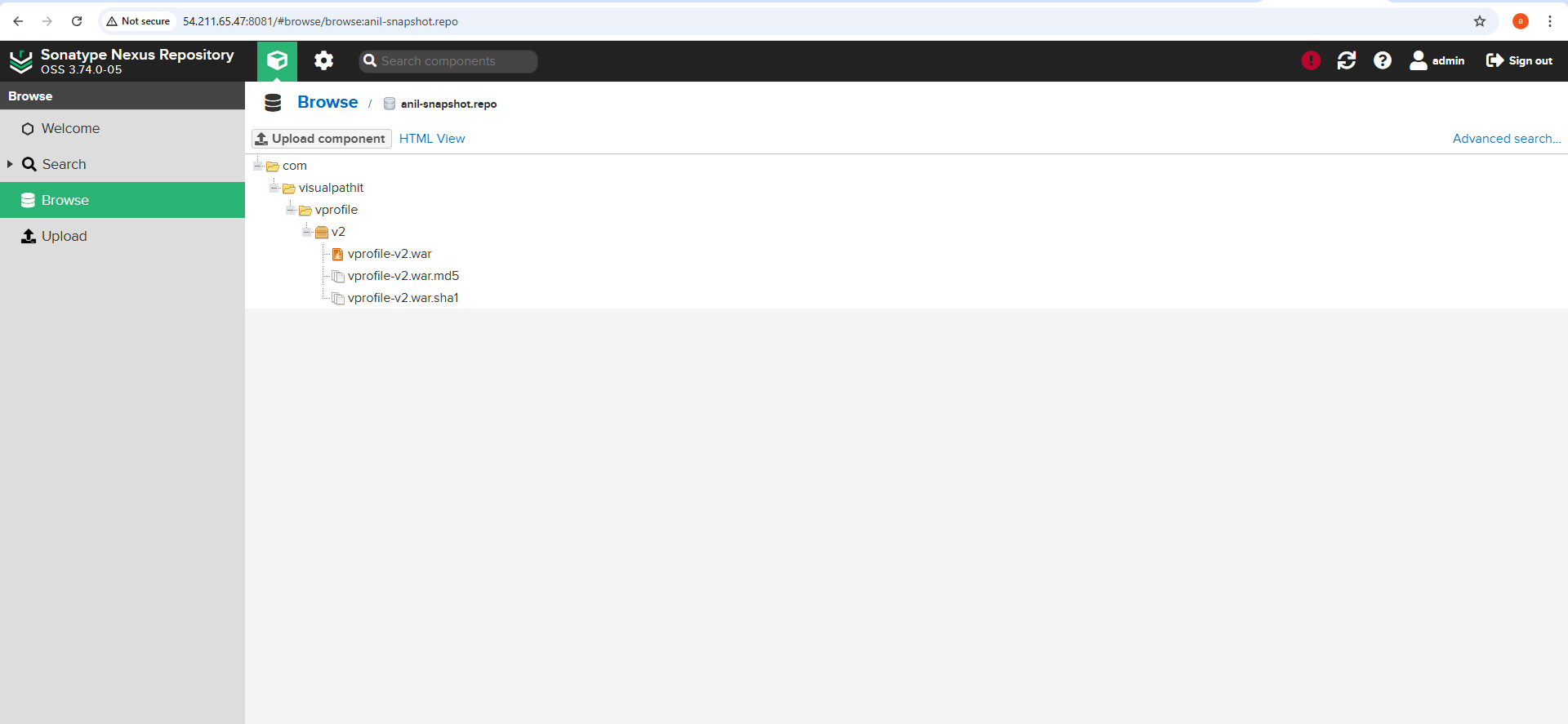
Copy pipeline syntax and paste in script like blow script

stage('artifacts-nexus'){

nexusArtifactUploader artifacts: [[artifactId: 'vprofile', classifier: '', file: 'target/vprofile-v2.war', type: 'war']], credentialsId: 'nexus', groupId: 'com.visualpathit', nexusUrl: '54.172.86.255:8081/', nexusVersion: 'nexus3', protocol: 'http', repository: 'anil-snapshot.repo', version: 'v2'

}

5.9) Artifacts uploaded to Nexus Portal



Step - 6) Tomcat Deploy

6.1) In AWS Portal, create a new instance

Allow: port number: 8080 (Nexus default port number)

6.3) Download tomcat server

https://tomcat.apache.org/download-90.cgi

6.4) Login into Tomcat server

6.5) Tomcat plugin installation go to Jenkins page

6.6) Tomcat syntax

stage('Final-deploy'){

sshagent(['tomcatnew']) {

sh "scp -o StrictHostKeyChecking=no target/vprofile-v2.war ec2-user@34.224.223.128:/opt/apache-tomcat-9.0.97/webapps"

}

}

6.7) Tomcat build success

6.8) Successfully Project deployed to tomcat server

Click on project name

