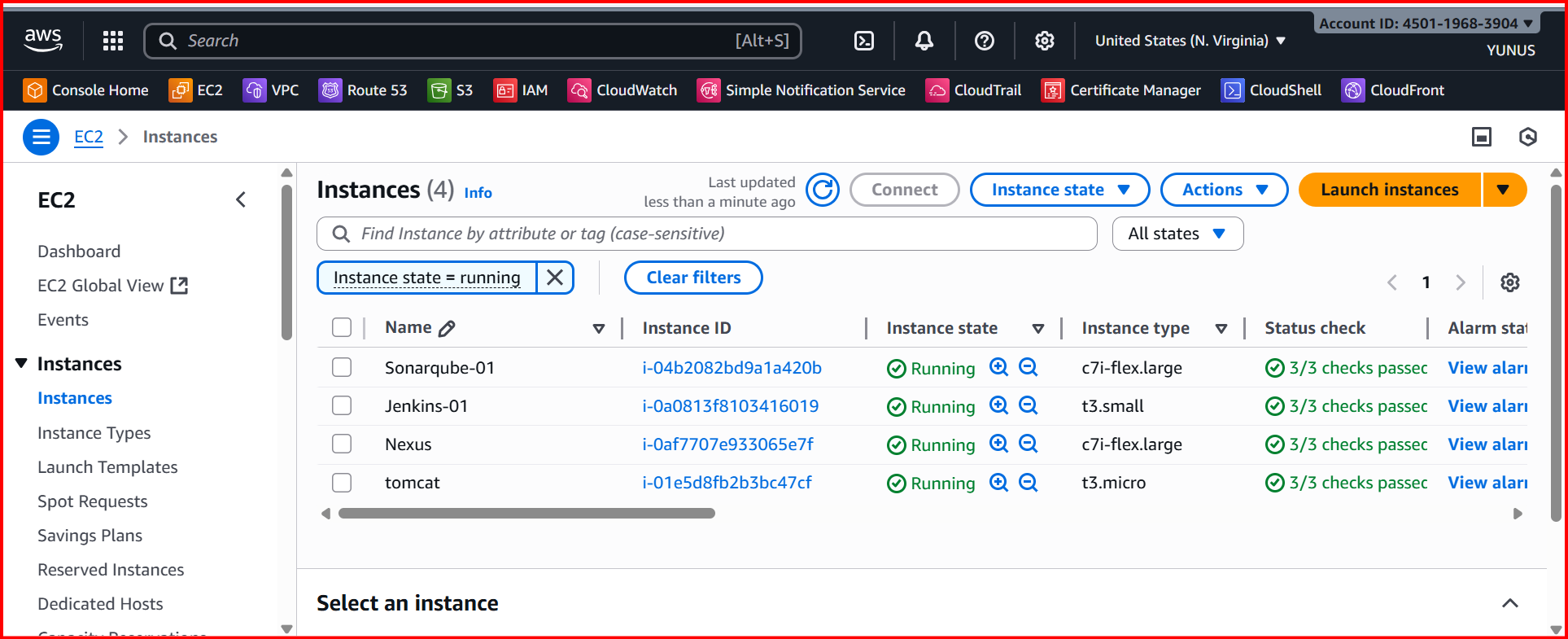
**Jenkins Task -4**

**1) Setup jenkins CICD pipeline using freestyle job using Docker containers using below code.**[**https://github.com/betawins/hiring-app.git**](https://github.com/betawins/hiring-app.git) **Stages:  
   1) Git Clone  
   2) Sonarqube Integration  
   3) Maven Compilation  
   4) Nexus Artifactory  
   5) Slack Notification  
   6) Deploy On tomcat**

**Launch 4 Instances**

****

**IN JENKINS SERVER**

sudo dnf install -y java-17-amazon-corretto

java -version

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

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

Imports Jenkins signing key.

Ensures that the packages downloaded from Jenkins repo are trusted and not

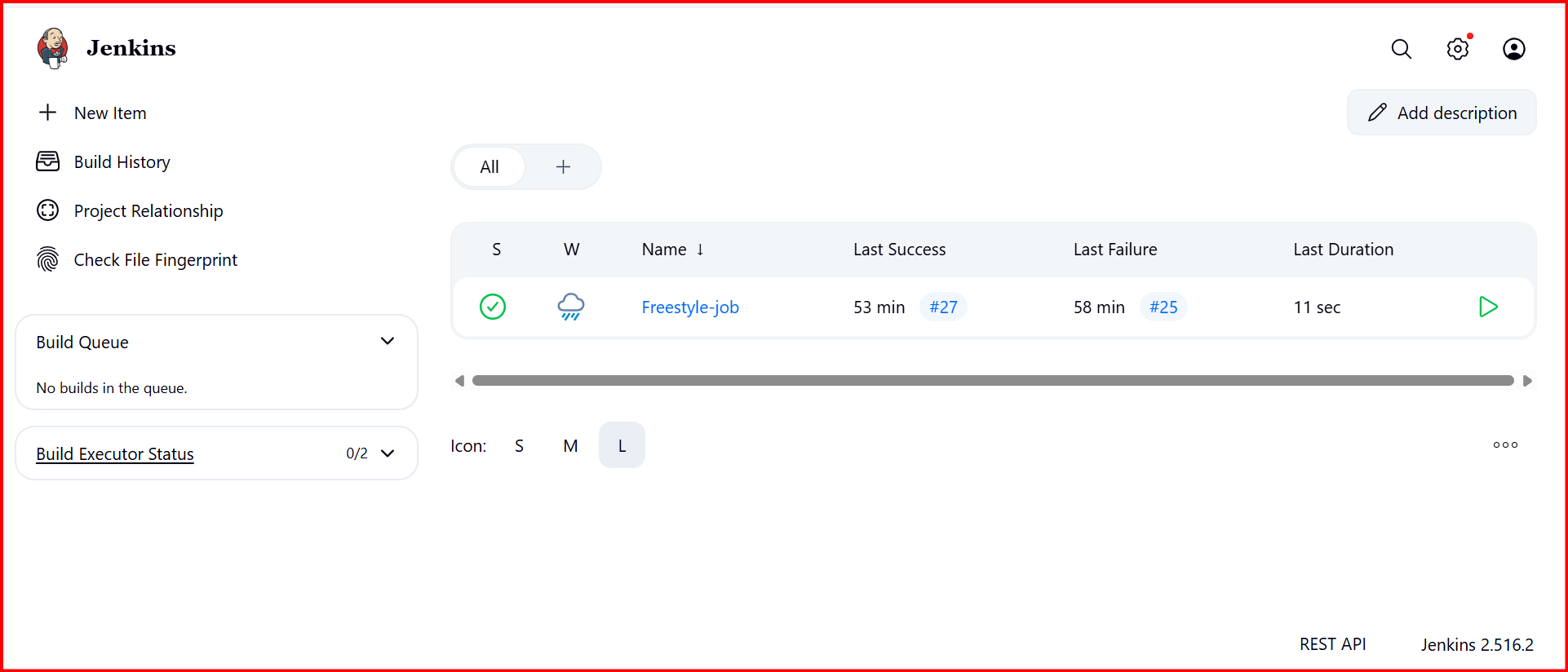
sudo yum clean all

sudo yum install jenkins -y

systemctl start jenkins

systemctl status Jenkins

<http://3.84.81.205:8080/>



**In Sonarqube server install sonarqure**

BY USING DOCKER INSTAED SONARQUB

Sudo Su-

yum -y install docker

systemctl start docker

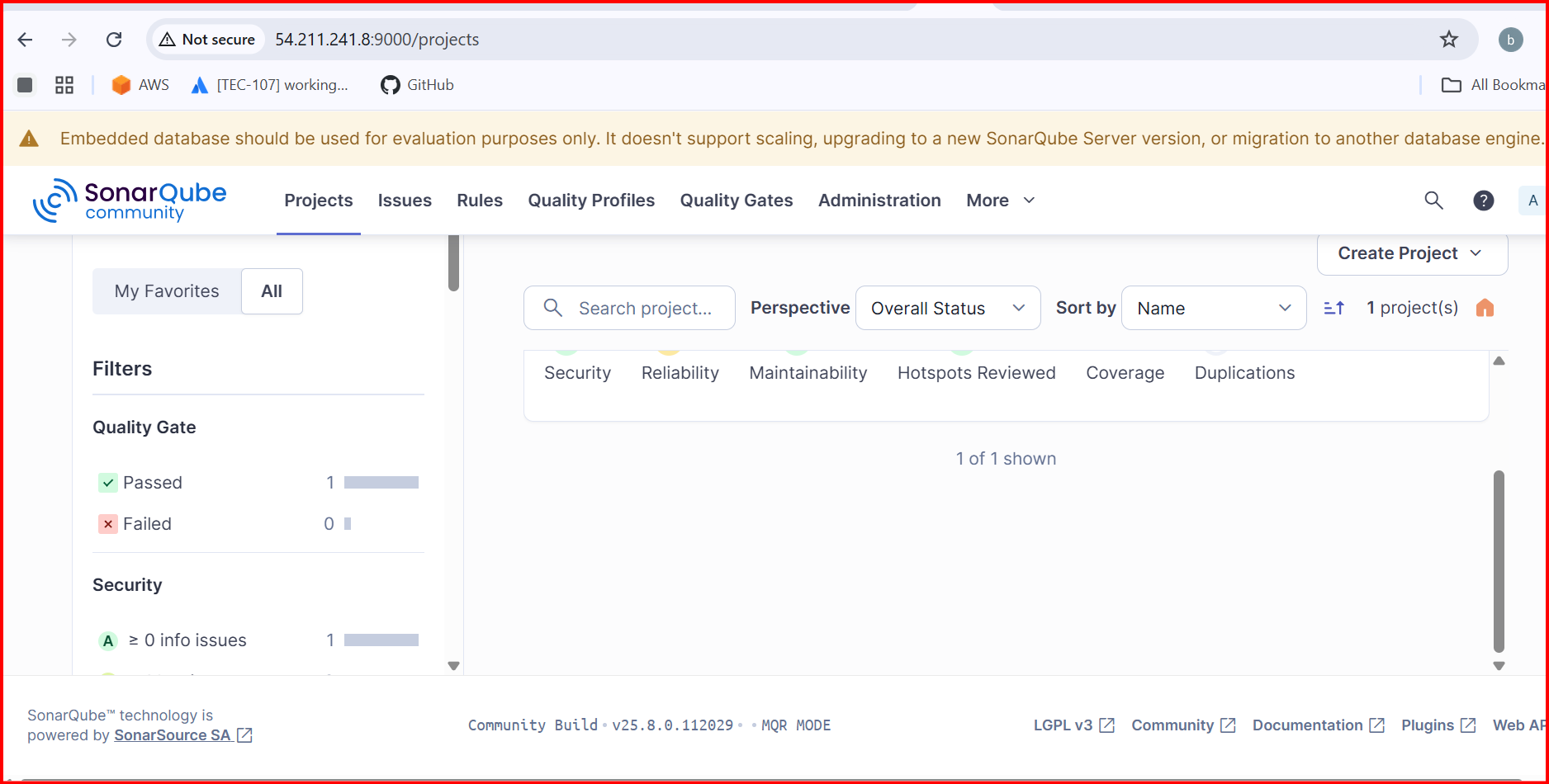
docker pull sonarqube

docker images

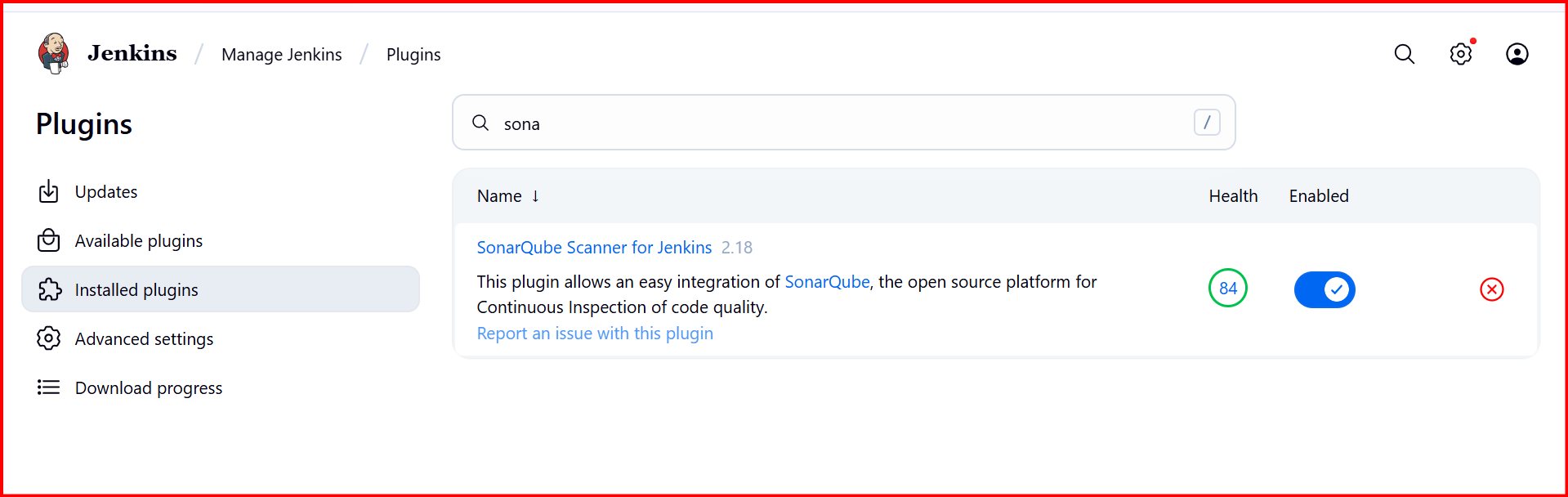
docker run -itd --name sonarqube -p 9000:9000 sonarqube

docker ps

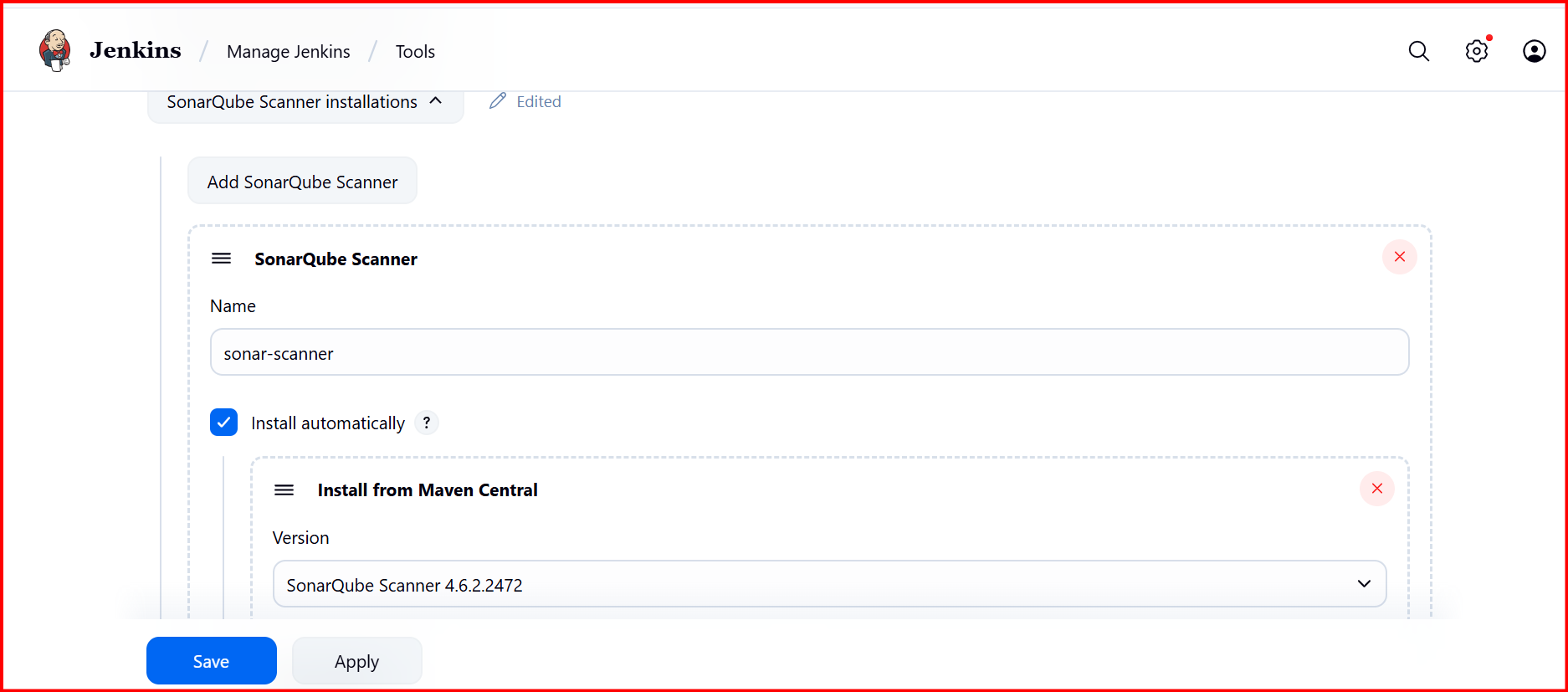
<http://54.211.241.8:9000/>



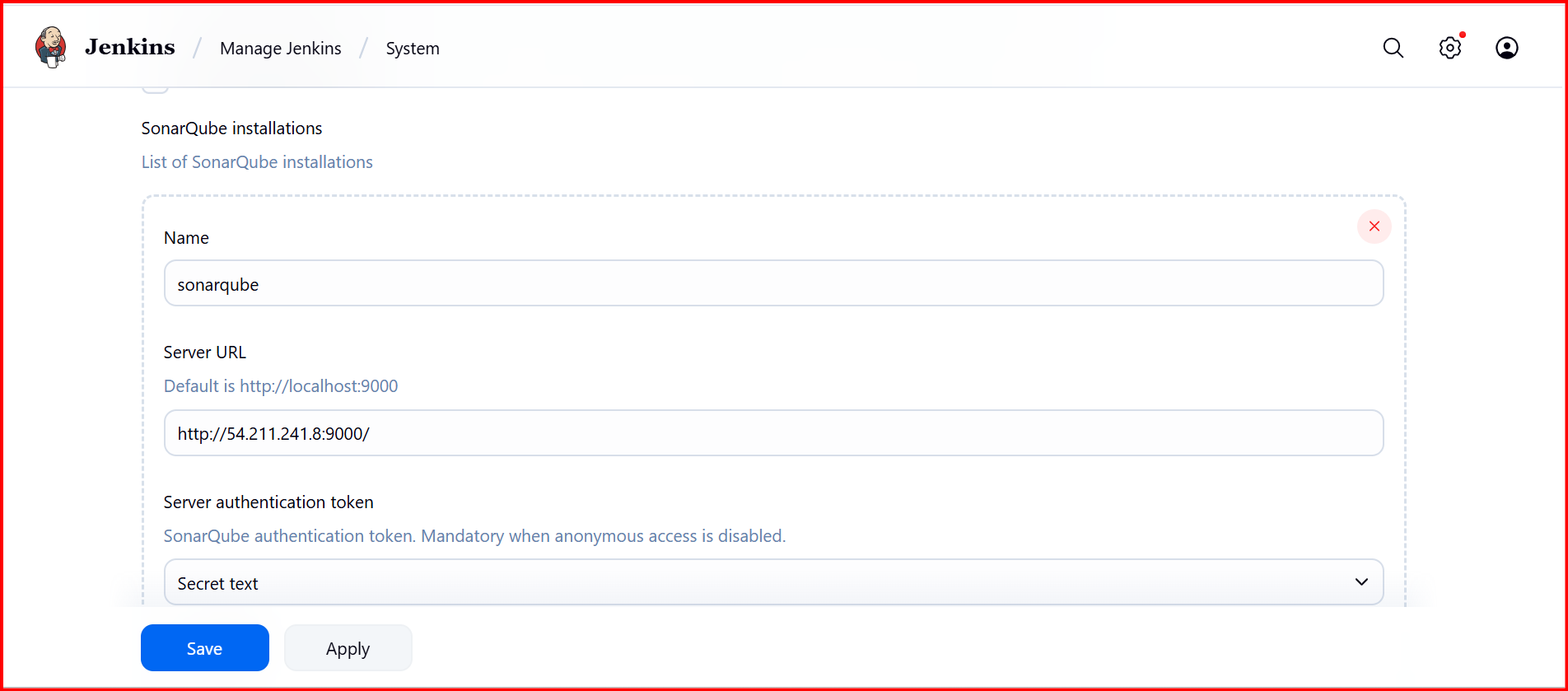
**Install Sonarscanner plugin**



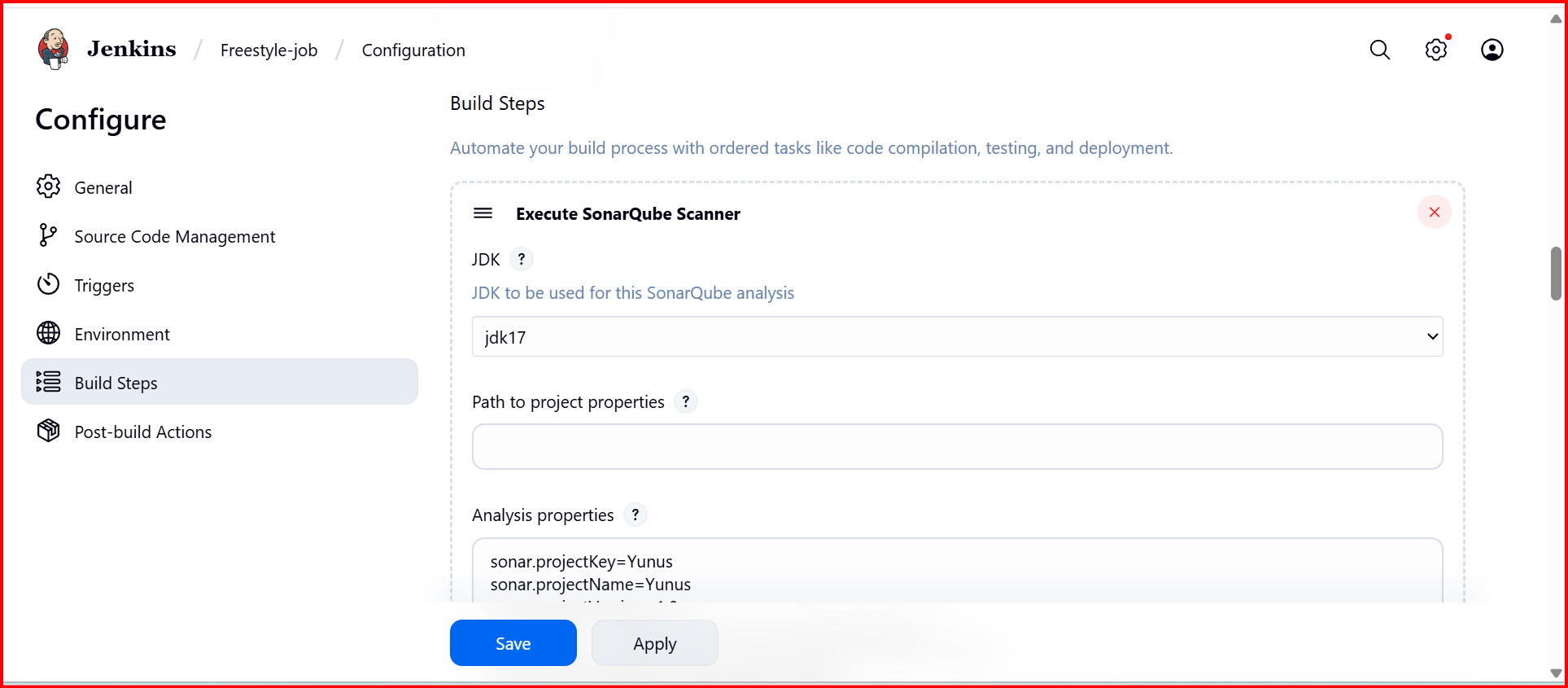
**Configure Jenkins to sonarscanner**

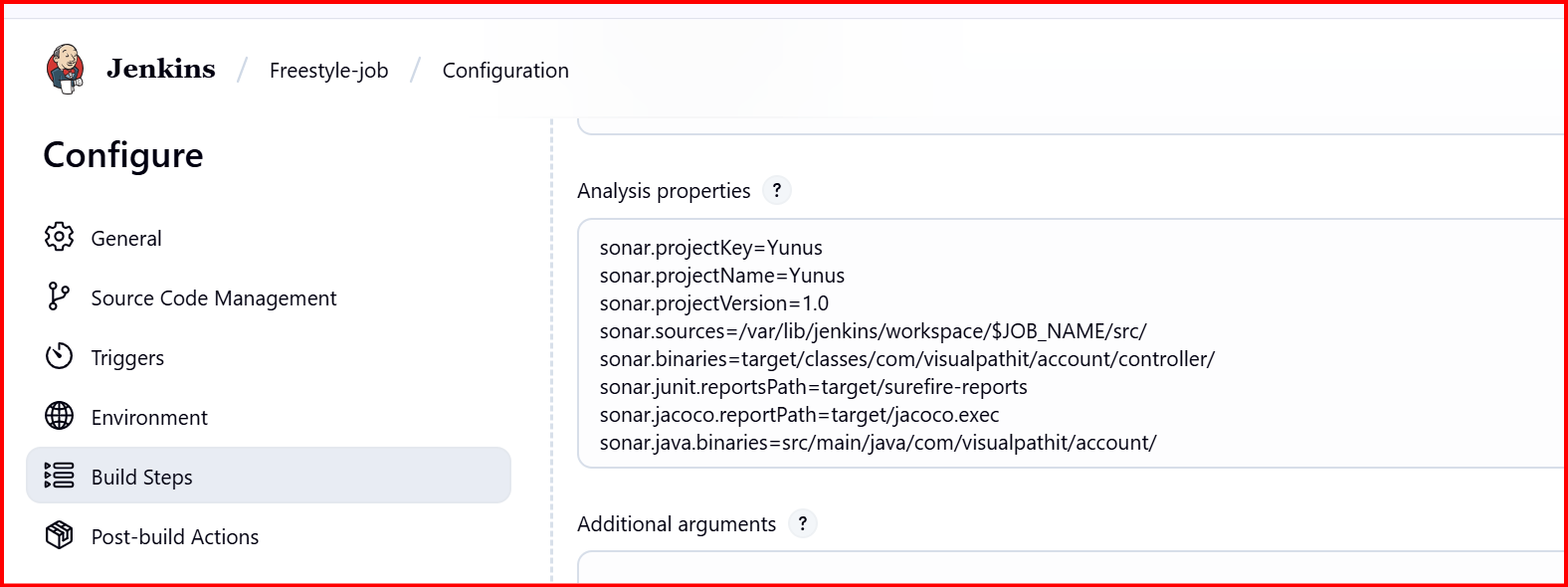


**Configure Jenkins to sonarqube**



In Job configuration





**I have Install maven in Jenkins server**

cd /opt/

ls

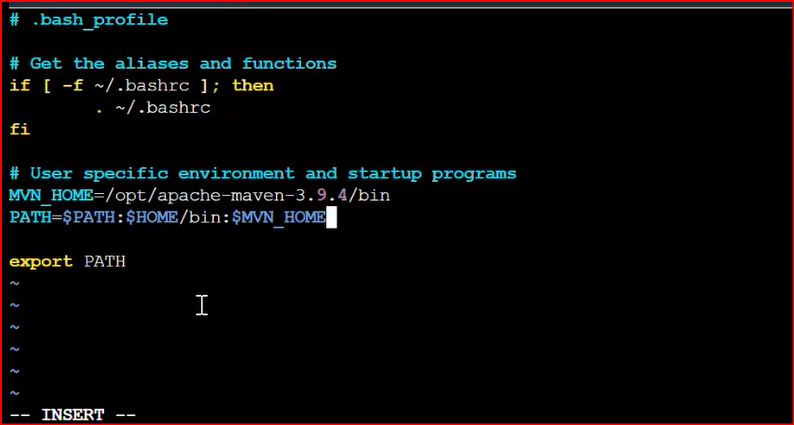
google maven download click first link select tar.gz

wget https://dlcdn.apache.org/maven/maven-3/3.9.11/binaries/apache-maven-3.9.11-bin.tar.gz

tar xvf apache-maven-3.9.11-bin.tar.gz



Edit this file

****

**Go to this location 🡪/opt/apache-maven-3.9.11/bin**

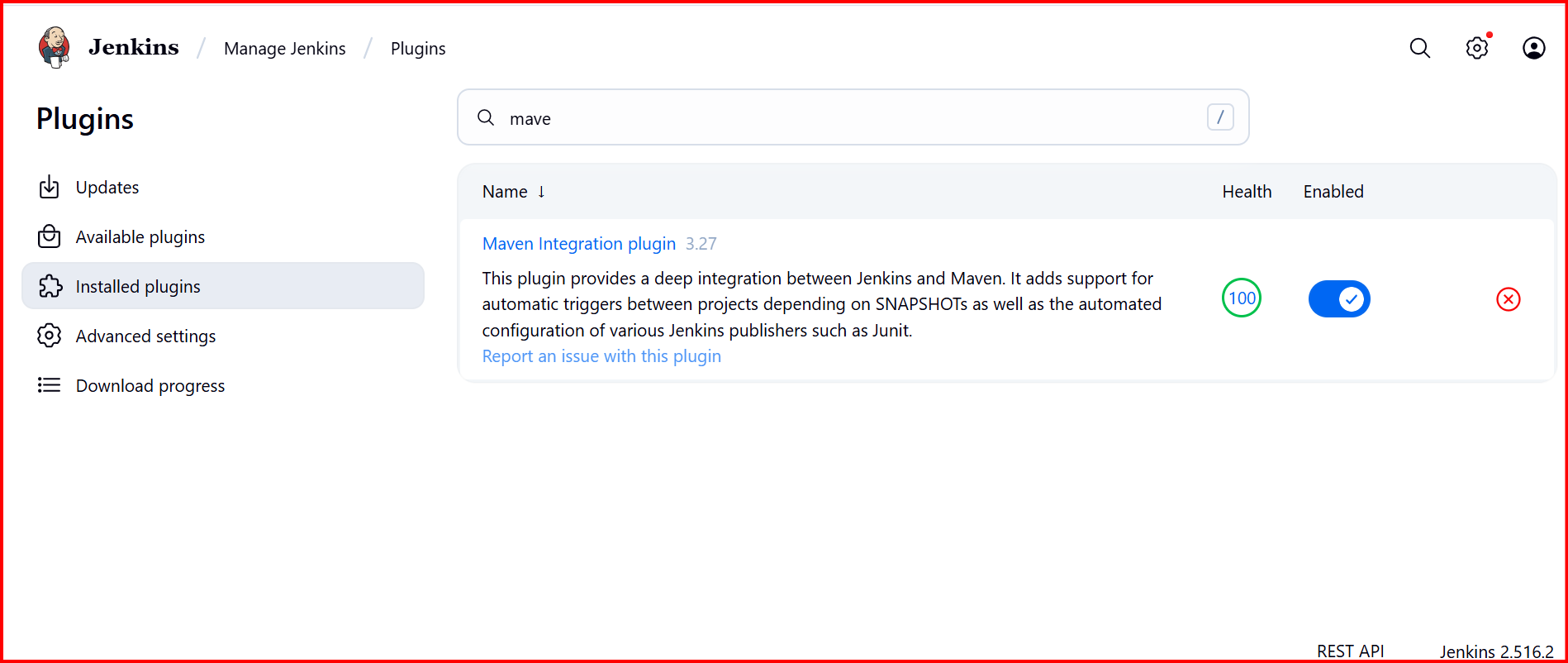
**source ~/.bash\_profile --> RESTART THE FILE**

**echo $MVN\_HOME --> TO FIND THE PATH**

**mvn –version**

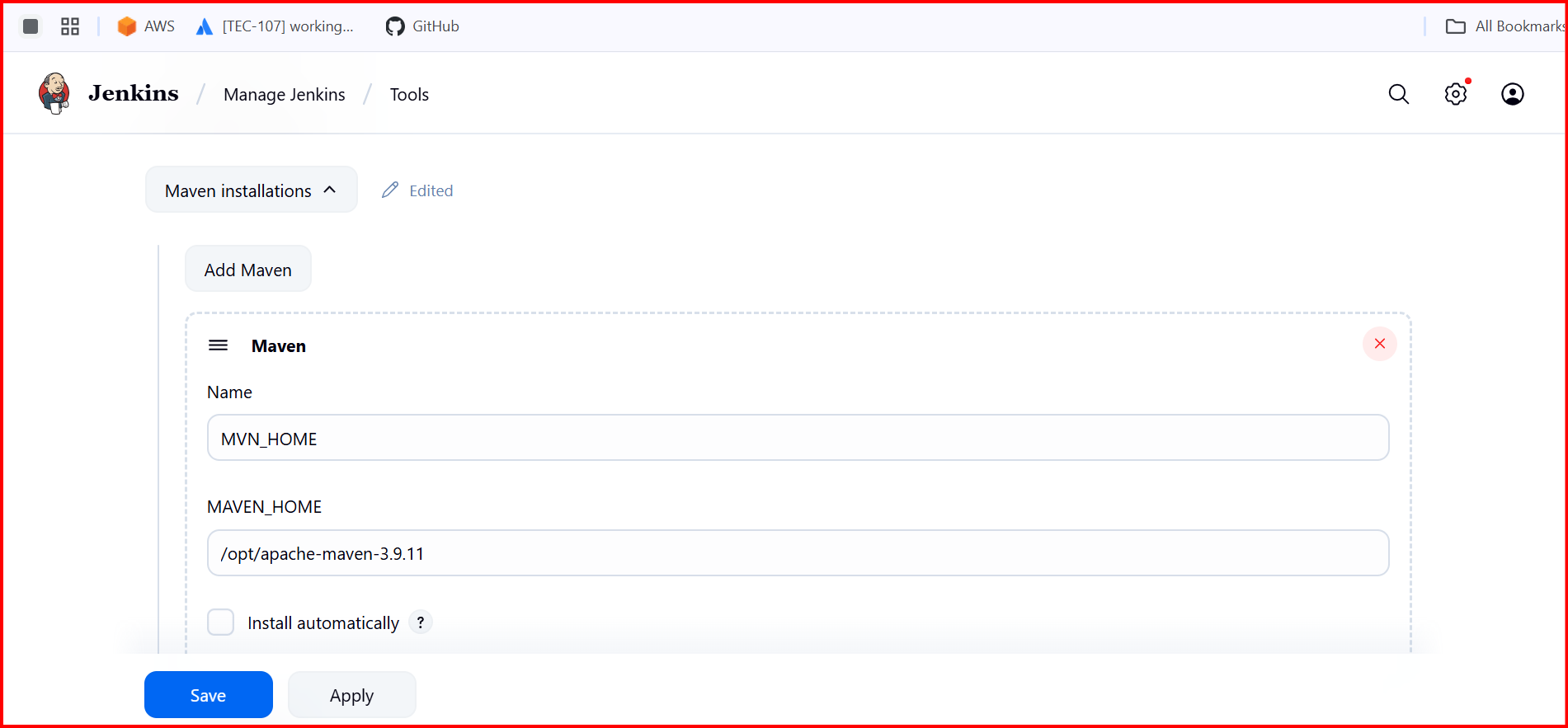
****

**Install Maven plugin**

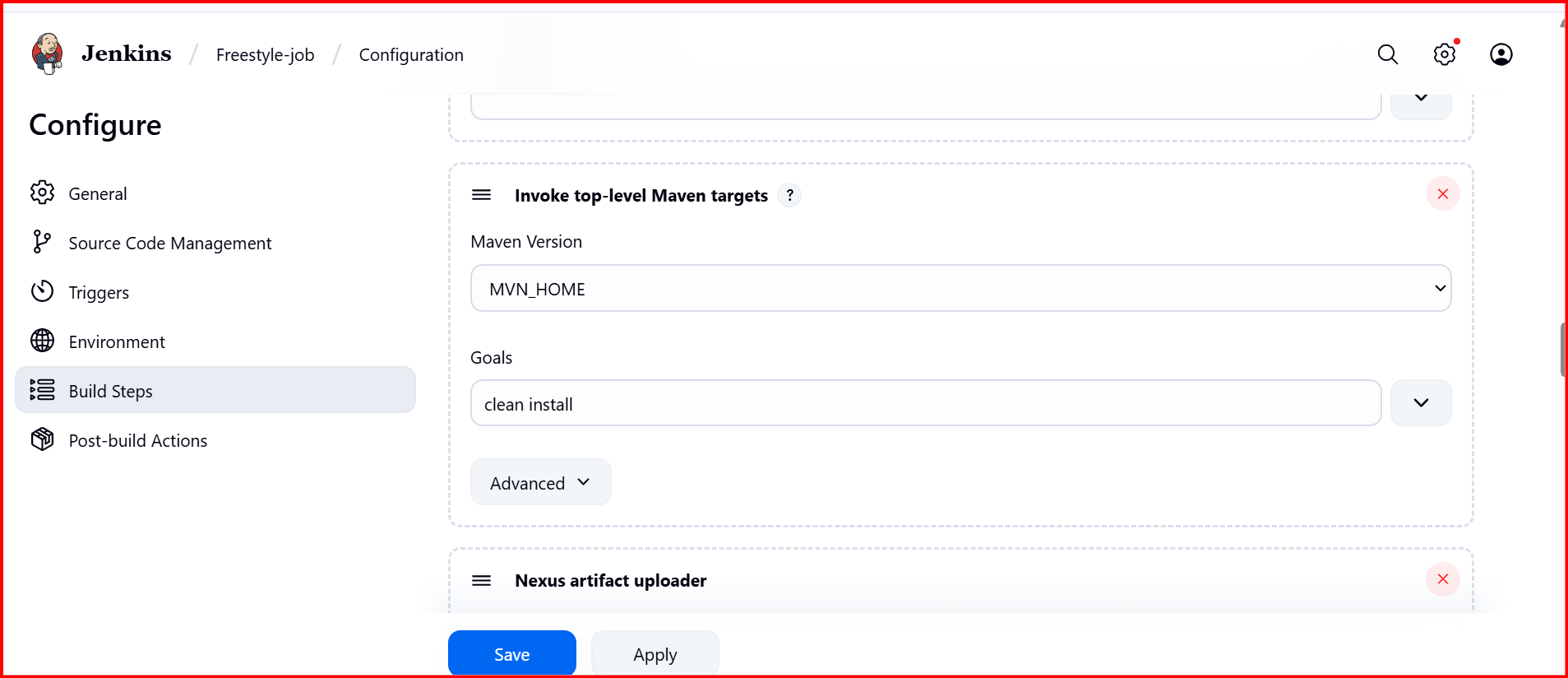
****

**Configure the maven in Jenkins**

**Give the maven path**

****

**In job configurations**

****

**Nexus**

**wget https://sonatype-download.global.ssl.fastly.net/nexus/3/nexus-3.0.2-02-unix.tar.gz**

**tar -zxvf nexus-3.0.2-02-unix.tar.gz**

**mv /opt/nexus-3.0.2-02 /opt/nexus**

**cd nexus/**

**vi nexus.rc**

**run\_as\_user="ec2-user"**

**exit**

**cd**

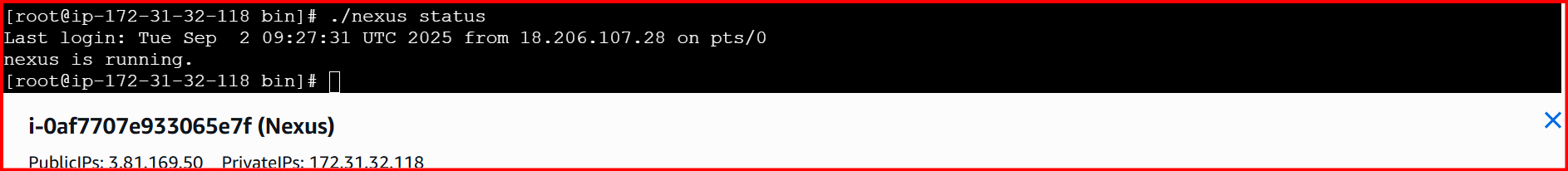
**sudo chown -R ec2-user:ec2-user nexus**

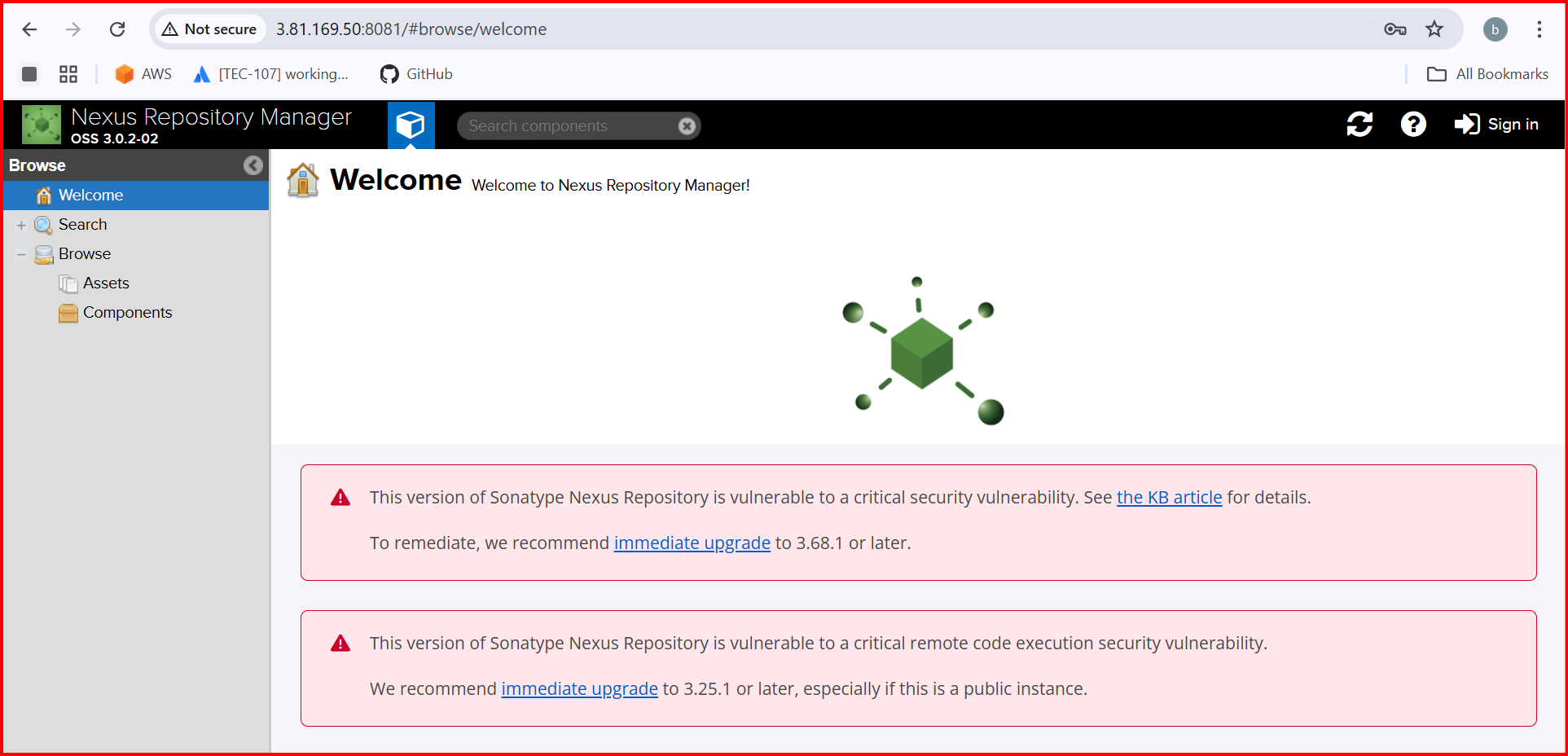
**cd nexus/**

**cd bin/**

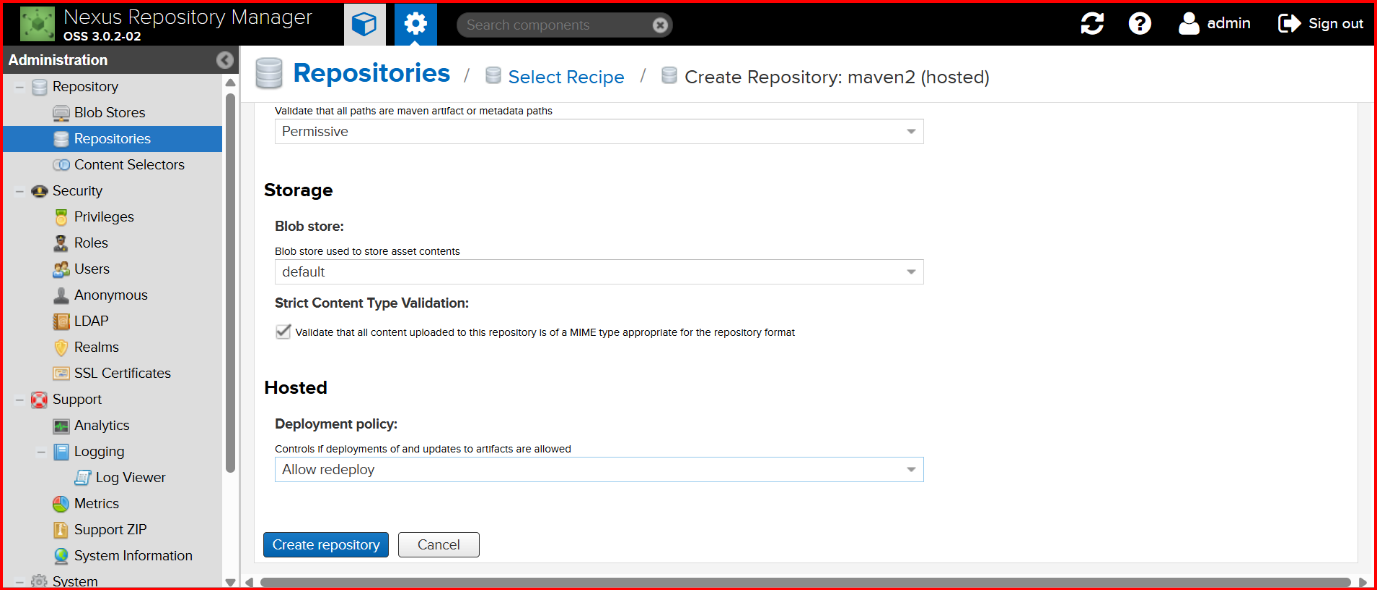
**./nexus start**

**./nexus status** [**http://3.81.169.50:8081/**](http://3.81.169.50:8081/)

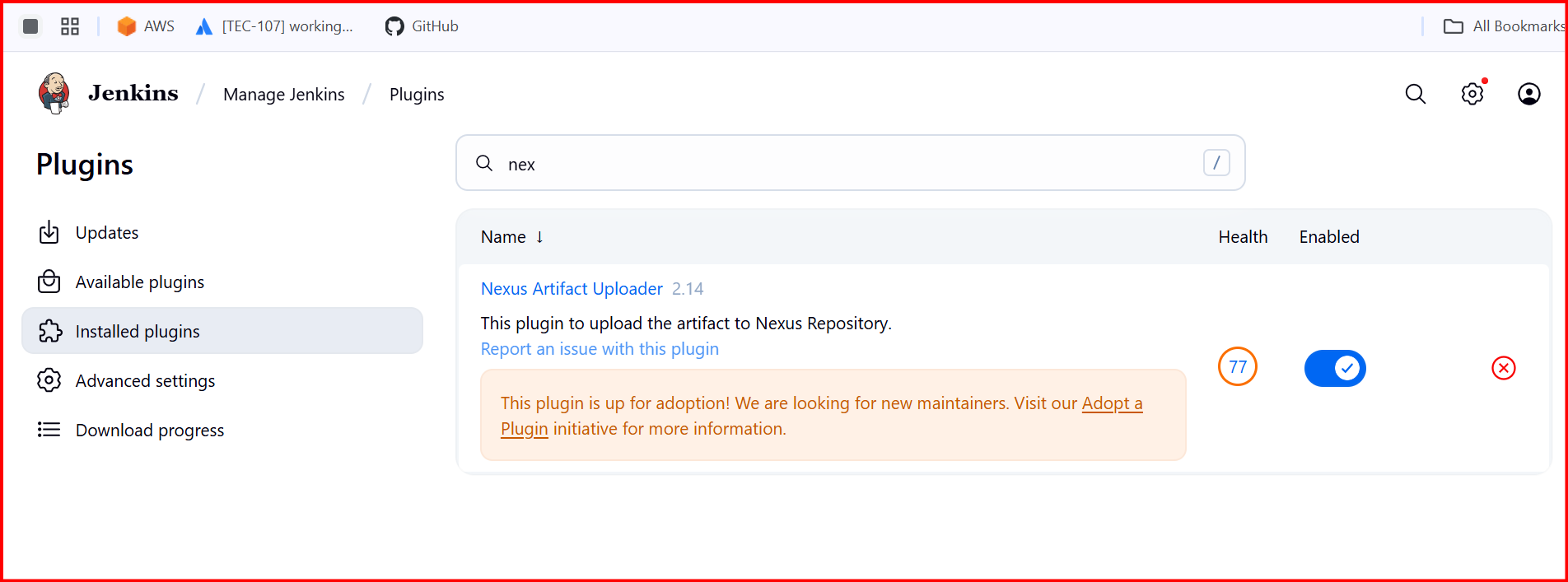
****

****

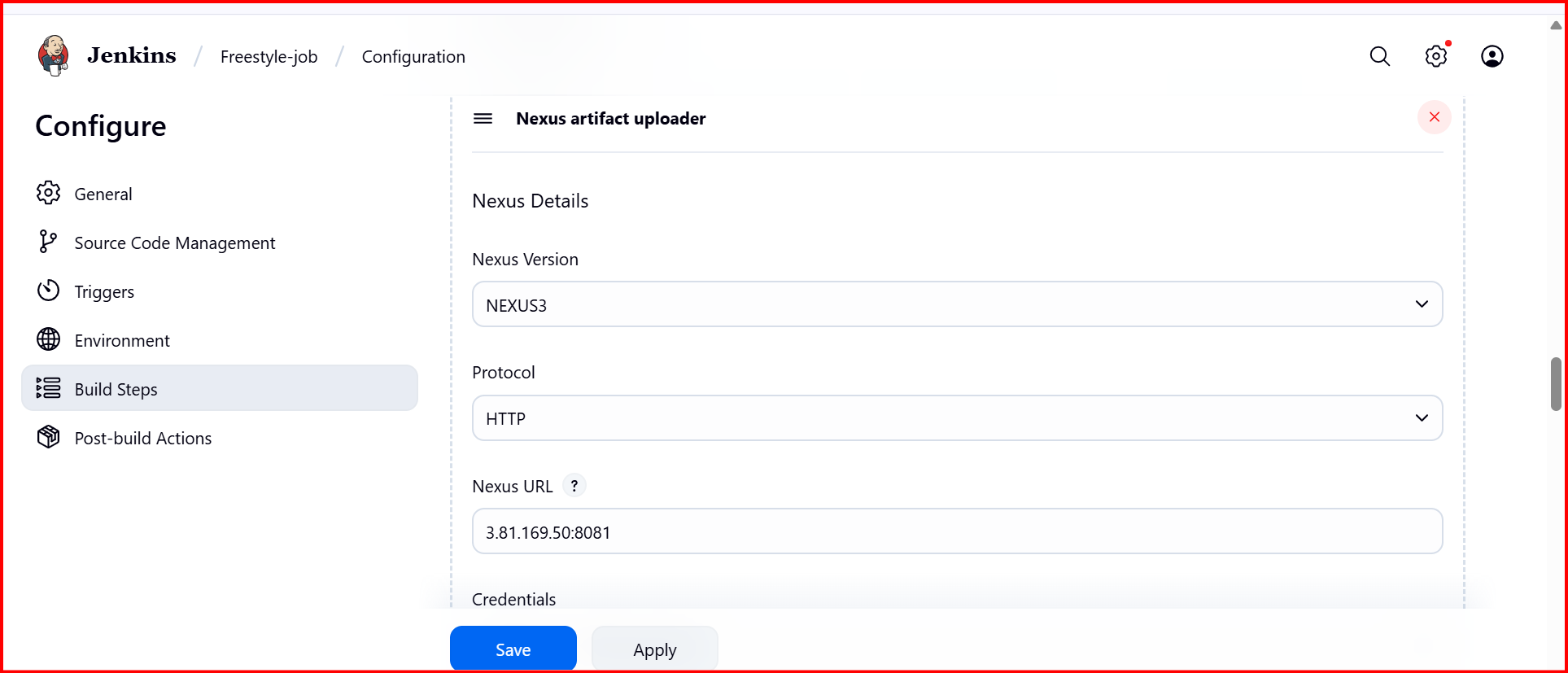
**Nexus --> Settings --> Repositories --> create repo --> Maven2hosted -->**

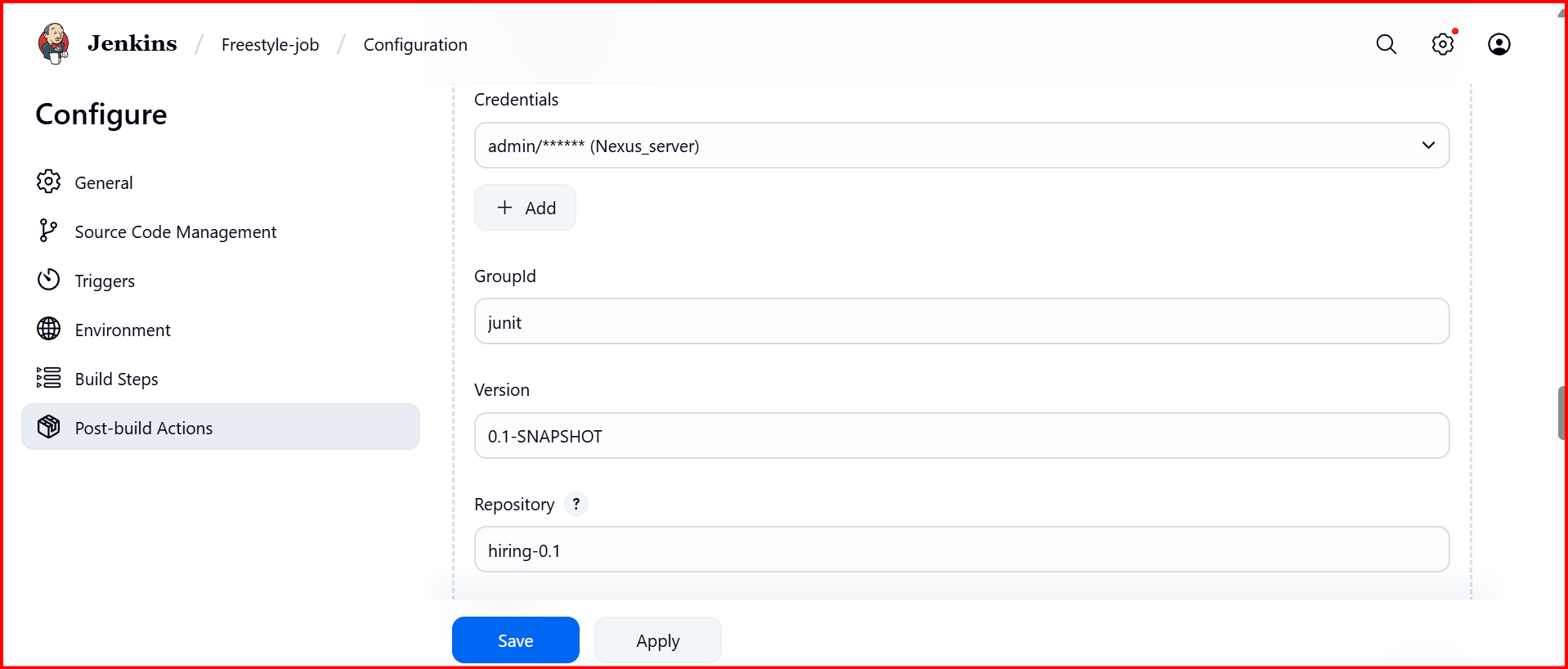
****

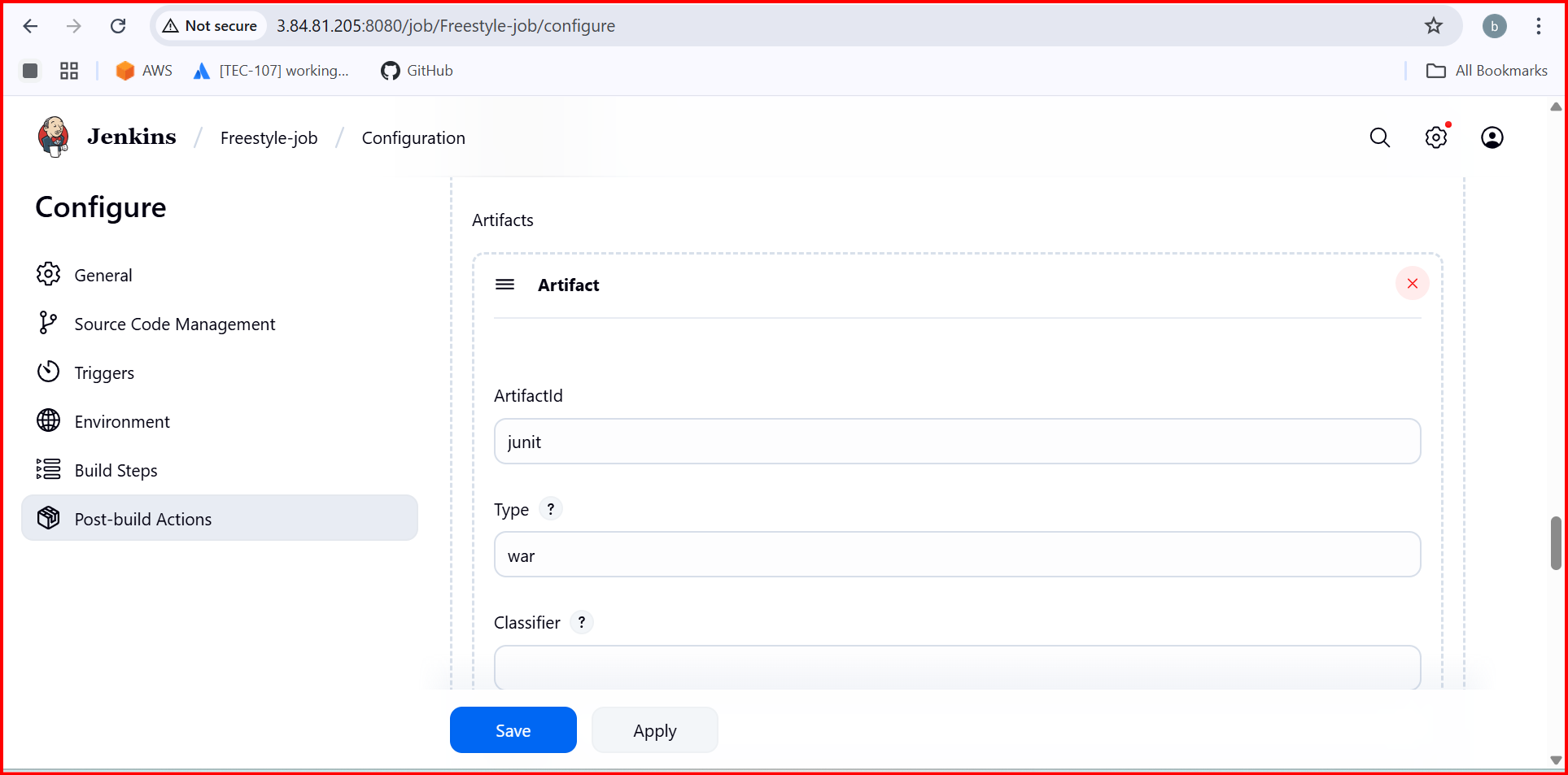
**Install nexus plugin**

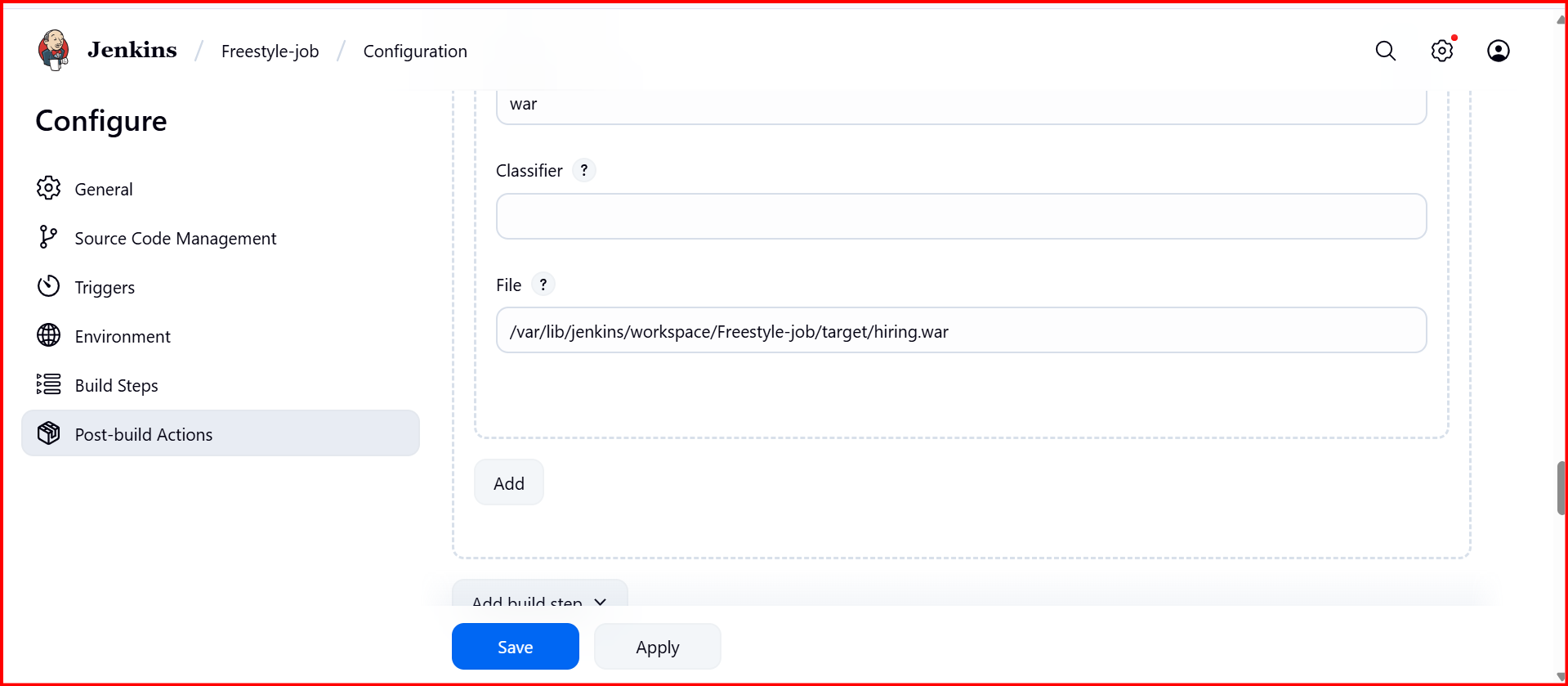
****

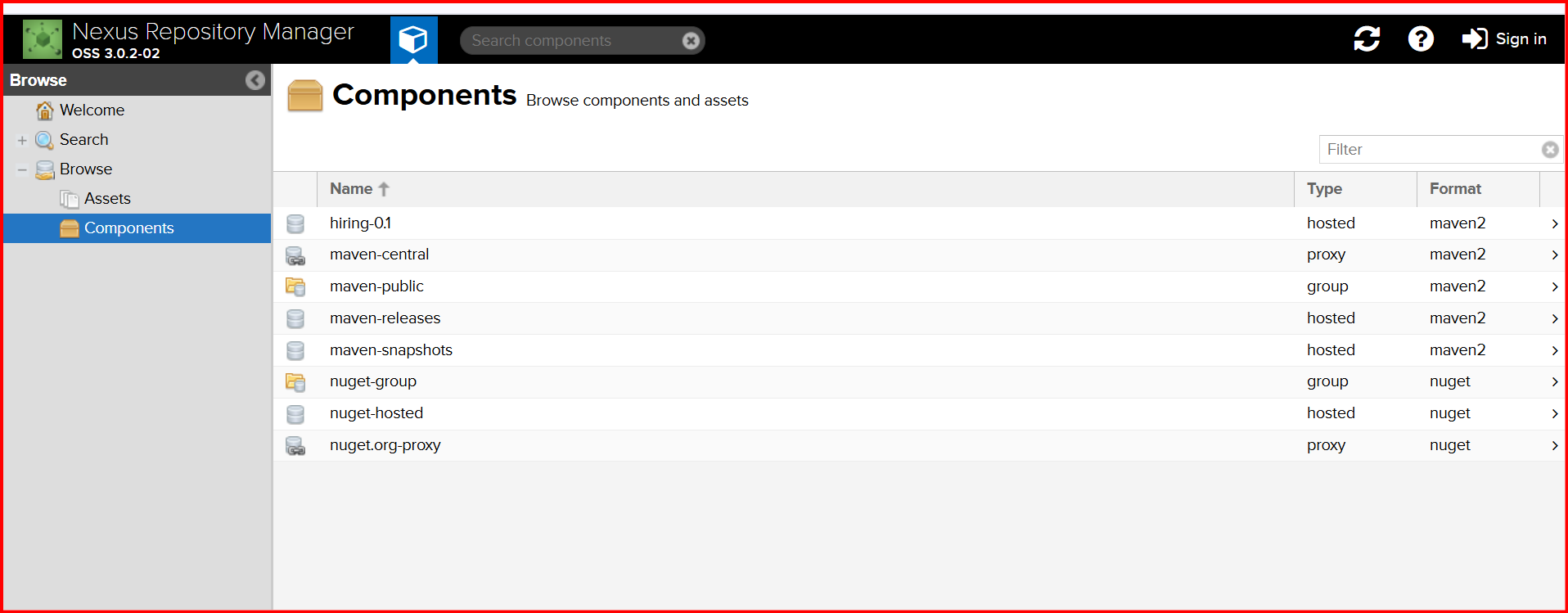
**Job configuration**

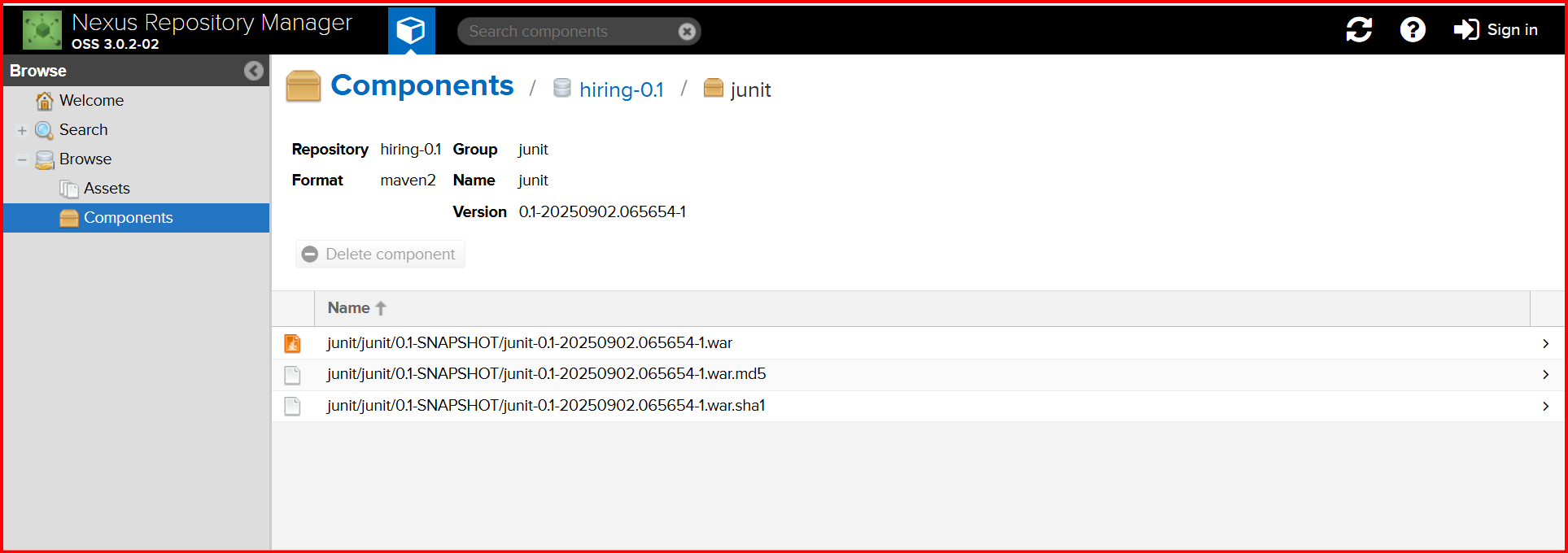
****

****

****

****

****

****

**Tomcat install in Tomcat server**

**yum install java-1.8\***

**wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.108/bin/apache-tomcat-9.0.108.tar.gz**

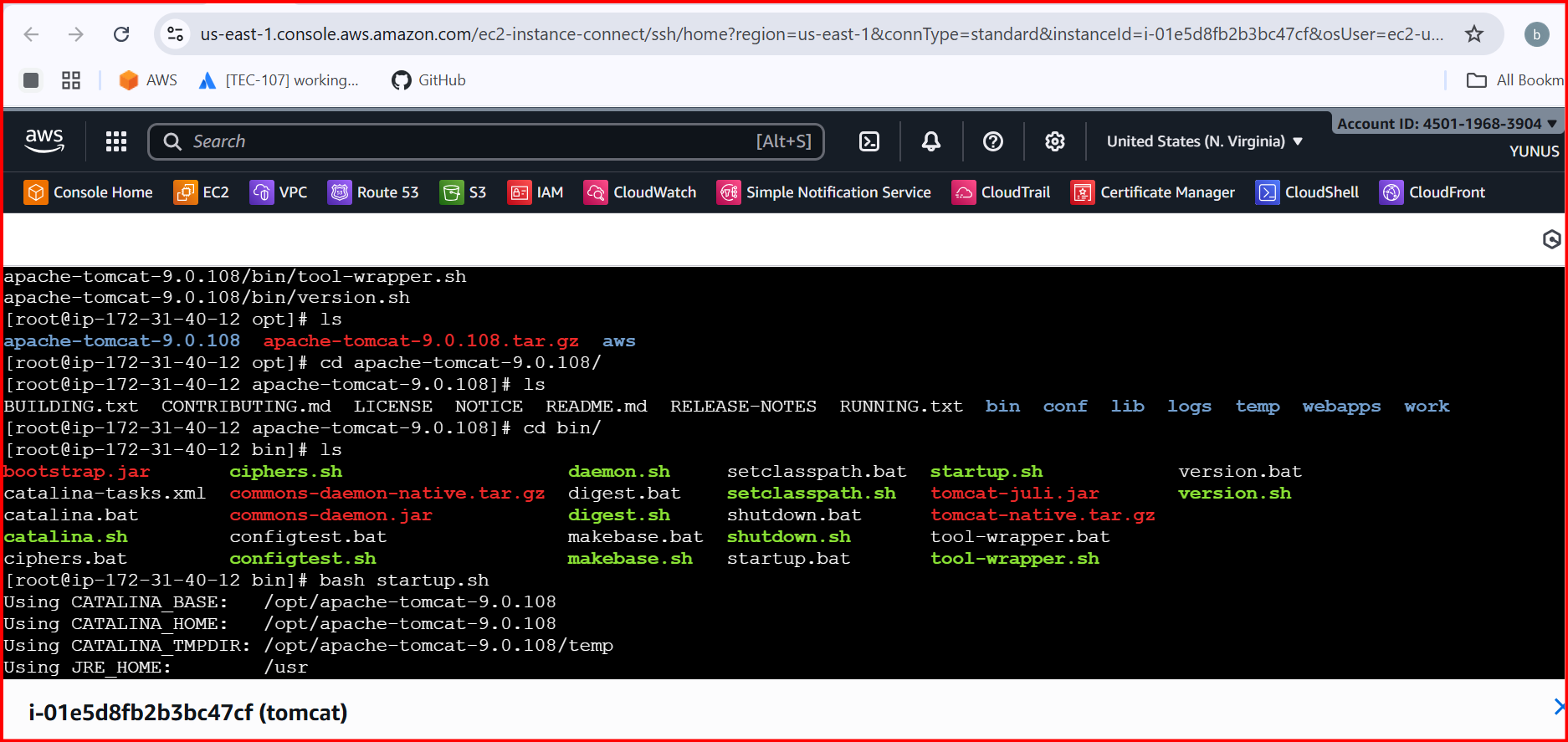
**tar xvf apache-tomcat-9.0.108.tar.gz**

**cd apache-tomcat-9.0.108/**

**cd bin/**

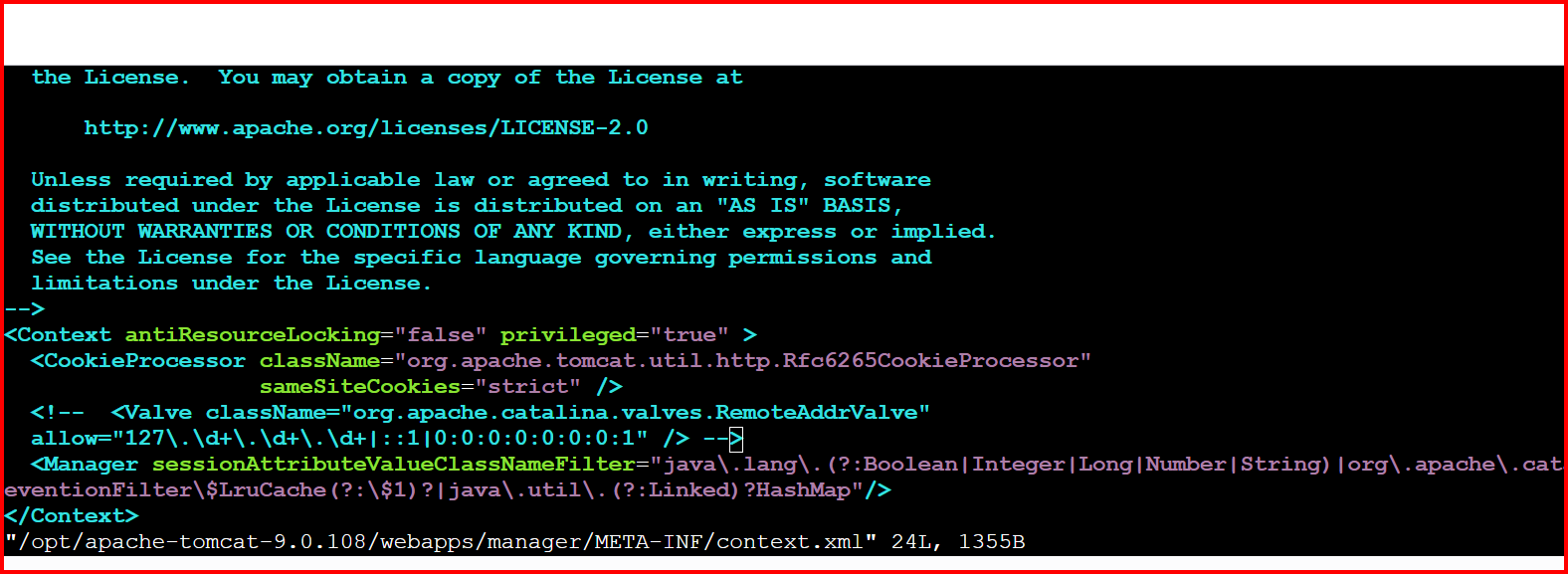
**ls**

**bash startup.sh**

**tar xvf apache-tomcat-9.0.108.tar.g**

**find / -name context.xml**

**4 files will be avalible we can edit the files file value there <!-- -->**

****

**Find Valve then edit <!-- last -- > like this do for all the steps**

**[root@ip-172-31-40-12 apache-tomcat-9.0.108]# cd conf/**

**vi tomcat-users.xml**

**<role rolename="manager-gui"/>**

**<role rolename="manager-script"/>**

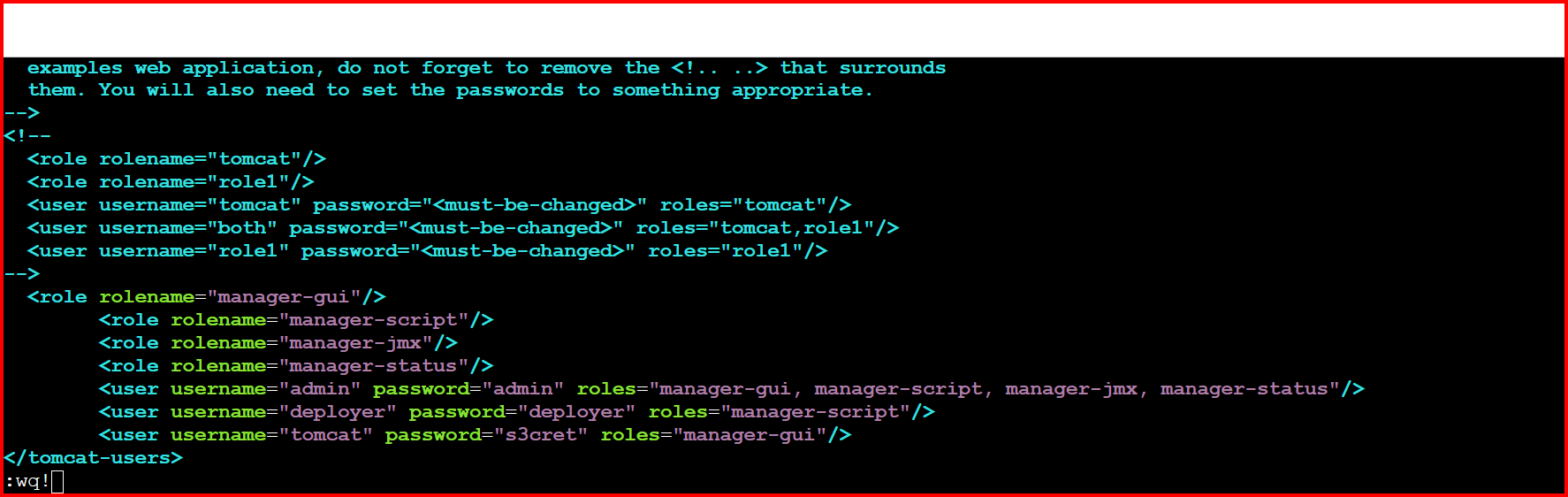
**<role rolename="manager-jmx"/>**

**<role rolename="manager-status"/>**

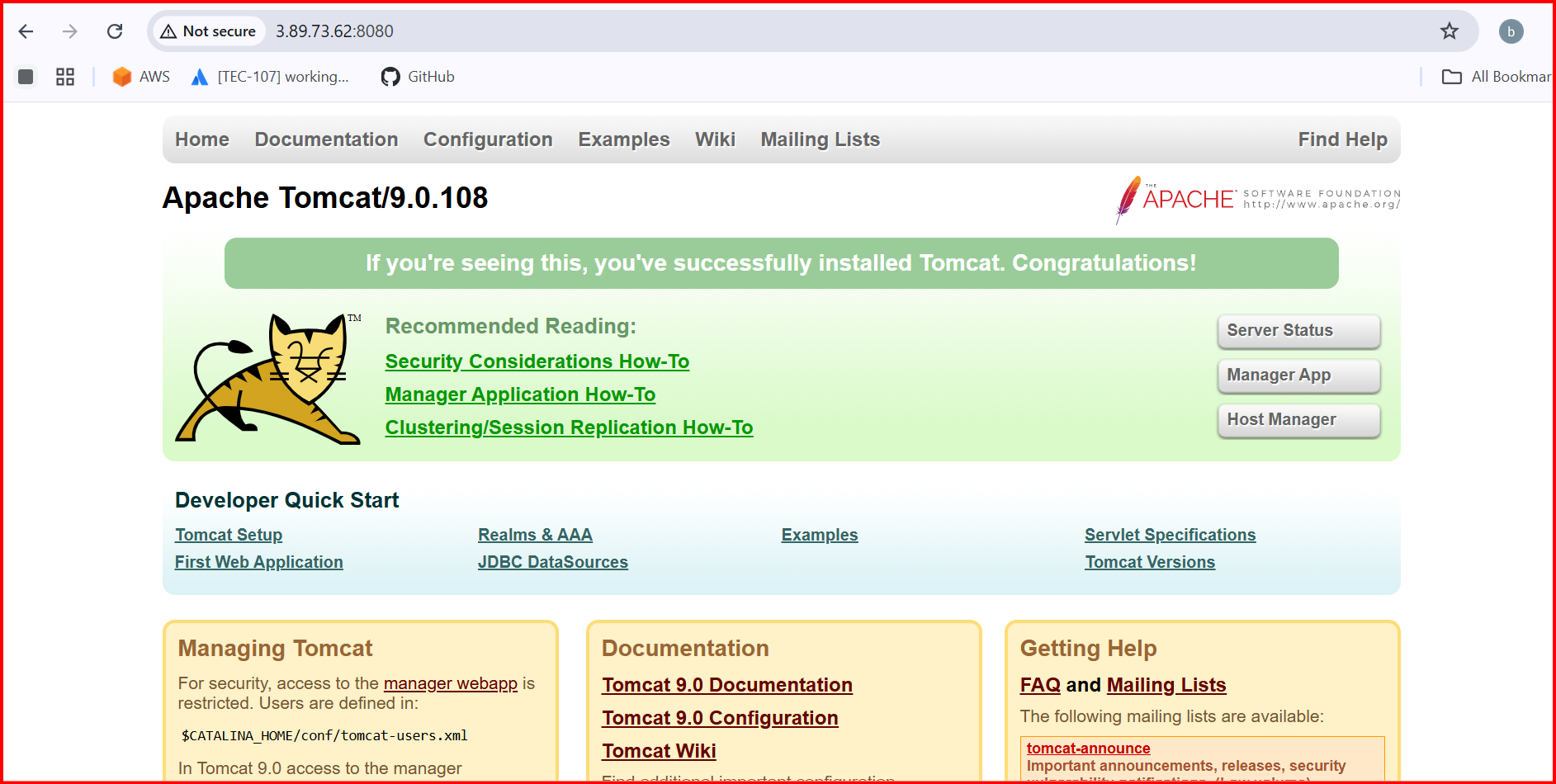
**<user username="admin" password="admin" roles="manager-gui, manager-script, manager-jmx, manager-status"/>**

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

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

****

[**http://3.89.73.62:8080/**](http://3.89.73.62:8080/)

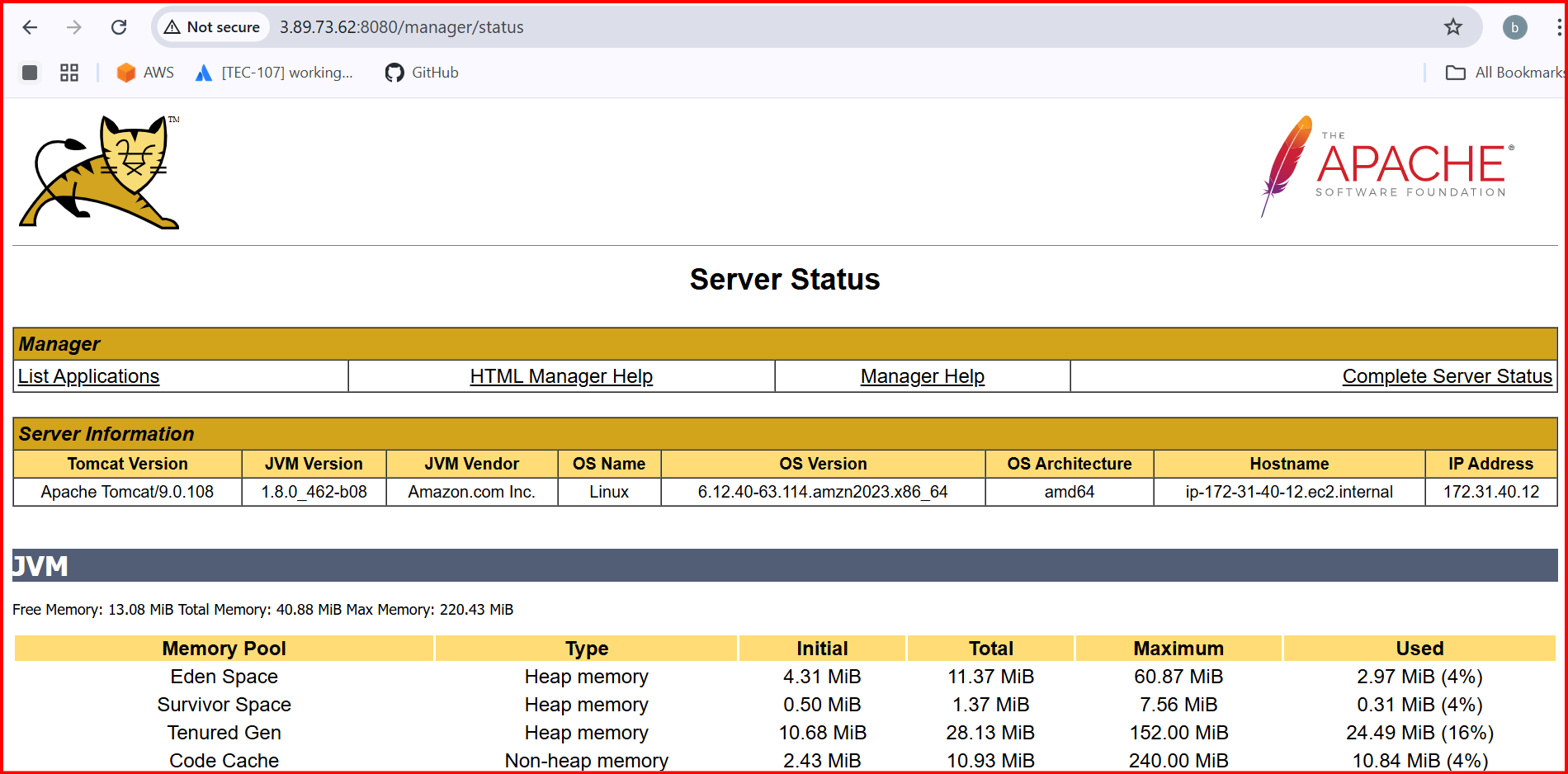
****

**Click any thing it will ask login password**

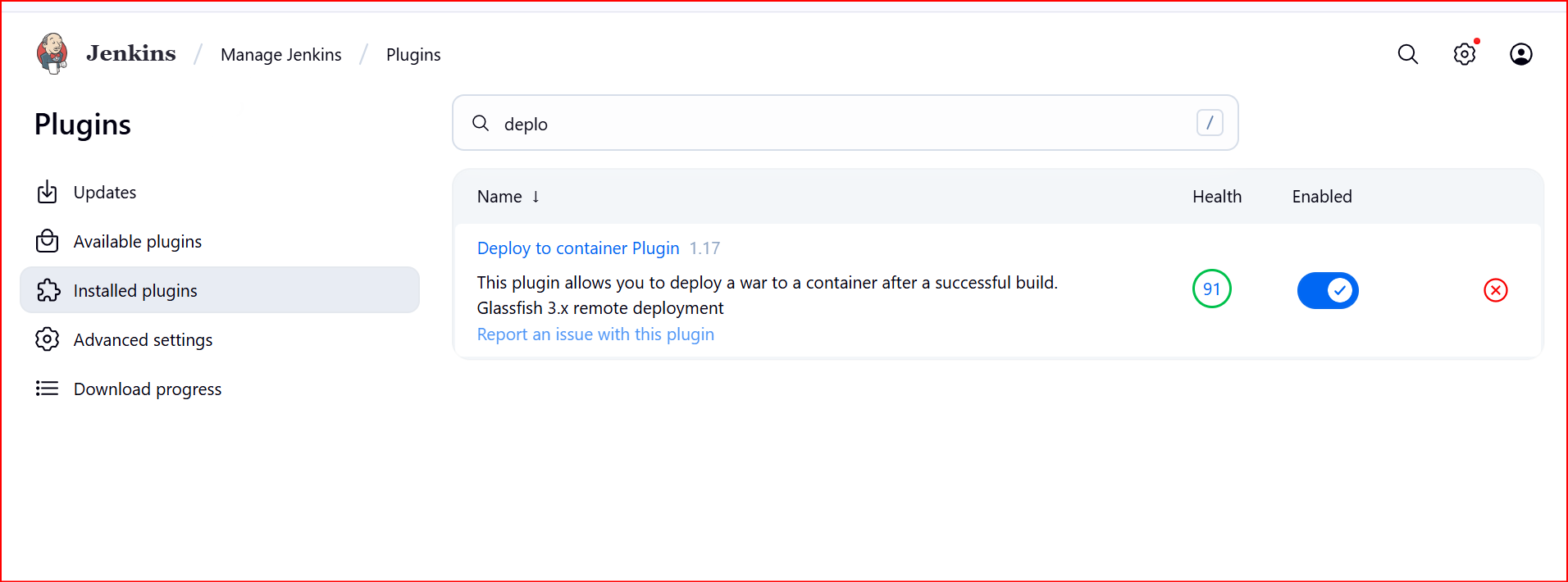
****

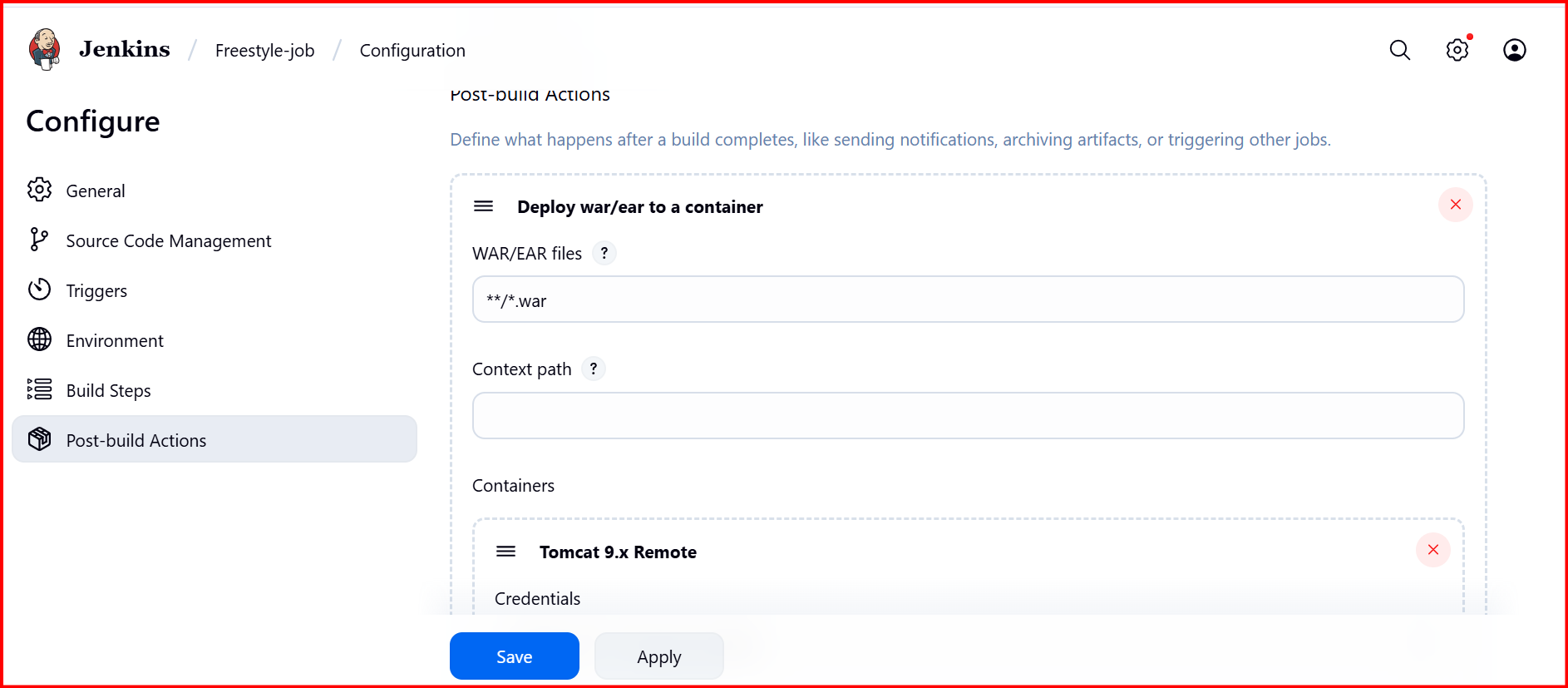
**Username :** **deployer**

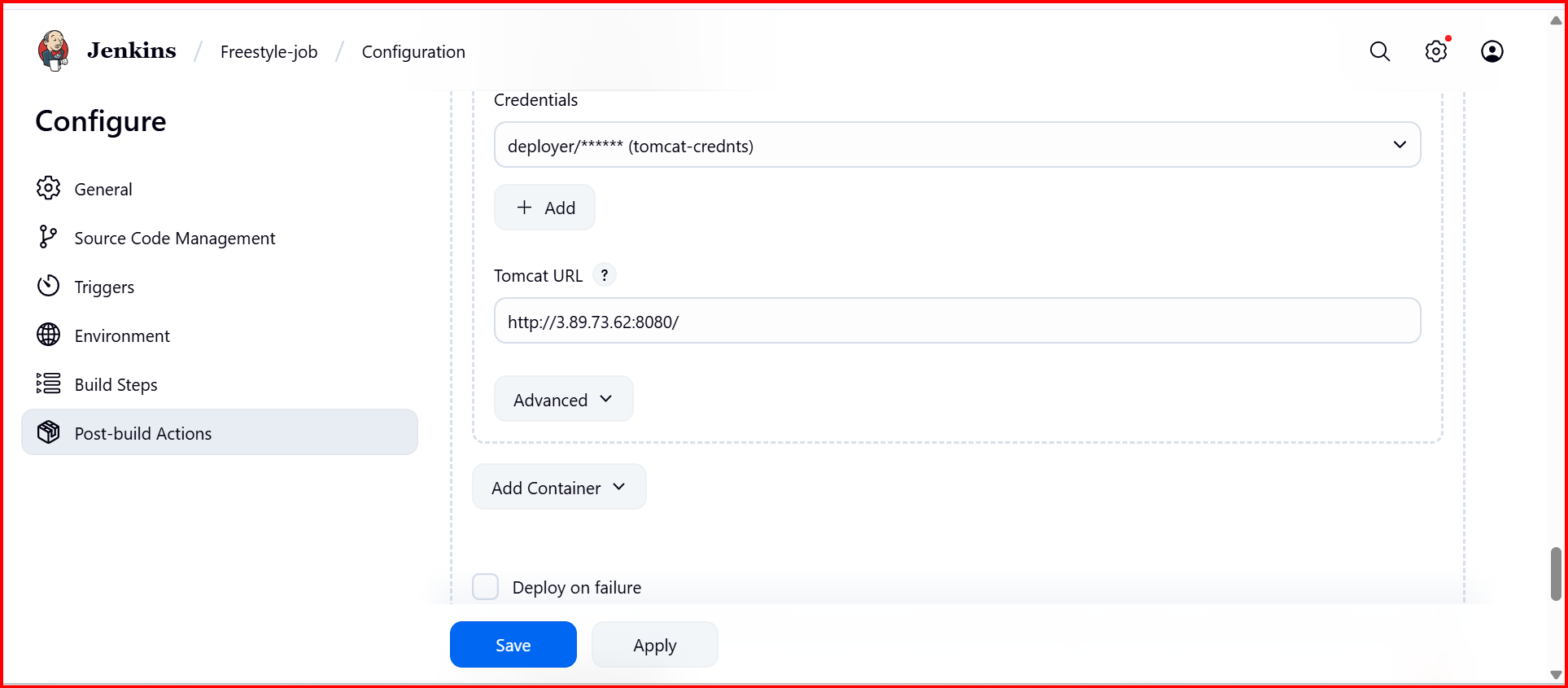
**Password: deployer**

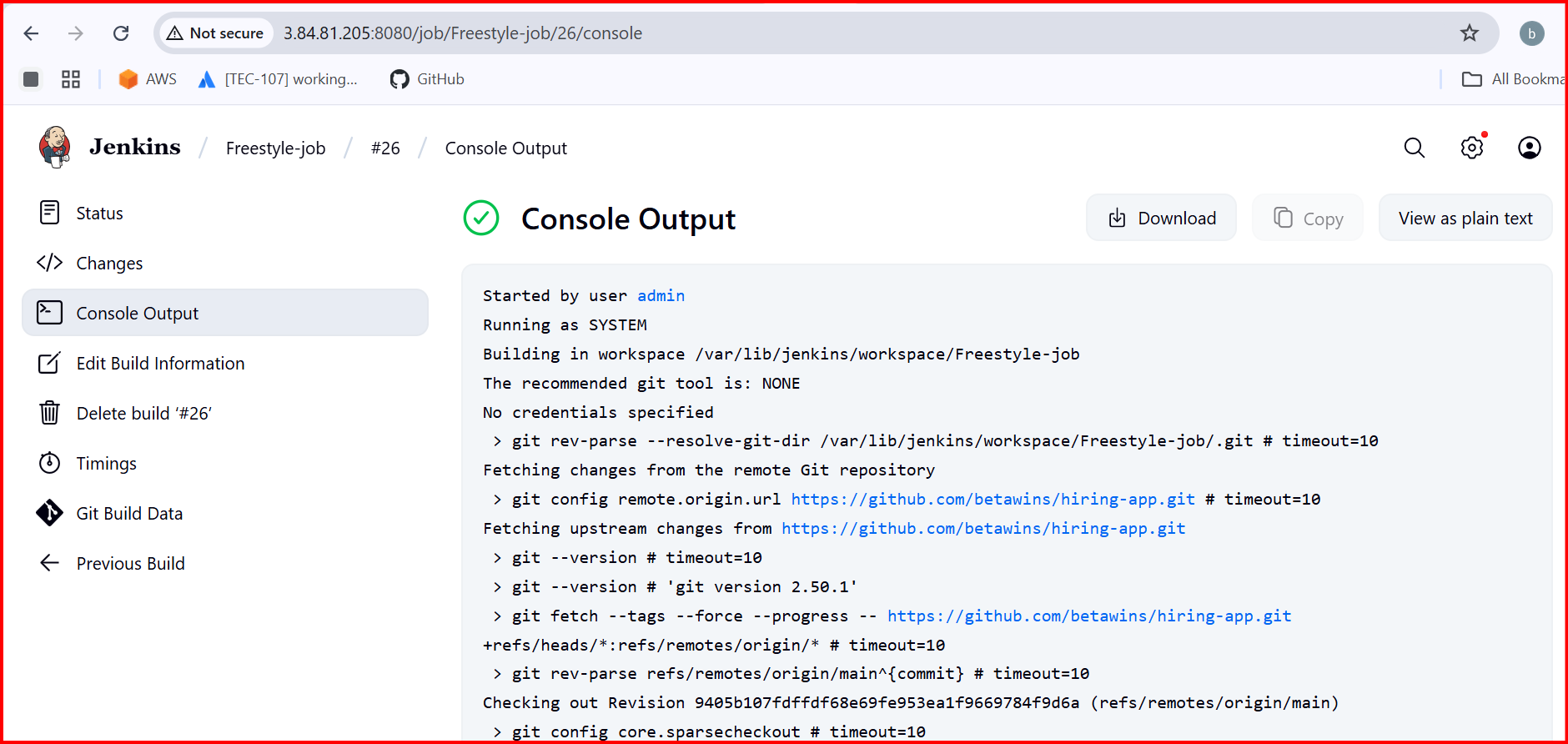
****

**Install deploy to container plugin**

****

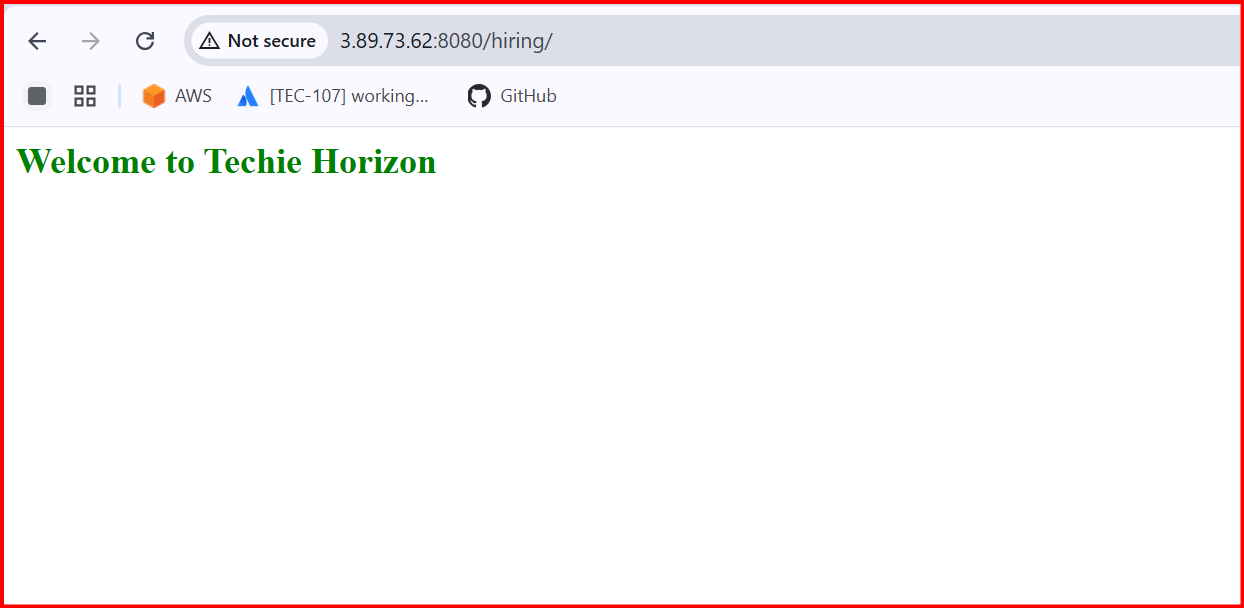
****

****

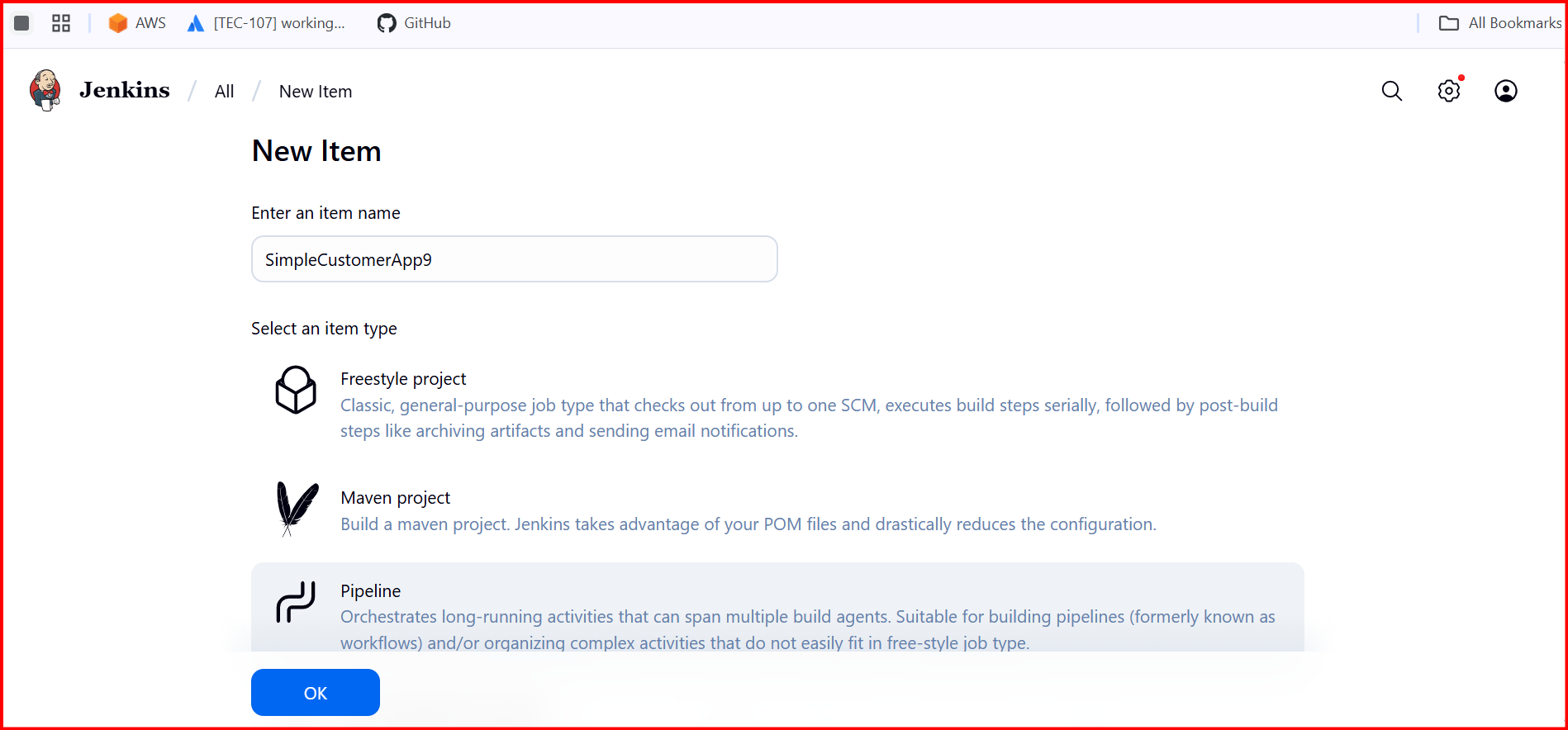
****

****

[**http://3.89.73.62:8080/hiring/**](http://3.89.73.62:8080/hiring/)

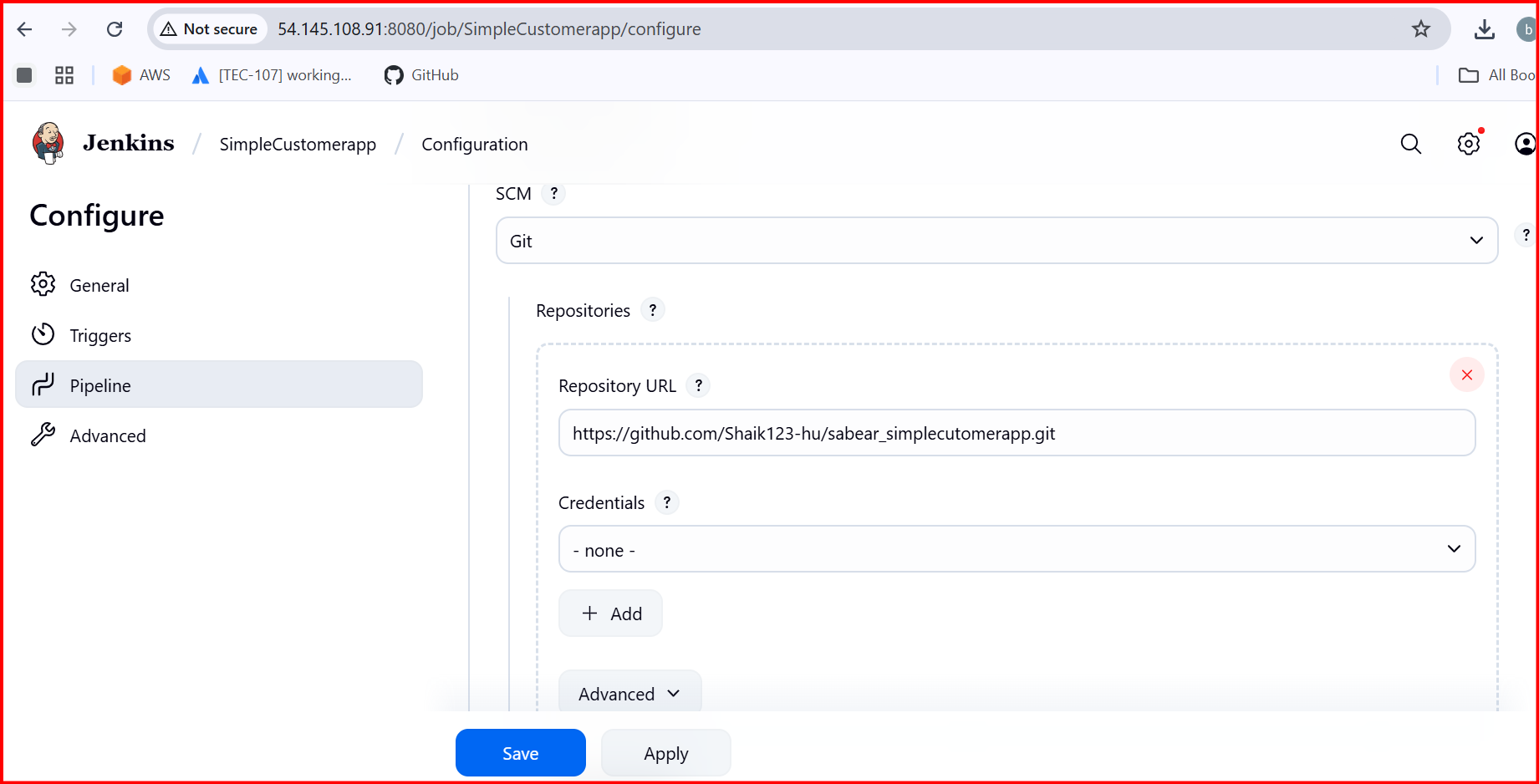
****

**2) Setup a jenkins CICD pipeline using Declarative pipeline using feature-1.1 branch.**[**https://github.com/betawins/sabear\_simplecutomerapp/tree/feature-1.1**](https://github.com/betawins/sabear_simplecutomerapp/tree/feature-1.1) **stages:  
   1) Git Clone  
   2) Sonarqube Integration  
   3) Maven Compilation  
   4) Nexus Artifactory  
   5) Slack Notification  
   6) Deploy On tomcat**

****

**Select pipeline**

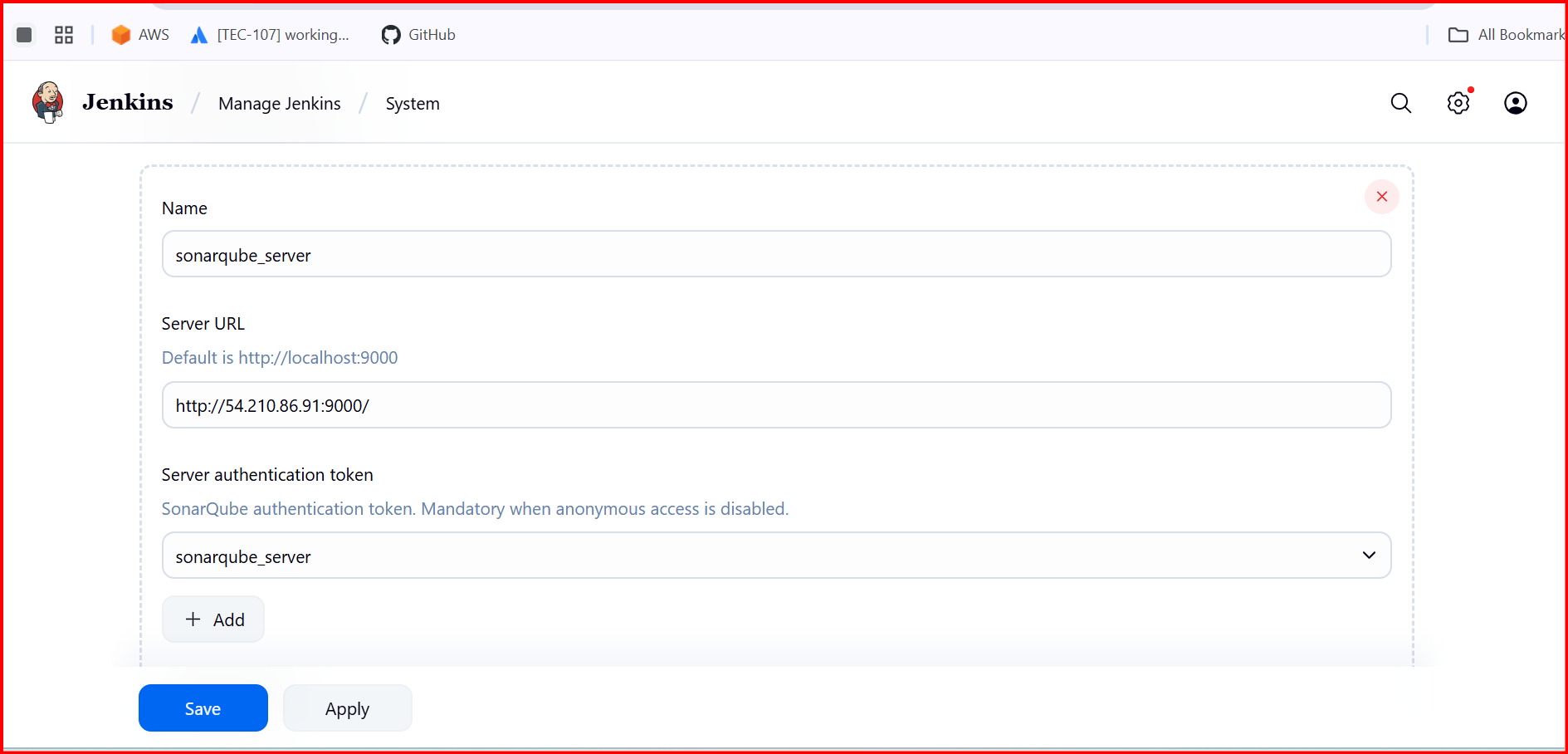
[**https://github.com/Shaik123-hu/sabear\_simplecutomerapp.git**](https://github.com/Shaik123-hu/sabear_simplecutomerapp.git)

****

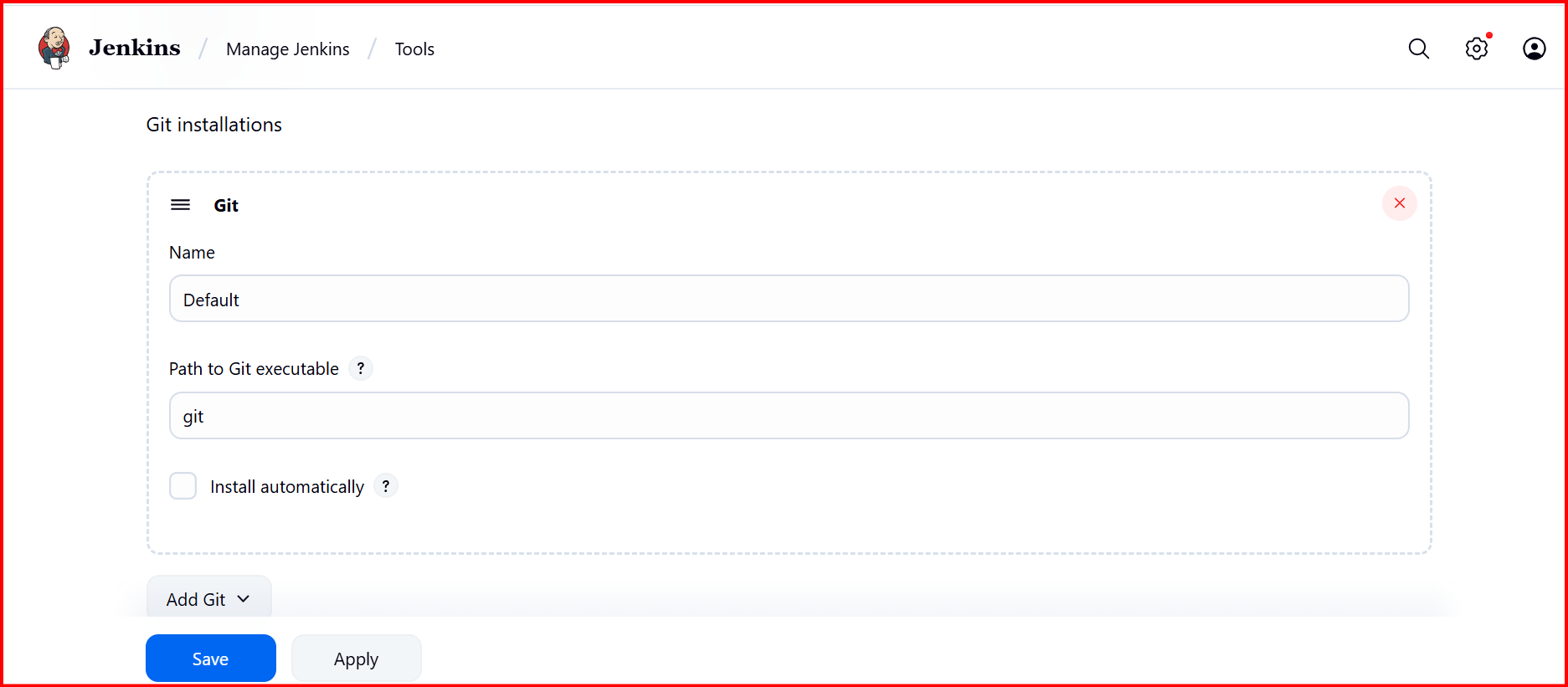
**Branch Specifier (blank for 'any')**

**\*/feature-1.1**

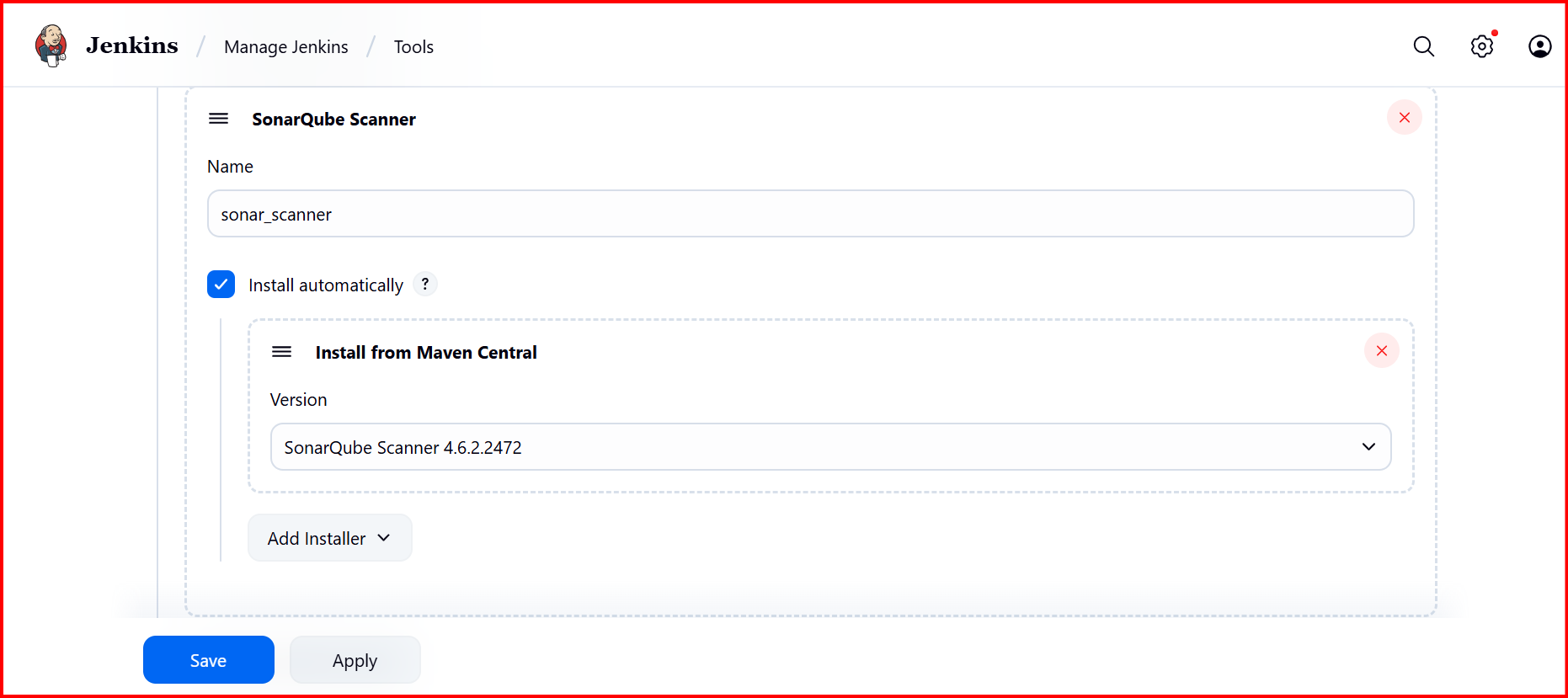
**Add sonarqube server to Jenkins**

****

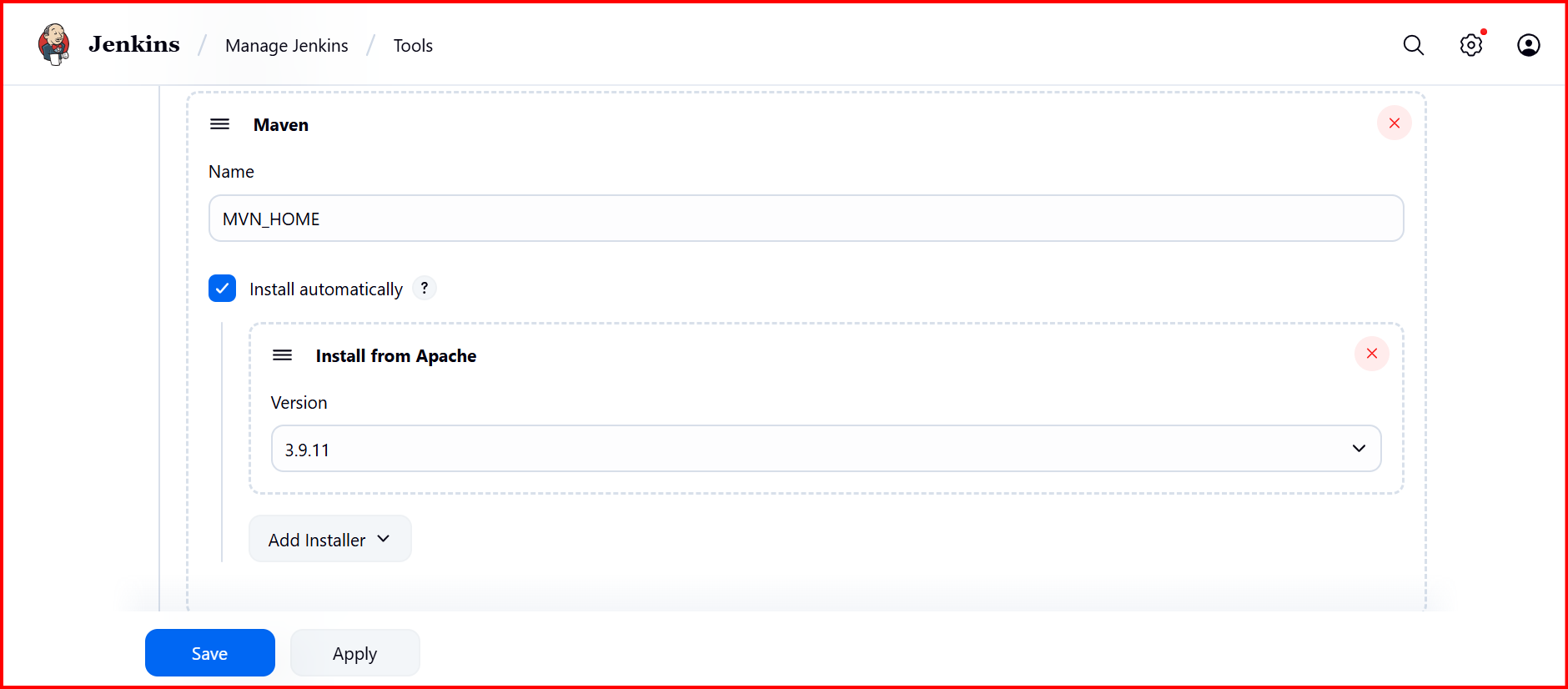
**Git**

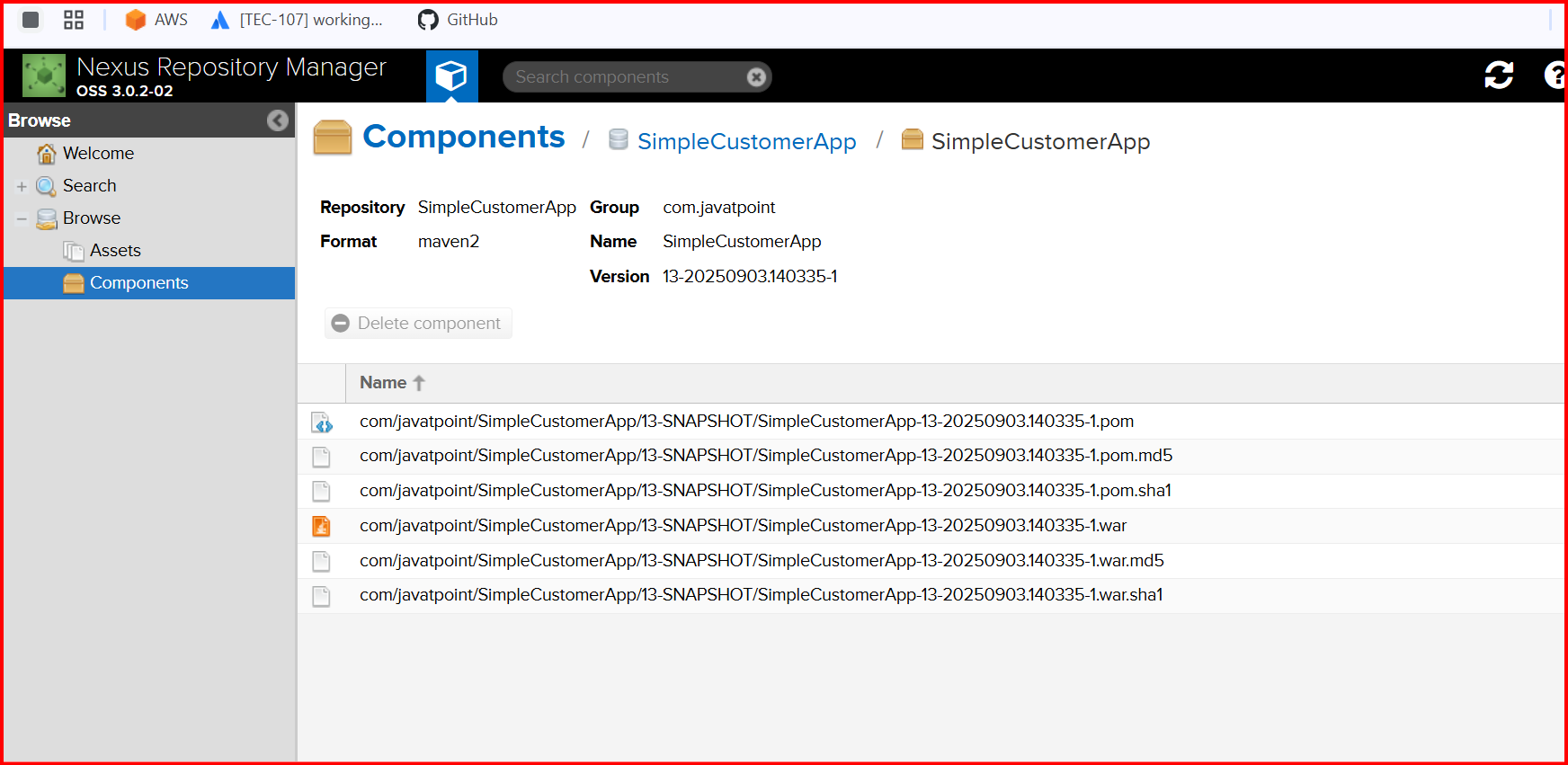
****

**Sonar scanner**

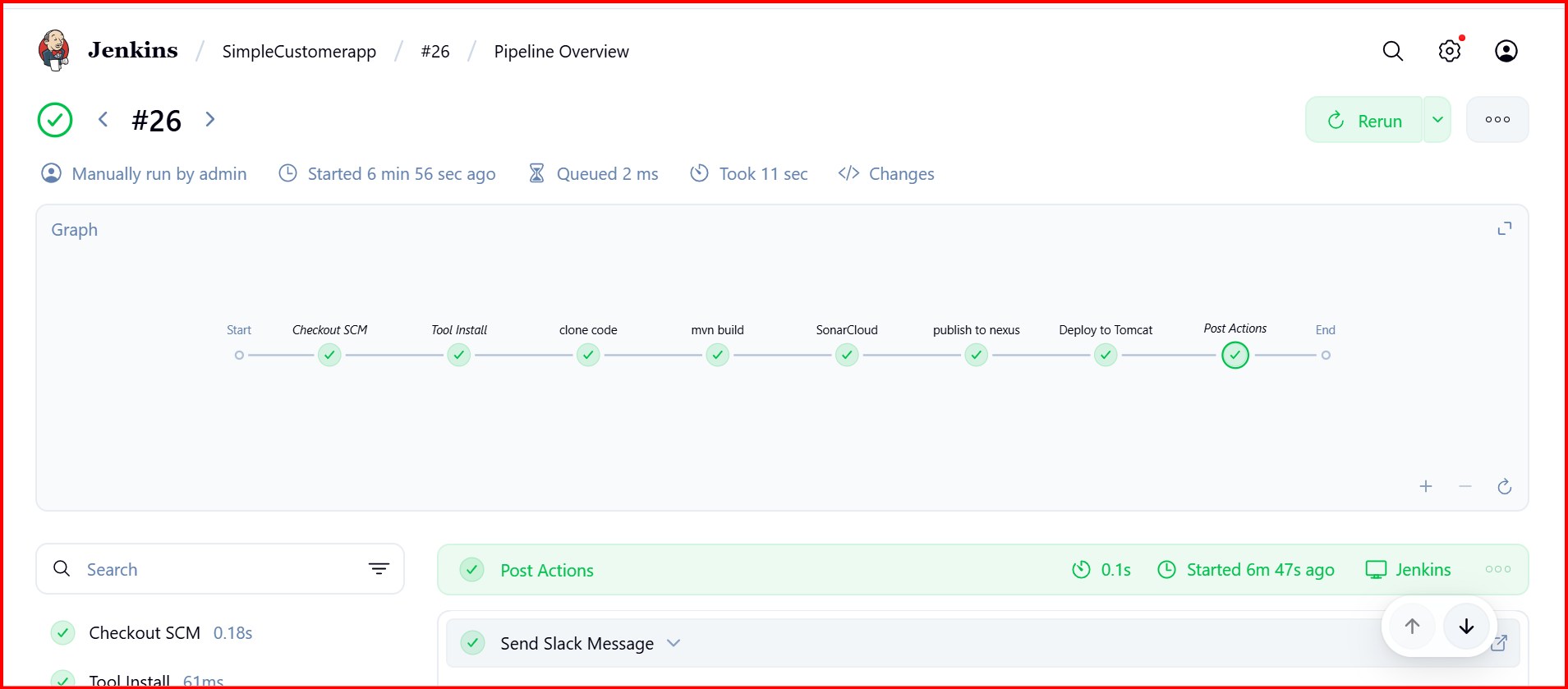
****

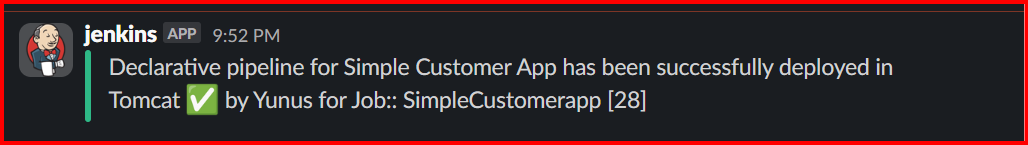
**Maven**

****

****

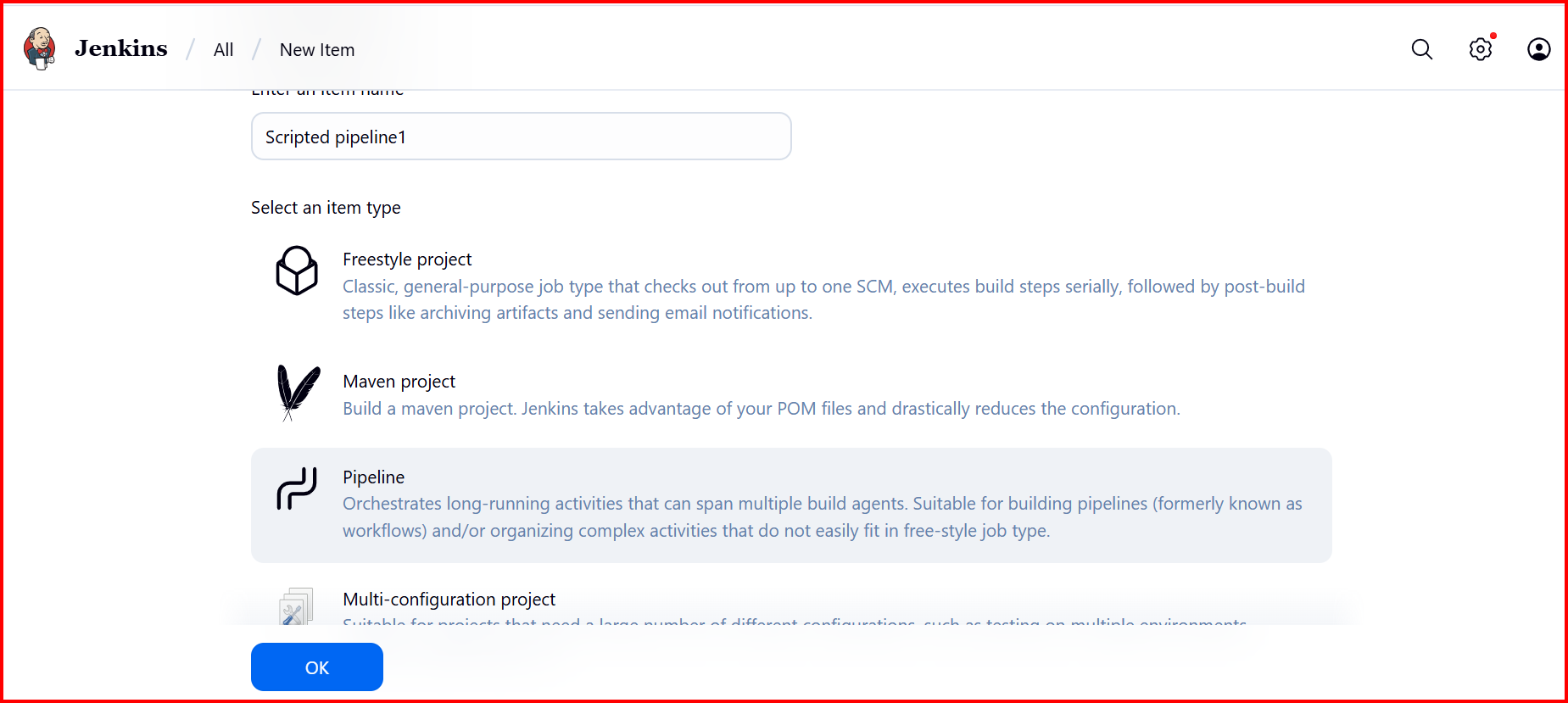
****

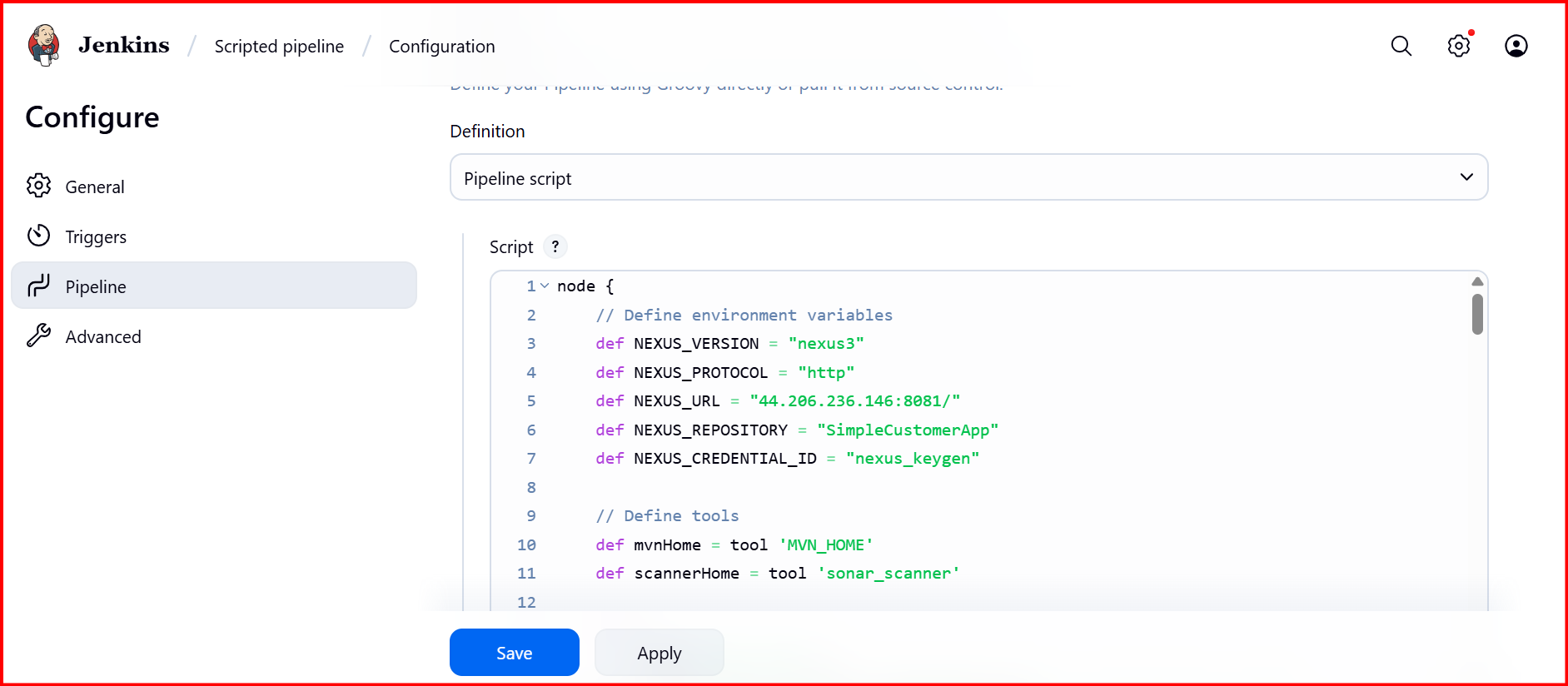
****

****

****

**3) Setup a jenkins CICD pipeline using Scripted pipeline using feature-1.1 branch.**[**https://github.com/betawins/sabear\_simplecutomerapp/tree/feature-1.1**](https://github.com/betawins/sabear_simplecutomerapp/tree/feature-1.1) **stages:  
   1) Git Clone  
   2) Sonarqube Integration  
   3) Maven Compilation  
   4) Nexus Artifactory  
   5) Slack Notification  
   6) Deploy On tomcat**

****

****

**node {**

**// Define environment variables**

**def NEXUS\_VERSION = "nexus3"**

**def NEXUS\_PROTOCOL = "http"**

**def NEXUS\_URL = "44.206.236.146:8081/"**

**def NEXUS\_REPOSITORY = "SimpleCustomerApp"**

**def NEXUS\_CREDENTIAL\_ID = "nexus\_keygen"**

**// Define tools**

**def mvnHome = tool 'MVN\_HOME'**

**def scannerHome = tool 'sonar\_scanner'**

**try {**

**stage('Clone Code') {**

**git 'https://github.com/Shaik123-hu/sabear\_simplecutomerapp.git'**

**}**

**stage('Maven Build') {**

**sh "${mvnHome}/bin/mvn -Dmaven.test.failure.ignore=true clean install"**

**}**

**stage('SonarCloud') {**

**withSonarQubeEnv('sonarqube\_server') {**

**// Define directories**

**def sourceDir = "${WORKSPACE}/src/main/java"**

**def webappDir = "${WORKSPACE}/src/main/webapp"**

**def binaryDir = "${WORKSPACE}/target/classes"**

**def reportDir = "${WORKSPACE}/target/surefire-reports"**

**// Determine source directories to analyze**

**def sonarSources = []**

**if (fileExists(sourceDir)) {**

**sonarSources.add(sourceDir)**

**}**

**if (fileExists(webappDir)) {**

**sonarSources.add(webappDir)**

**}**

**if (sonarSources.isEmpty()) {**

**echo "Warning: No source directories (src/main/java or src/main/webapp) found. Skipping SonarQube analysis."**

**return**

**}**

**// Build SonarQube command**

**def sonarCommand = """**

**${scannerHome}/bin/sonar-scanner \**

**-Dsonar.projectKey=Ncodeit \**

**-Dsonar.projectName=Ncodeit \**

**-Dsonar.projectVersion=2.0 \**

**-Dsonar.sources="${sonarSources.join(',')}" \**

**-Dsonar.junit.reportsPath="${reportDir}" \**

**-Dsonar.verbose=true**

**"""**

**// Add binaries only if target/classes exists**

**if (fileExists(binaryDir)) {**

**sonarCommand += " -Dsonar.java.binaries=\"${binaryDir}\""**

**} else {**

**echo "Warning: Binary directory ${binaryDir} does not exist. Skipping Java binaries analysis."**

**}**

**// Execute SonarQube analysis**

**sh sonarCommand**

**}**

**}**

**stage('Publish to Nexus') {**

**def pom = readMavenPom file: 'pom.xml'**

**def groupId = pom.groupId**

**def artifactId = pom.artifactId**

**// Use a fallback version if ${BUILD\_NUMBER} causes issues**

**def version = pom.version.replace('${BUILD\_NUMBER}', env.BUILD\_NUMBER ?: '3')**

**def packaging = pom.packaging**

**def filesByGlob = findFiles(glob: "target/\*.${packaging}")**

**if (filesByGlob.length == 0) {**

**error "\*\*\* No artifact found in target/\*.${packaging}"**

**}**

**def artifactPath = filesByGlob[0].path**

**if (fileExists(artifactPath)) {**

**echo "\*\*\* Uploading ${artifactPath} to Nexus (group: ${groupId}, version: ${version}, packaging: ${packaging})"**

**nexusArtifactUploader(**

**nexusVersion: NEXUS\_VERSION,**

**protocol: NEXUS\_PROTOCOL,**

**nexusUrl: NEXUS\_URL,**

**groupId: groupId,**

**version: version,**

**repository: NEXUS\_REPOSITORY,**

**credentialsId: NEXUS\_CREDENTIAL\_ID,**

**artifacts: [**

**[artifactId: artifactId, classifier: '', file: artifactPath, type: packaging],**

**[artifactId: artifactId, classifier: '', file: 'pom.xml', type: 'pom']**

**]**

**)**

**} else {**

**error "\*\*\* File: ${artifactPath}, could not be found"**

**}**

**}**

**stage('Deploy to Tomcat') {**

**withCredentials([usernamePassword(credentialsId: 'tomcat', usernameVariable: 'TOMCAT\_USER', passwordVariable: 'TOMCAT\_PASS')]) {**

**def warFile = sh(script: "ls target/\*.war | head -n 1", returnStdout: true).trim()**

**echo "Deploying ${warFile} to Tomcat at context path /SimpleCustomerApp..."**

**sh """**

**curl -u ${TOMCAT\_USER}:${TOMCAT\_PASS} \**

**-T ${warFile} \**

**"http://52.23.219.234:8080/manager/text/deploy?path=/SimpleCustomerApp&update=true"**

**"""**

**}**

**}**

**// Success notification**

**slackSend(**

**channel: '#jenkins-integration',**

**color: 'good',**

**message: "Scripted pipeline for Simple Customer App has been successfully deployed in Tomcat :white\_check\_mark: by Yunus for Job{env.JOB\_NAME} [${env.BUILD\_NUMBER}]",**

**tokenCredentialId: 'Slackid'**

**)**

**} catch (e) {**

**// Failure notification**

**slackSend(**

**channel: '#jenkins-integration',**

**color: 'danger',**

**message: "❌ Build/Deploy Failed: ${env.JOB\_NAME} [${env.BUILD\_NUMBER}]",**

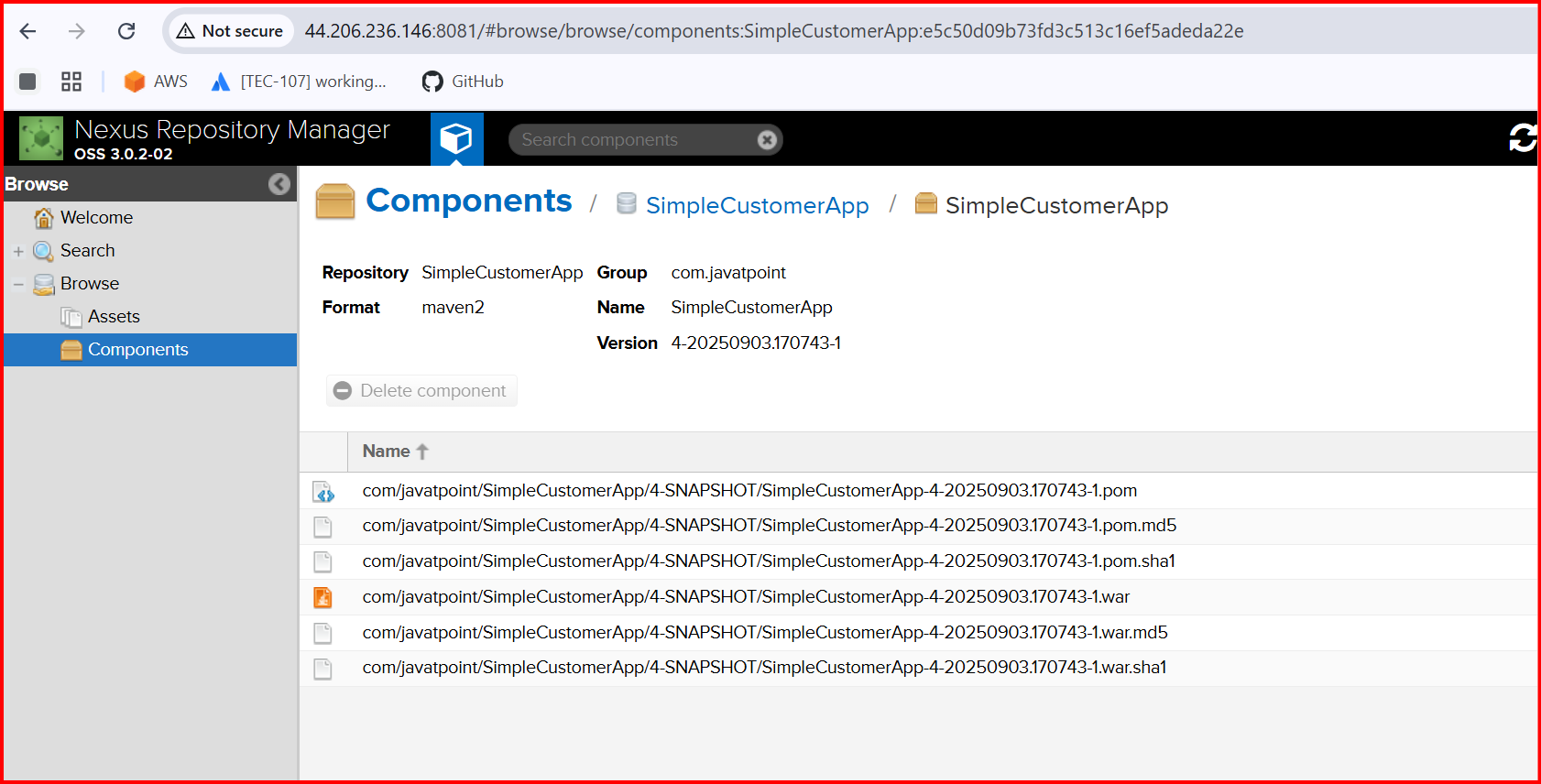
**tokenCredentialId: 'Slackid'**

**)**

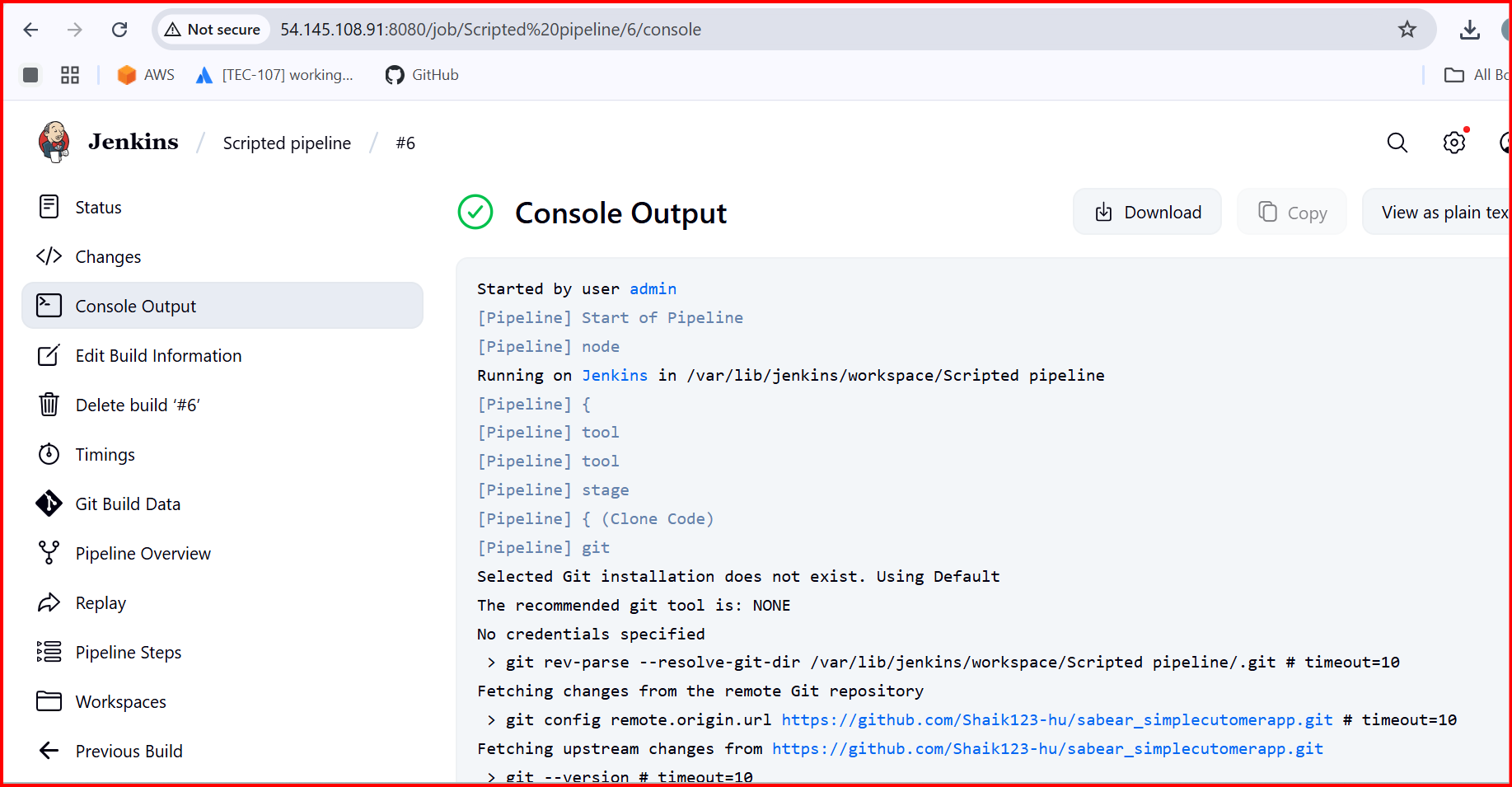
**throw e**

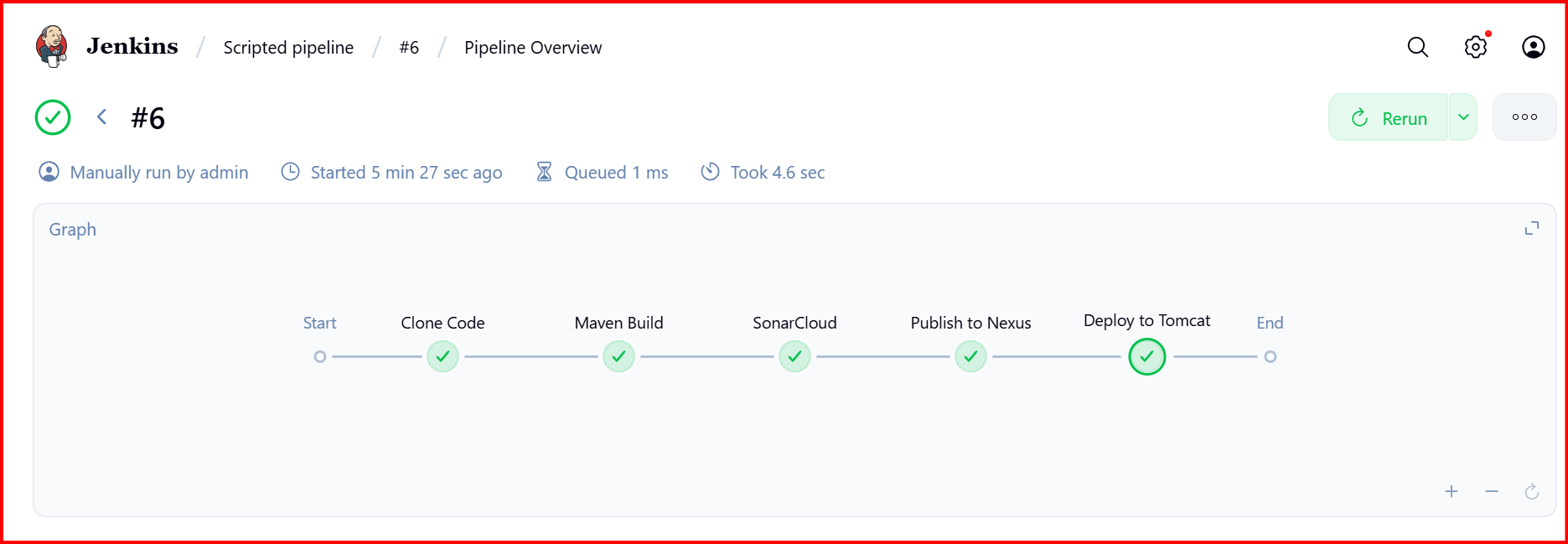
**}**

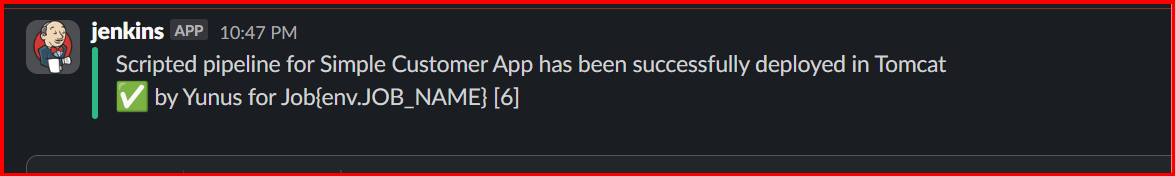
**}**

****

**Other configuration as above**

****

****

****

**4) Write sample skeleton of pipelines.**

**Scripted Pipeline Skeleton**

**node {**

**// Define tools and env if needed**

**def mvnHome = tool name: 'MVN\_HOME', type: 'maven'**

**stage('Checkout') {**

**git branch: 'feature-1.1',**

**url: 'https://github.com/your-repo/your-project.git'**

**}**

**stage('Build') {**

**sh "${mvnHome}/bin/mvn clean install -DskipTests=true"**

**}**

**stage('SonarQube Analysis') {**

**withSonarQubeEnv('sonarqube\_server') {**

**sh "${mvnHome}/bin/mvn sonar:sonar"**

**}**

**}**

**stage('Publish to Nexus') {**

**sh "${mvnHome}/bin/mvn deploy"**

**}**

**stage('Deploy to Tomcat') {**

**sh "cp target/\*.war /opt/tomcat/webapps/"**

**}**

**stage('Notify Slack') {**

**slackSend channel: '#jenkins-integration',**

**color: 'good',**

**message: "✅ Build & Deploy successful!"**

**}**

**}**

**Declarative Pipeline Skeleton**

**pipeline {**

**agent any**

**tools {**

**maven "MVN\_HOME"**

**}**

**environment {**

**BRANCH\_NAME = "feature-1.1"**

**SONARQUBE\_ENV = "sonarqube\_server"**

**}**

**stages {**

**stage('Checkout') {**

**steps {**

**git branch: "${BRANCH\_NAME}",**

**url: 'https://github.com/your-repo/your-project.git'**

**}**

**}**

**stage('Build') {**

**steps {**

**sh "mvn clean install -DskipTests=true"**

**}**

**}**

**stage('SonarQube Analysis') {**

**steps {**

**withSonarQubeEnv("${SONARQUBE\_ENV}") {**

**sh "mvn sonar:sonar"**

**}**

**}**

**}**

**stage('Publish to Nexus') {**

**steps {**

**sh "mvn deploy"**

**}**

**}**

**stage('Deploy to Tomcat') {**

**steps {**

**sh "cp target/\*.war /opt/tomcat/webapps/"**

**}**

**}**

**stage('Notify Slack') {**

**steps {**

**slackSend(channel: '#jenkins-integration',**

**color: 'good',**

**message: "✅ Build & Deploy successful!")**

**}**

**}**

**}**

**}**