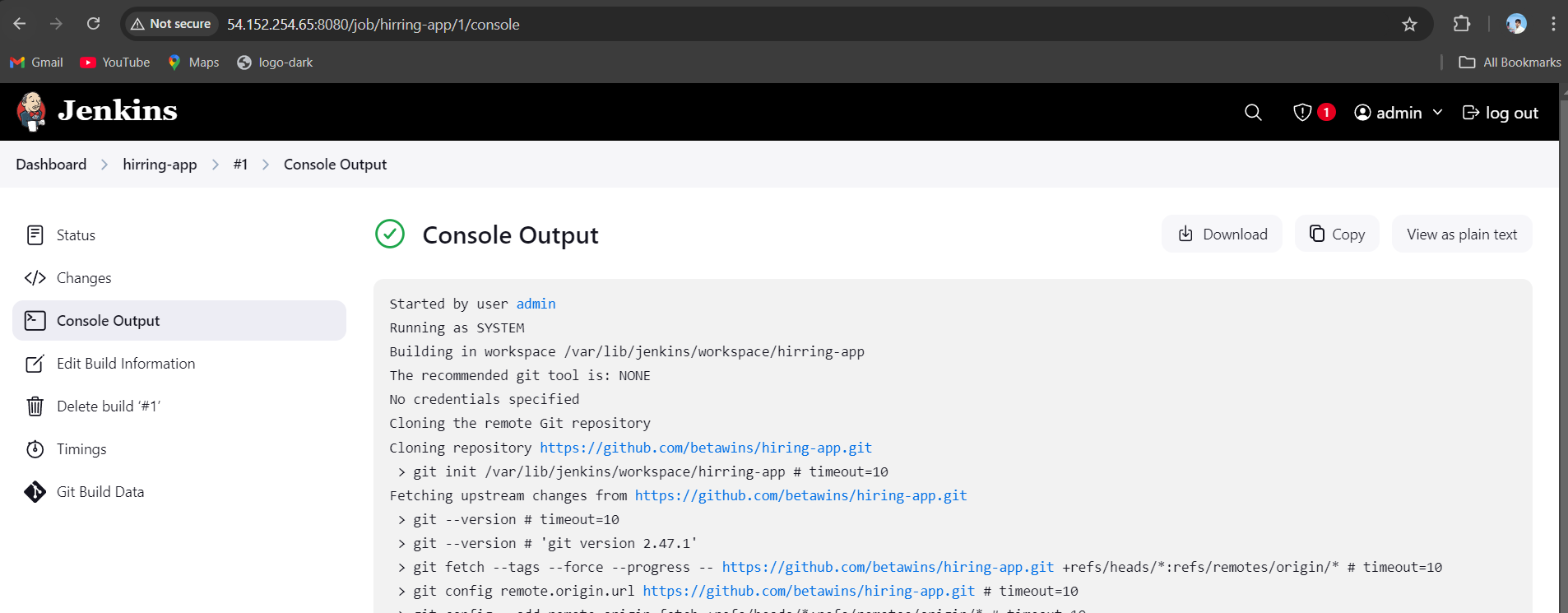
Jenkins-task04

1) Setup Jenkins CICD pipeline using freestyle jobs using Docker containers using below code.

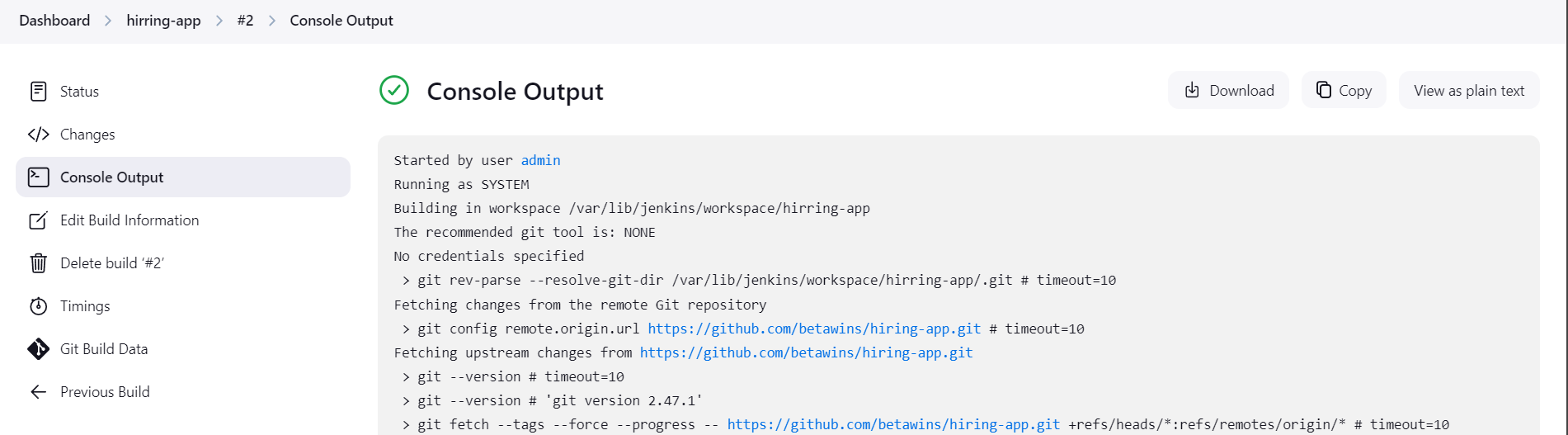
https://github.com/betawins/hiring-app.git

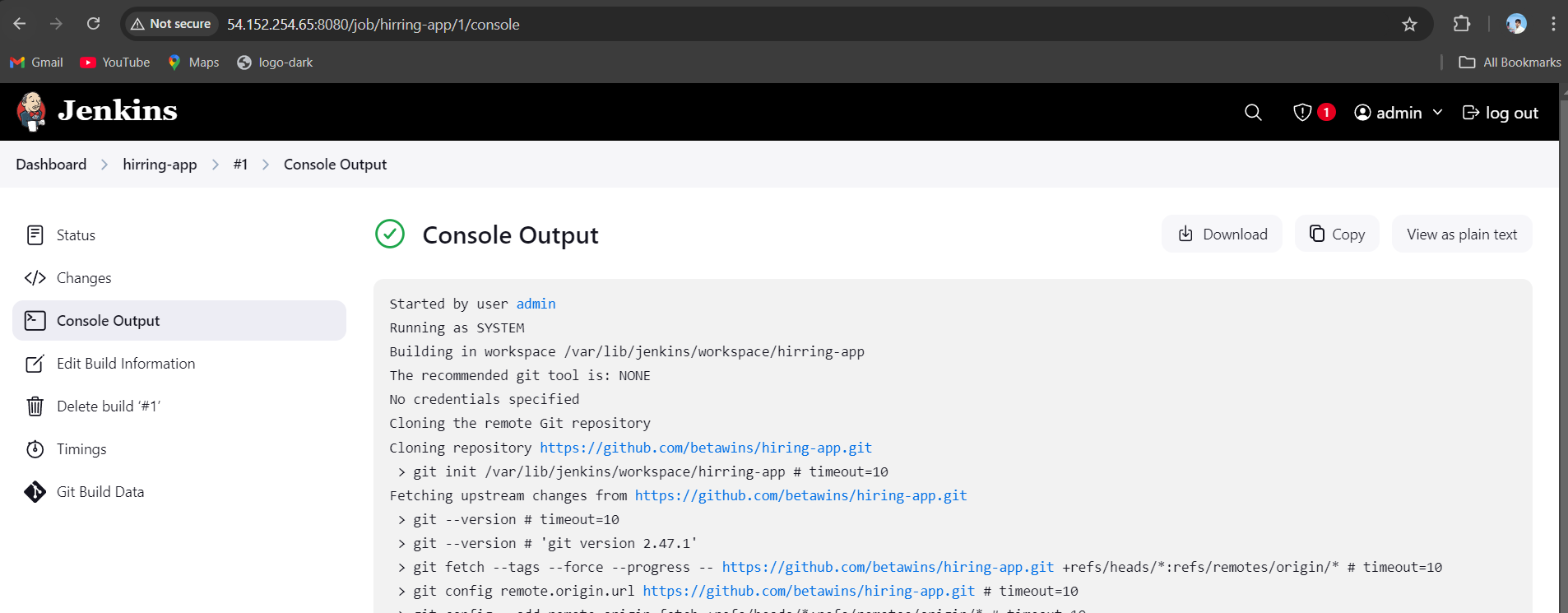
Stages:

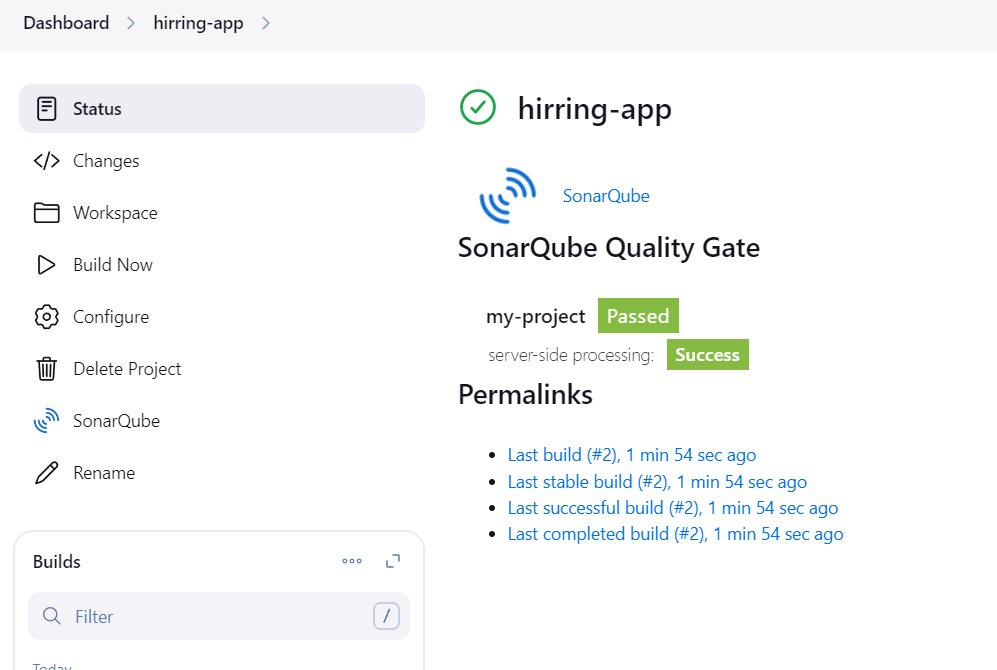
1) Git Clone

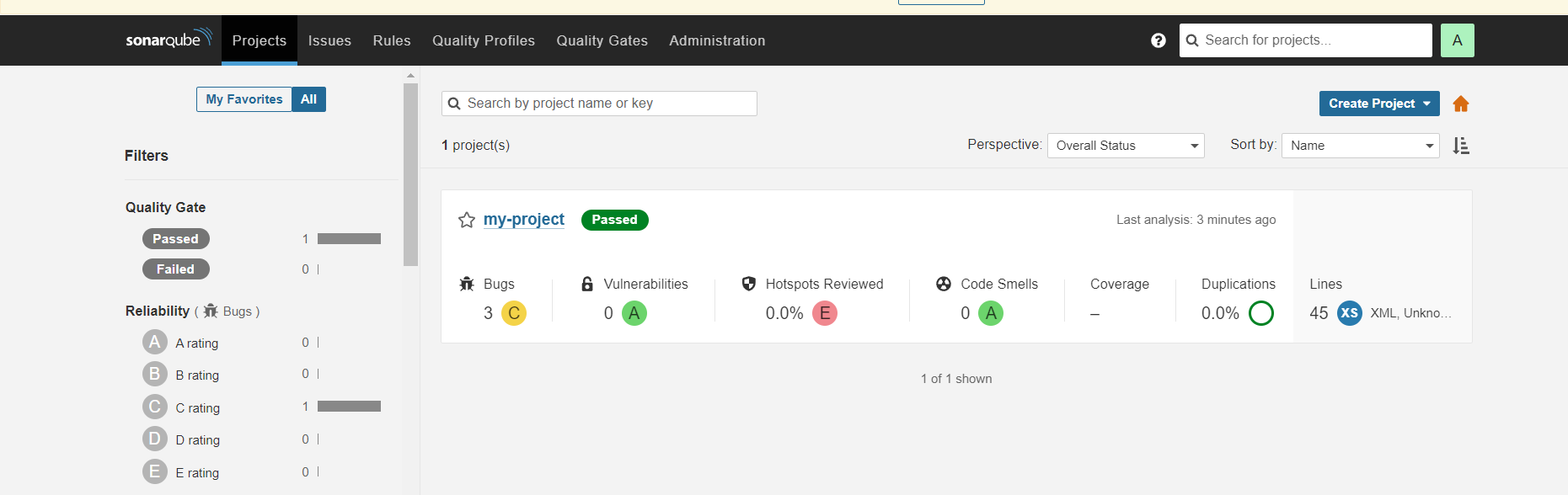


2) Sonarqube Integration

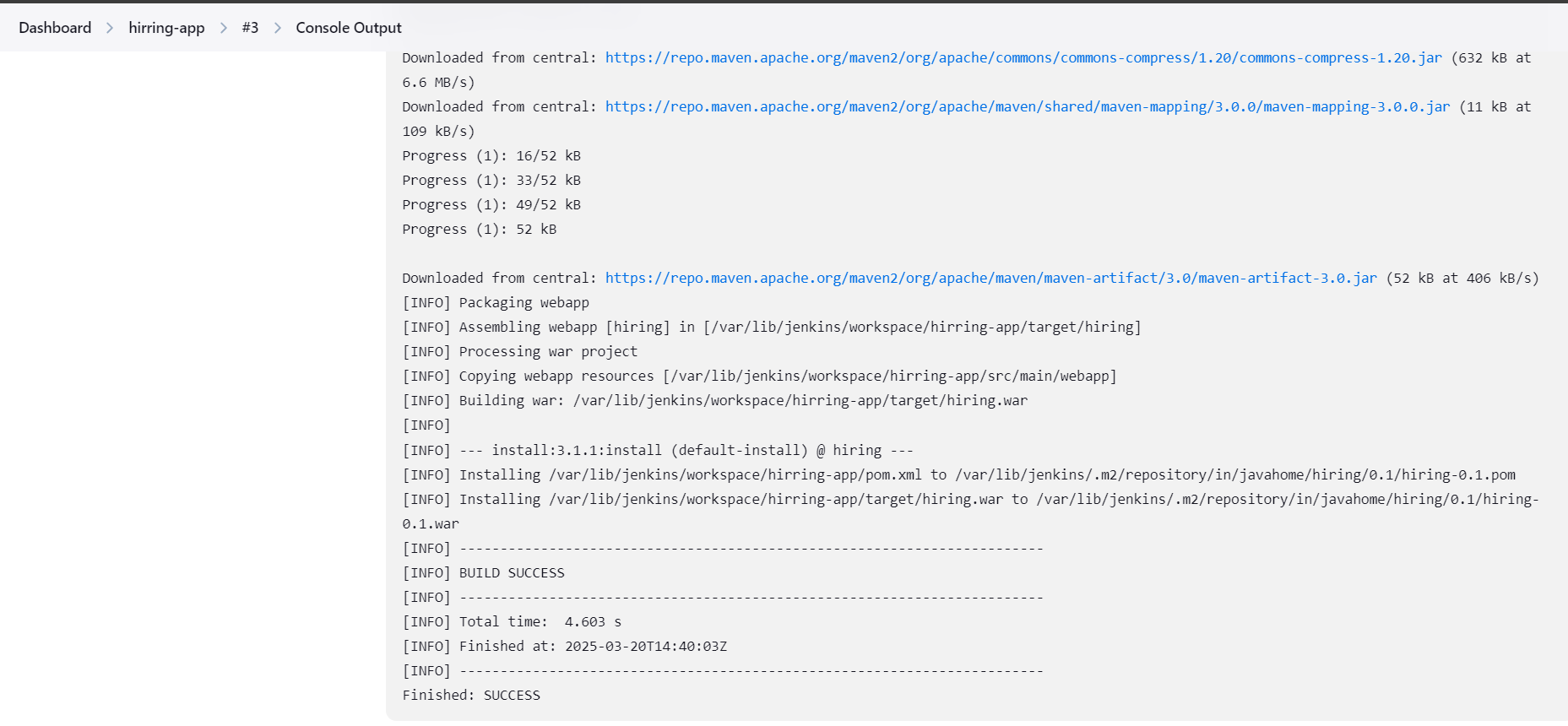


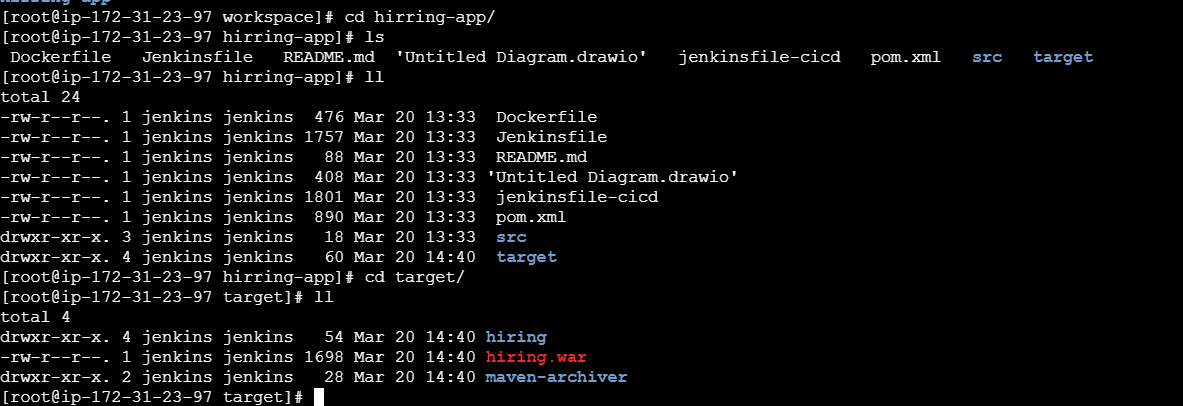






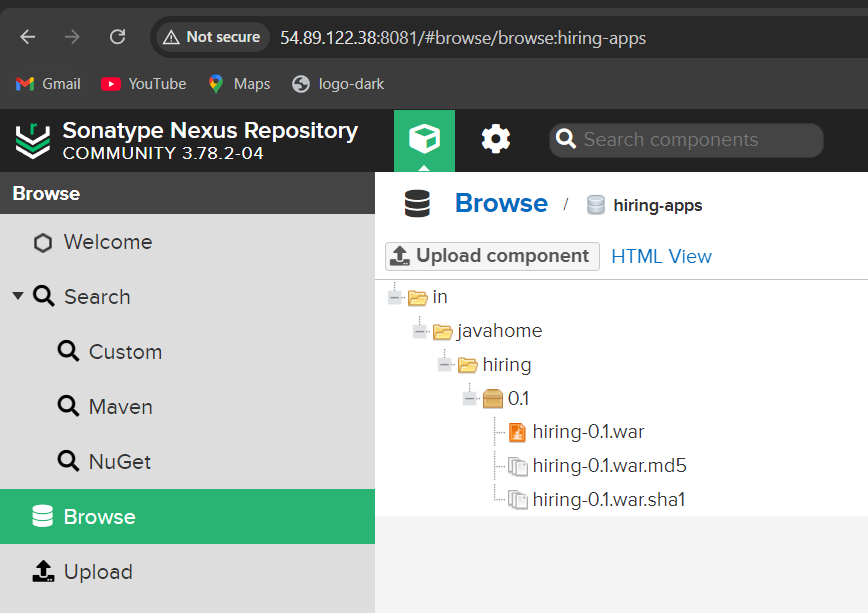
3) Maven Compilation



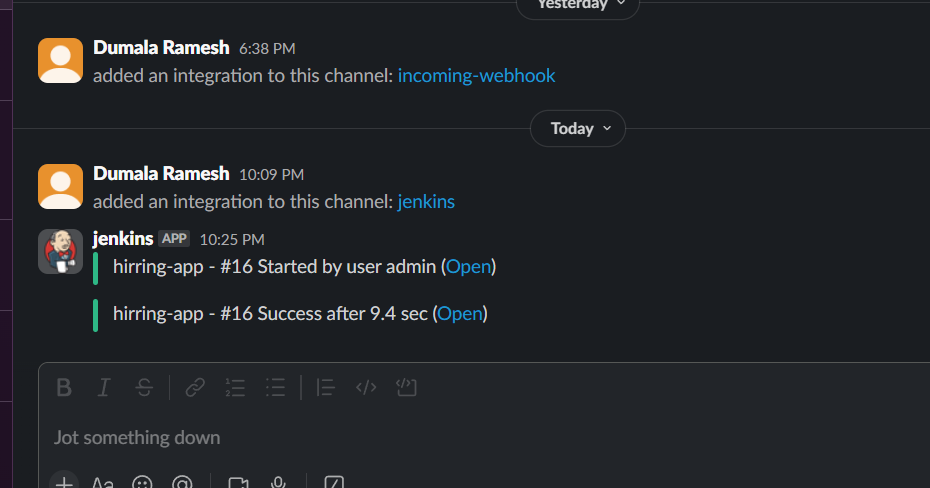


4) Nexus Artifactory

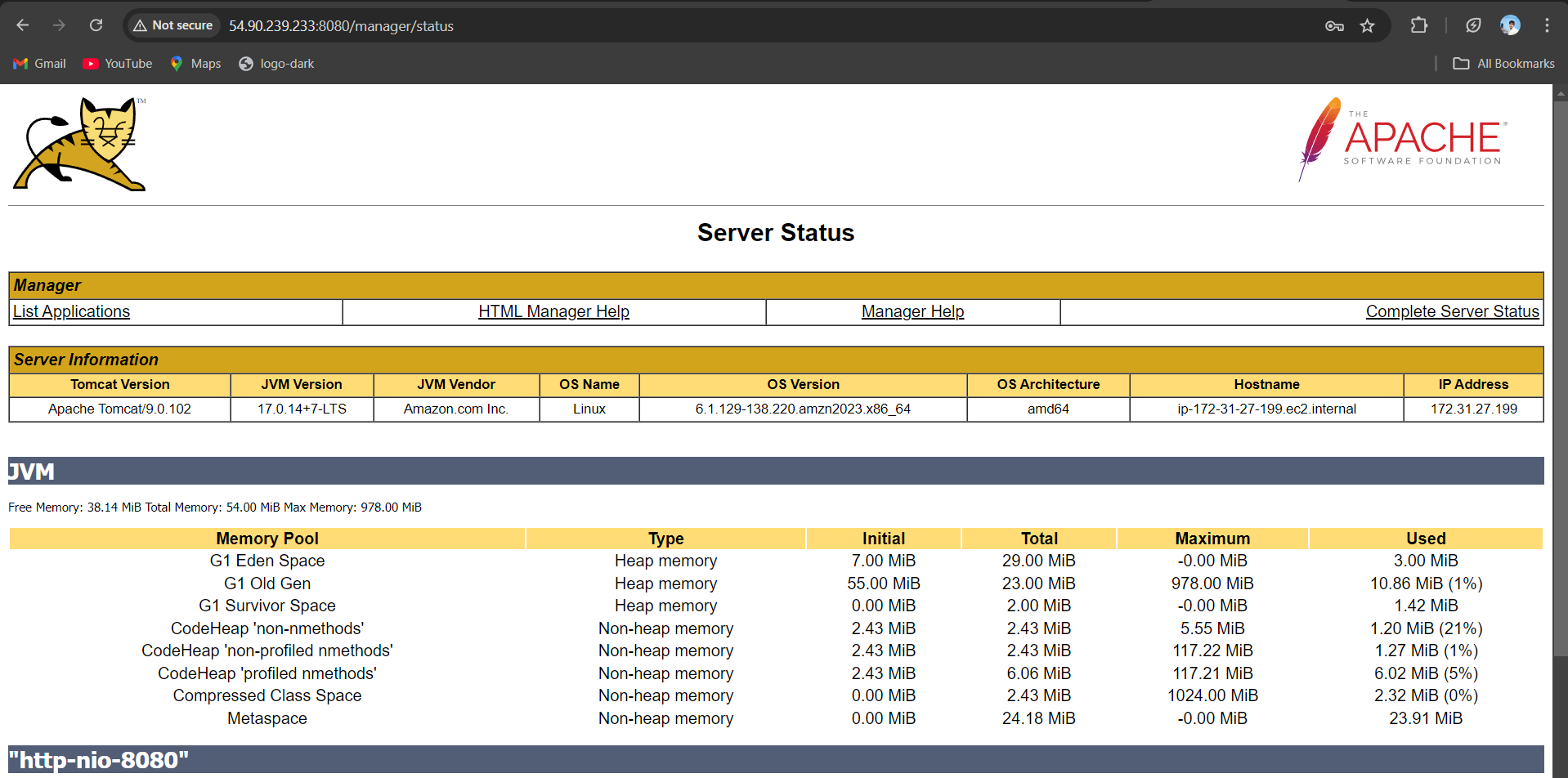


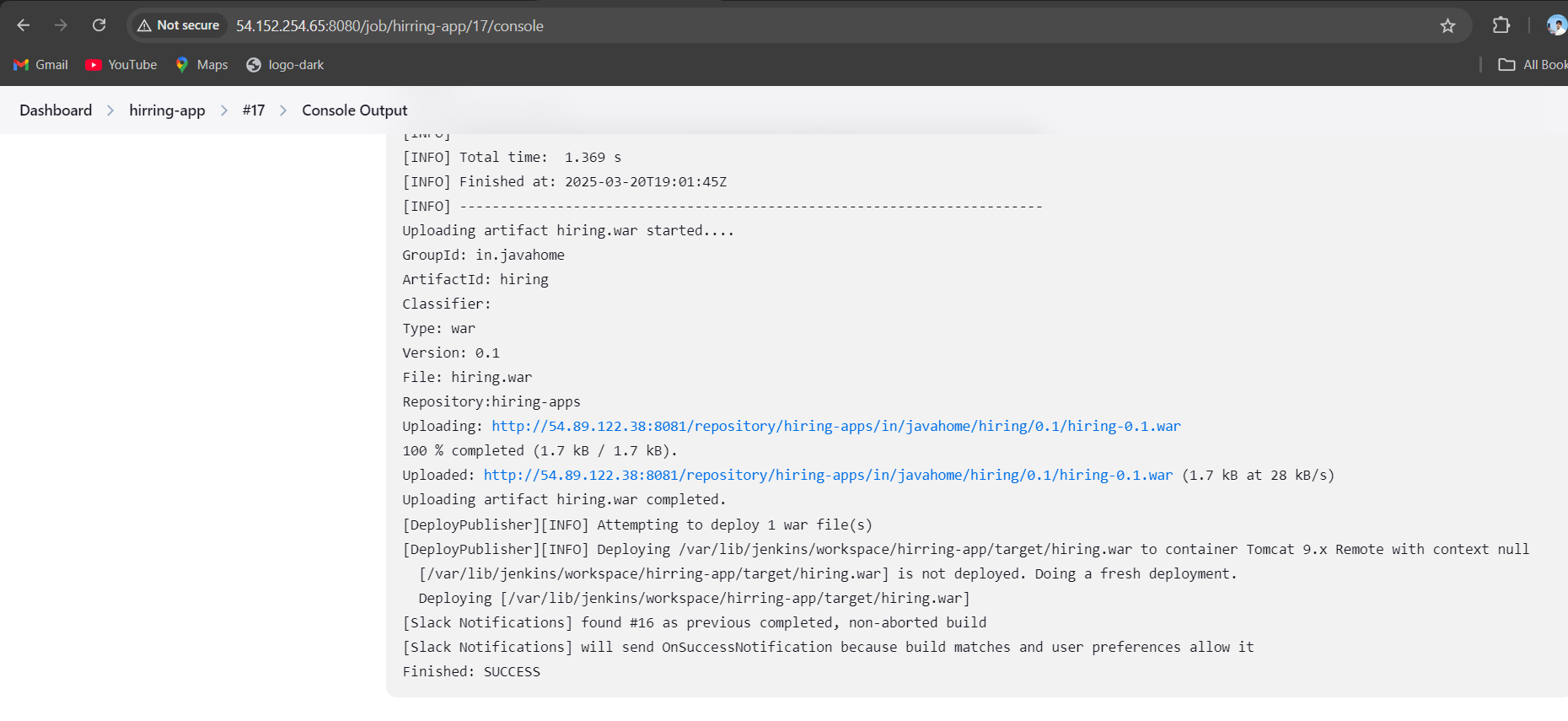


5) Slack Notification



6) Deploy On tomcat





2) Setup a Jenkins CICD pipeline using Declarative pipeline using feature-1.1 branch.

pipeline {

agent any

tools {

maven "MVN\_HOME"

}

environment {

NEXUS\_VERSION = "nexus3"

NEXUS\_PROTOCOL = "http"

NEXUS\_URL = "34.229.51.200:8081"

NEXUS\_REPOSITORY = "sonarqube"

NEXUS\_CREDENTIAL\_ID = "nexus"

SCANNER\_HOME = tool 'sonar\_scanner'

SLACK\_TOKEN = credentials('SLACK\_TOKEN')

}

stages {

stage("Clone Code") {

steps {

git 'https://github.com/betawins/sabear\_simplecutomerapp.git'

}

}

stage("Maven Build") {

steps {

sh 'mvn -Dmaven.test.failure.ignore=true clean install'

}

}

stage("SonarQube Analysis") {

steps {

withSonarQubeEnv('sonarqube\_server') {

sh '''

${SCANNER\_HOME}/bin/sonar-scanner \

-Dsonar.projectKey=Ncodeit \

-Dsonar.projectName=Ncodeit \

-Dsonar.projectVersion=2.0 \

-Dsonar.sources=src \

-Dsonar.java.binaries=target

'''

}

}

}

stage("Publish to Nexus") {

steps {

script {

nexusArtifactUploader(

artifacts: [[

artifactId: 'SimpleCustomerApp',

classifier: '',

file: "target/SimpleCustomerApp-${BUILD\_NUMBER}-SNAPSHOT.war",

type: 'war'

]],

credentialsId: 'nexus',

groupId: 'com.javatpoint',

nexusUrl: "${NEXUS\_URL}",

nexusVersion: "${NEXUS\_VERSION}",

protocol: "${NEXUS\_PROTOCOL}",

repository: "${NEXUS\_REPOSITORY}",

version: "${BUILD\_NUMBER}-SNAPSHOT"

)

}

}

}

stage("Deploy on Tomcat") {

steps {

script {

deploy adapters: [tomcat9(credentialsId: 'TOMCAT\_CREDENTIALS', path: '', url: 'http://3.86.14.58:8080/')],

contextPath: 'SimpleCustomerApp',

war: "target/SimpleCustomerApp-${BUILD\_NUMBER}-SNAPSHOT.war"

}

}

}

stage("Slack Notification") {

steps {

slackSend(

channel: "#jenkins-integration",

tokenCredentialId: "SLACK\_TOKEN",

message: "Build Completed: ${currentBuild.fullDisplayName}"

)

}

}

}

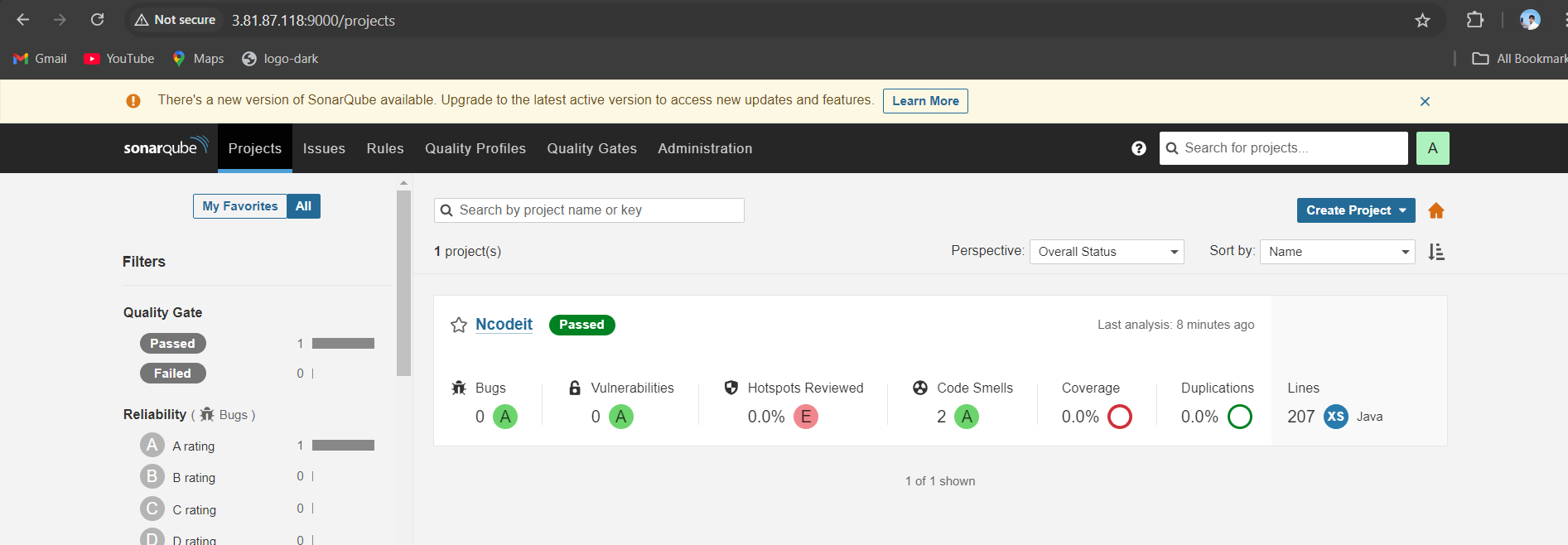
}

stages:

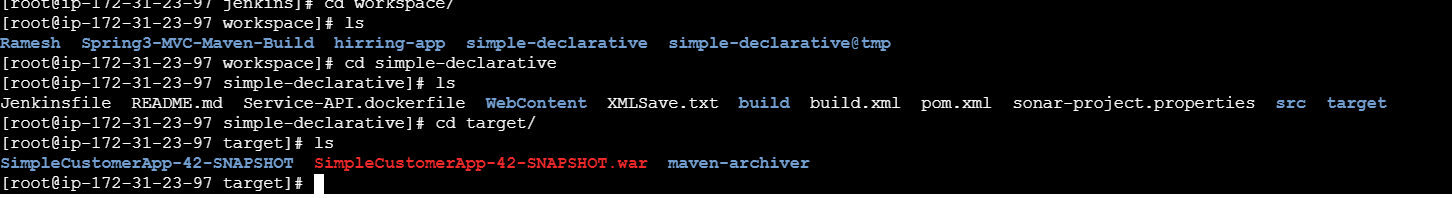
1) Git Clone



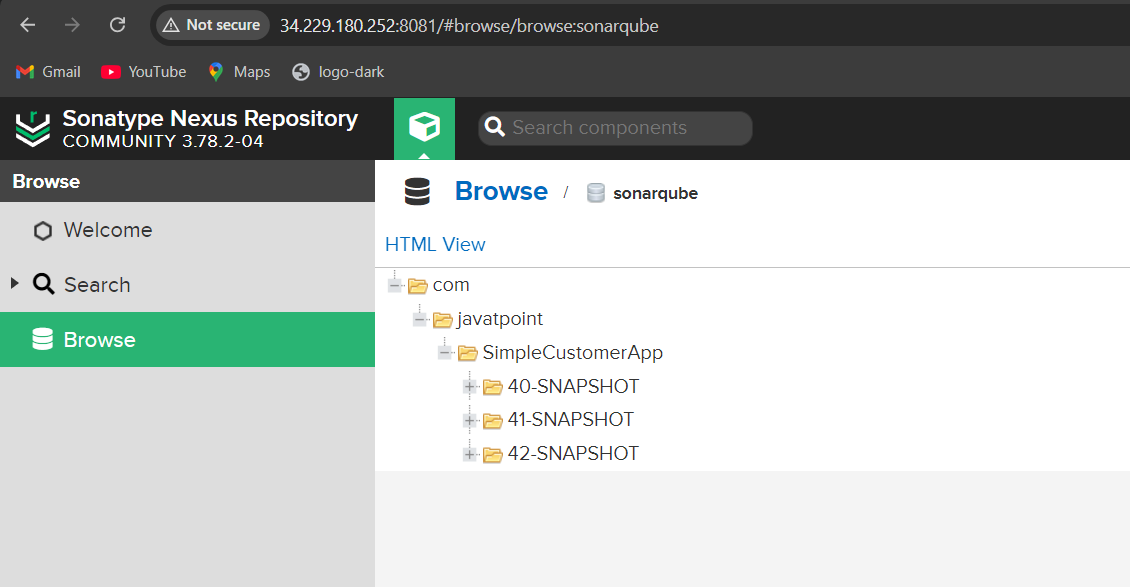
2) Sonarqube Integration



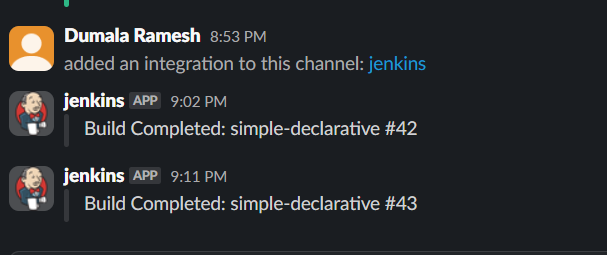
3) Maven Compilation



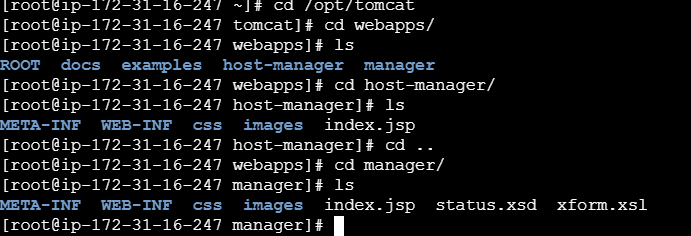
4) Nexus Artifactory



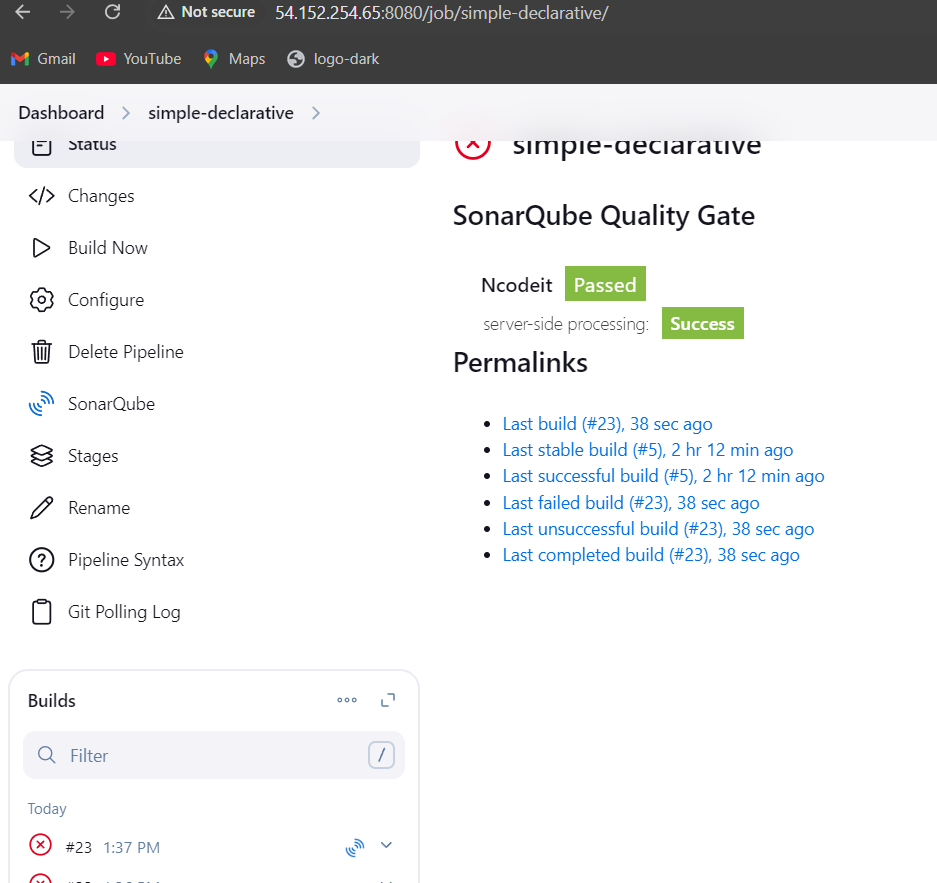
5) Slack Notification

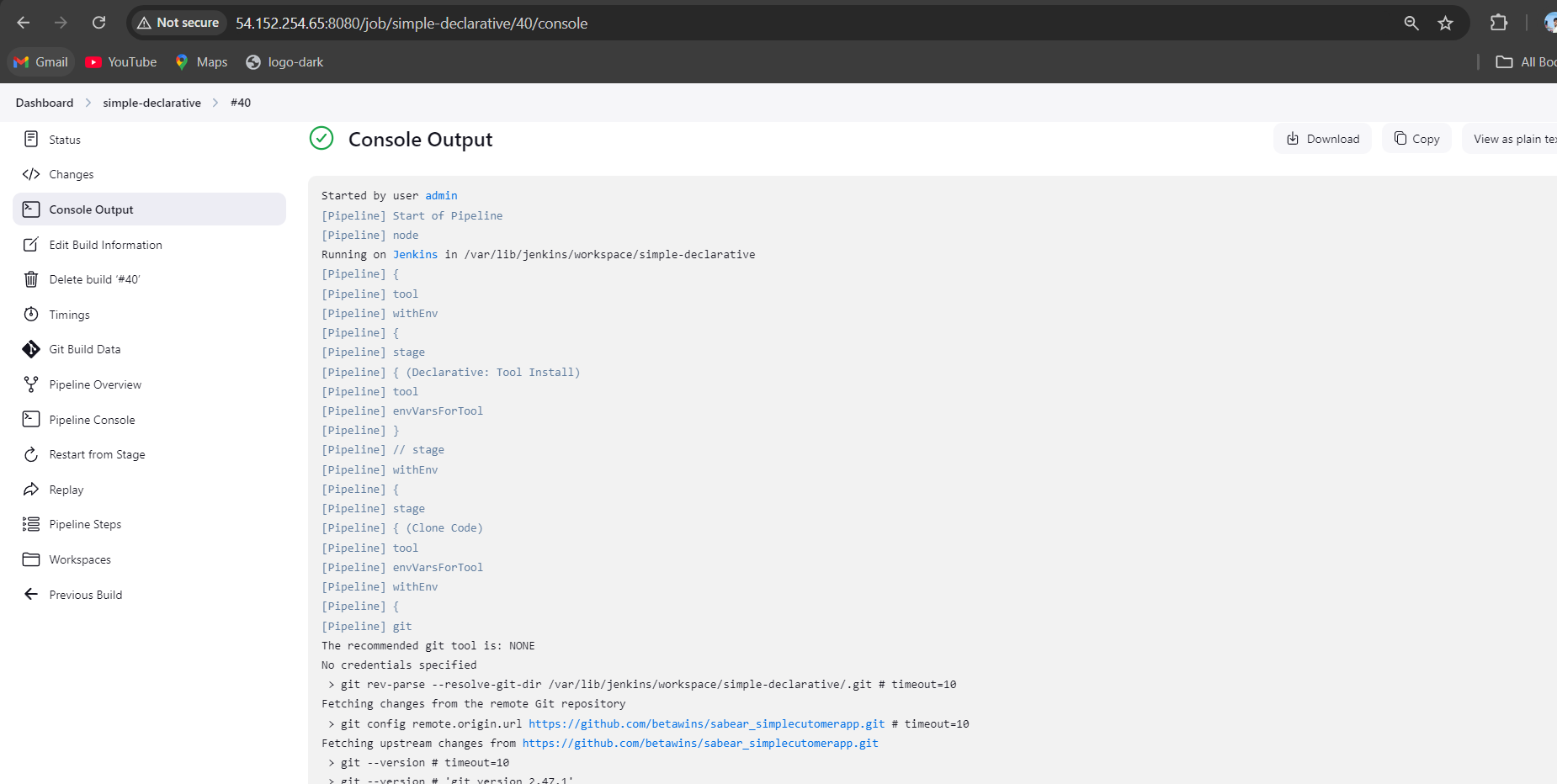


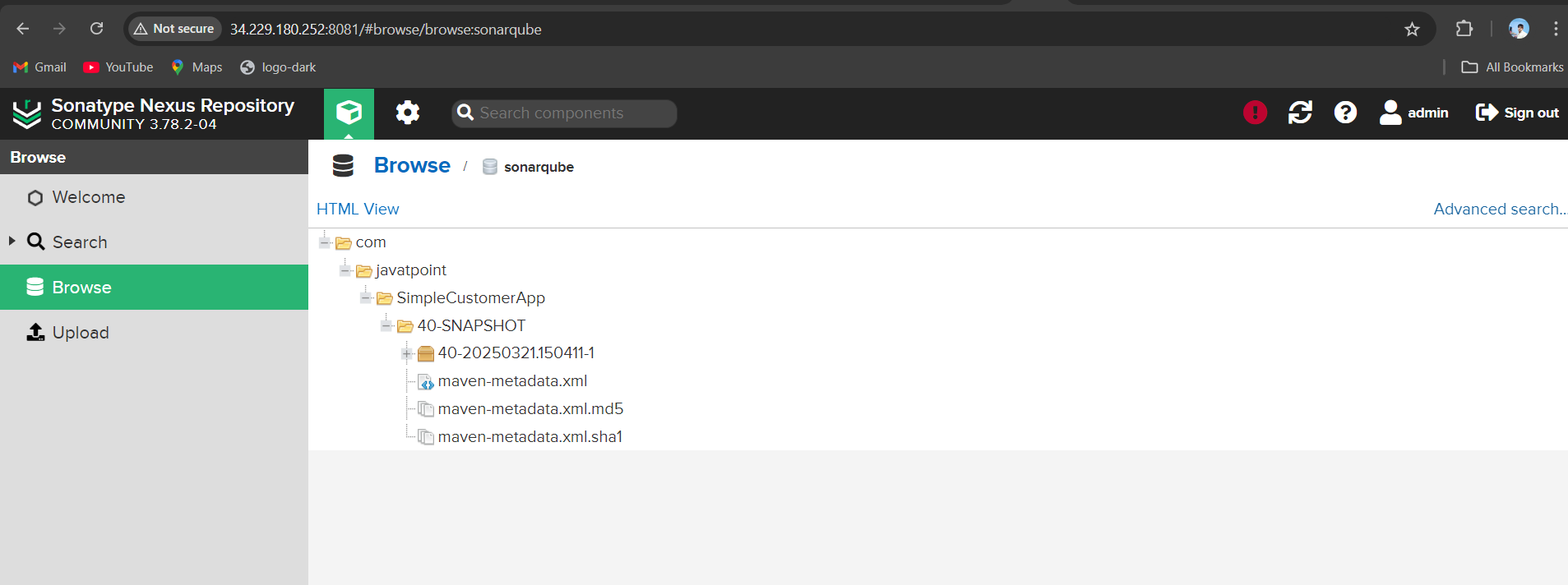
6) Deploy On tomcat

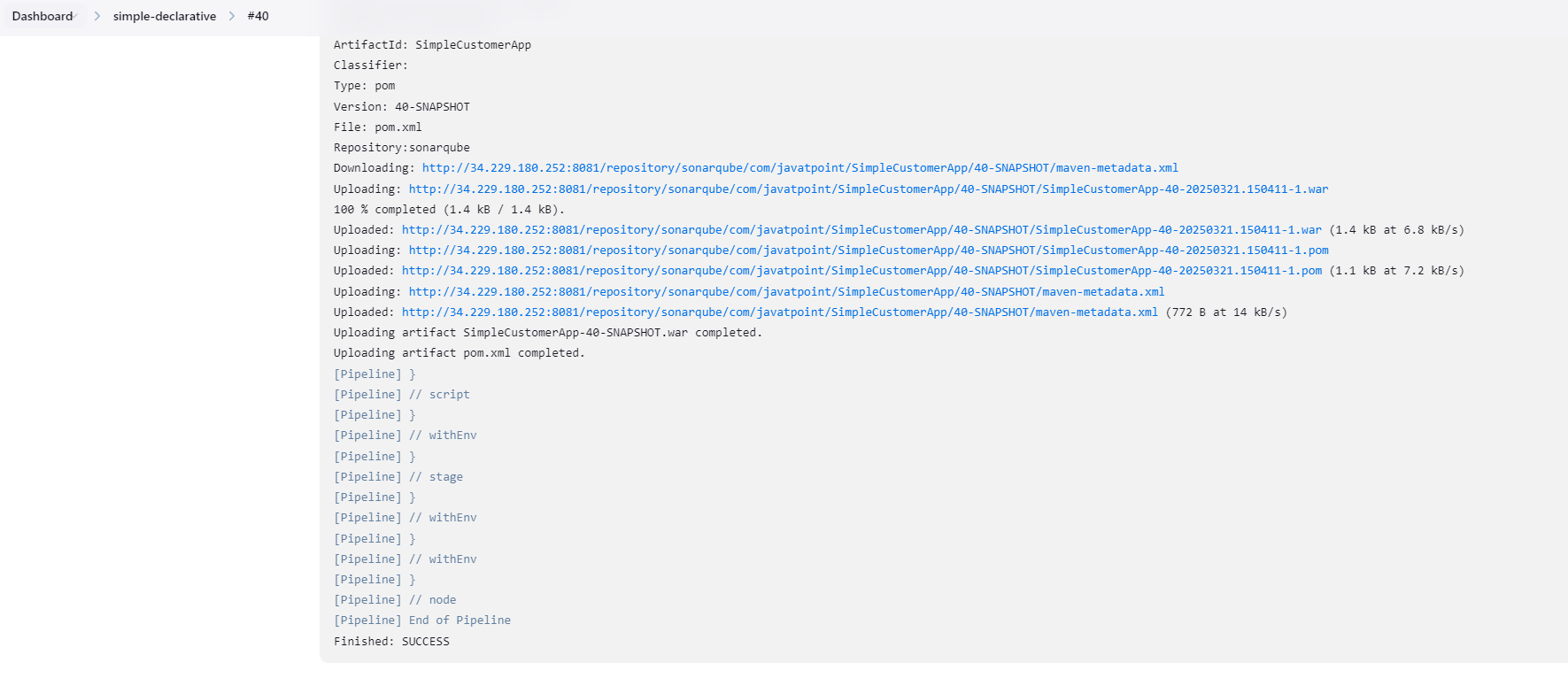




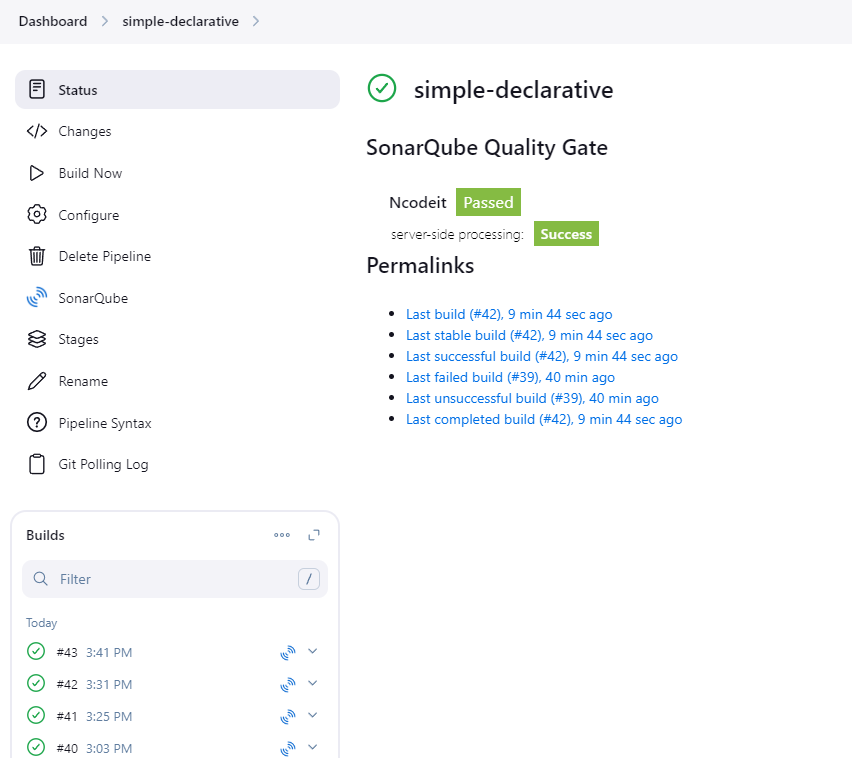












pipeline {

agent any

tools {

maven "MVN\_HOME"

}

environment {

NEXUS\_VERSION = "nexus3"

NEXUS\_PROTOCOL = "http"

NEXUS\_URL = "34.229.180.252:8081"

NEXUS\_REPOSITORY = "sonarqube"

NEXUS\_CREDENTIAL\_ID = "nexus"

SCANNER\_HOME = tool 'sonar\_scanner'

SLACK\_TOKEN = credentials('SLACK\_TOKEN')

}

stages {

stage("Clone Code") {

steps {

git 'https://github.com/betawins/sabear\_simplecutomerapp.git'

}

}

stage("Maven Build") {

steps {

sh 'mvn -Dmaven.test.failure.ignore=true clean install'

}

}

stage("SonarQube Analysis") {

steps {

withSonarQubeEnv('sonarqube\_server') {

sh '${SCANNER\_HOME}/bin/sonar-scanner \

-Dsonar.projectKey=Ncodeit \

-Dsonar.projectName=Ncodeit \

-Dsonar.projectVersion=2.0 \

-Dsonar.sources=src \

-Dsonar.java.binaries=target'

}

}

}

stage("Publish to Nexus") {

steps {

script {

pom = readMavenPom file: "pom.xml"

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

if (filesByGlob.length == 0) {

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

}

artifactPath = filesByGlob[0].path

echo "\*\*\* Uploading: ${artifactPath}, group: ${pom.groupId}, version: ${pom.version}"

nexusArtifactUploader(

nexusVersion: NEXUS\_VERSION,

protocol: NEXUS\_PROTOCOL,

nexusUrl: NEXUS\_URL,

groupId: pom.groupId,

version: pom.version,

repository: NEXUS\_REPOSITORY,

credentialsId: NEXUS\_CREDENTIAL\_ID,

artifacts: [

[artifactId: pom.artifactId, classifier: '', file: artifactPath, type: pom.packaging],

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

]

)

}

}

}

stage("Deploy on Tomcat") {

steps {

script {

def warFile = findFiles(glob: 'target/\*.war')

if (warFile.length == 0) {

error "No WAR file found for deployment!"

}

sh """

curl --upload-file ${warFile[0].path} \

--user admin:admin123 \

http://50.17.71.47:8080/manager/text/deploy?path=/SimpleCustomerApp&update=true

"""

}

}

}

stage("Slack Notification") {

steps {

slackSend(

channel: "#jenkins-integration",

tokenCredentialId: "SLACK\_TOKEN",

message: "Build Completed: ${currentBuild.fullDisplayName}"

)

}

}

}

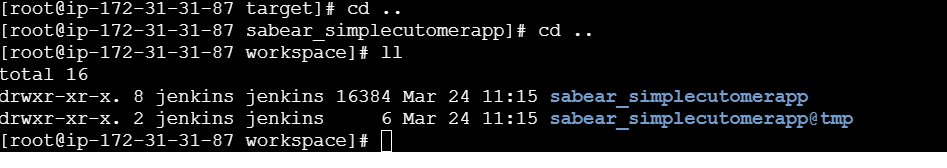
}

3) Setup a Jenkins CICD pipeline using Scripted pipeline using feature-1.1 branch.

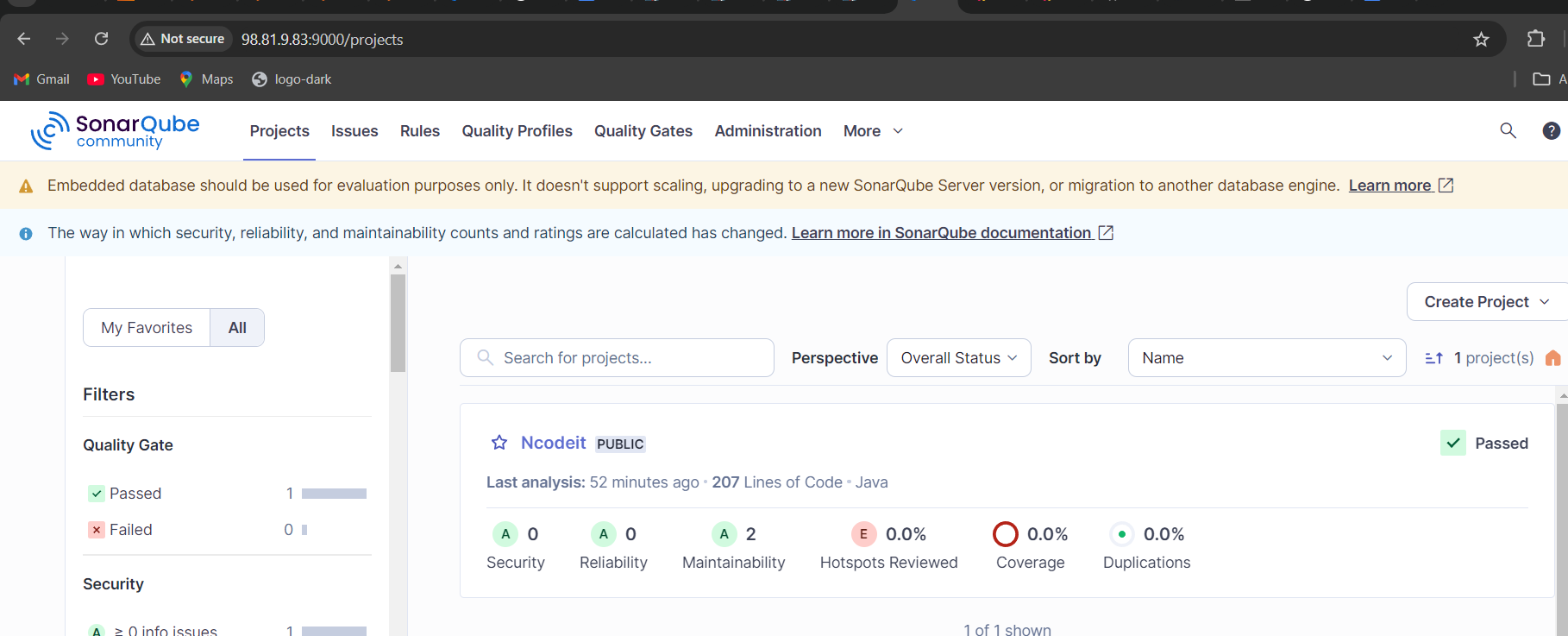
https://github.com/betawins/sabear\_simplecutomerapp/tree/feature-1.1

stages:

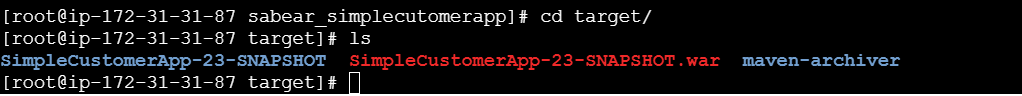
1) Git Clone



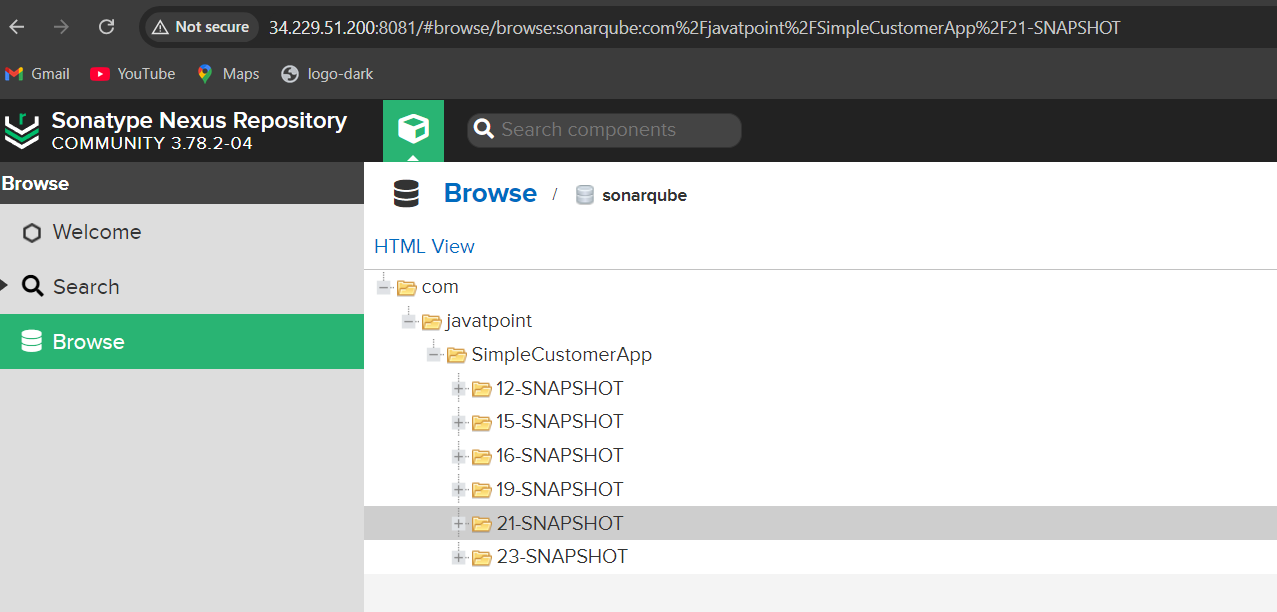
2) Sonarqube Integration



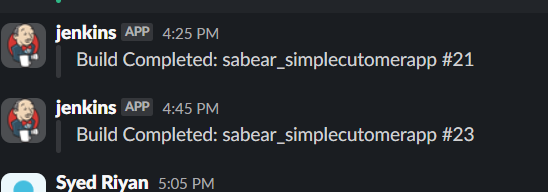
3) Maven Compilation



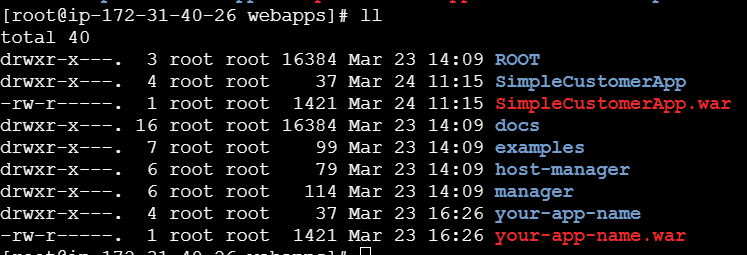
4) Nexus Artifactory



5) Slack Notification



6) Deploy On tomcat



node {

def NEXUS\_VERSION = "nexus3"

def NEXUS\_PROTOCOL = "http"

def NEXUS\_URL = "34.229.51.200:8081"

def NEXUS\_REPOSITORY = "sonarqube"

def NEXUS\_CREDENTIAL\_ID = "nexus"

def SCANNER\_HOME = tool 'sonar\_scanner'

def SLACK\_TOKEN = credentials('SLACK\_TOKEN')

stage("Clone Code") {

git 'https://github.com/betawins/sabear\_simplecutomerapp.git'

}

stage("Maven Build") {

sh 'mvn -Dmaven.test.failure.ignore=true clean install'

}

stage("SonarQube Analysis") {

withSonarQubeEnv('sonarqube\_server') {

sh """

${SCANNER\_HOME}/bin/sonar-scanner \

-Dsonar.projectKey=Ncodeit \

-Dsonar.projectName=Ncodeit \

-Dsonar.projectVersion=2.0 \

-Dsonar.sources=src \

-Dsonar.java.binaries=target

"""

}

}

stage("Publish to Nexus") {

script {

nexusArtifactUploader(

artifacts: [[

artifactId: 'SimpleCustomerApp',

classifier: '',

file: "target/SimpleCustomerApp-${BUILD\_NUMBER}-SNAPSHOT.war",

type: 'war'

]],

credentialsId: "${NEXUS\_CREDENTIAL\_ID}",

groupId: 'com.javatpoint',

nexusUrl: "${NEXUS\_URL}",

nexusVersion: "${NEXUS\_VERSION}",

protocol: "${NEXUS\_PROTOCOL}",

repository: "${NEXUS\_REPOSITORY}",

version: "${BUILD\_NUMBER}-SNAPSHOT"

)

}

}

stage("Deploy on Tomcat") {

script {

deploy adapters: [tomcat9(credentialsId: 'TOMCAT\_CREDENTIALS', path: '', url: 'http://3.86.14.58:8080/')],

contextPath: 'SimpleCustomerApp',

war: "target/SimpleCustomerApp-${BUILD\_NUMBER}-SNAPSHOT.war"

}

}

stage("Slack Notification") {

script {

slackSend(

channel: "#jenkins-integration",

tokenCredentialId: "SLACK\_TOKEN",

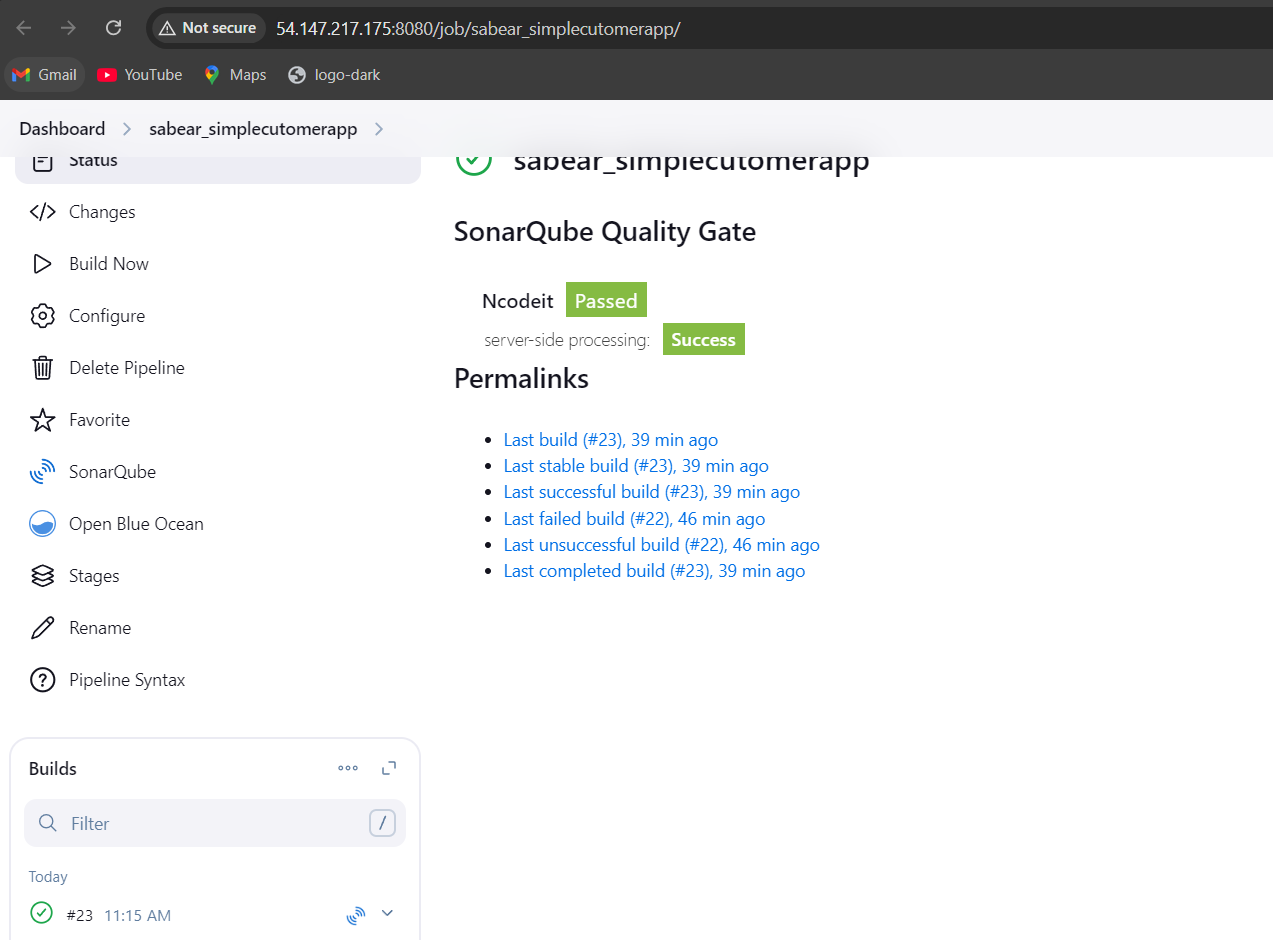
message: "Build Completed: ${currentBuild.fullDisplayName}"

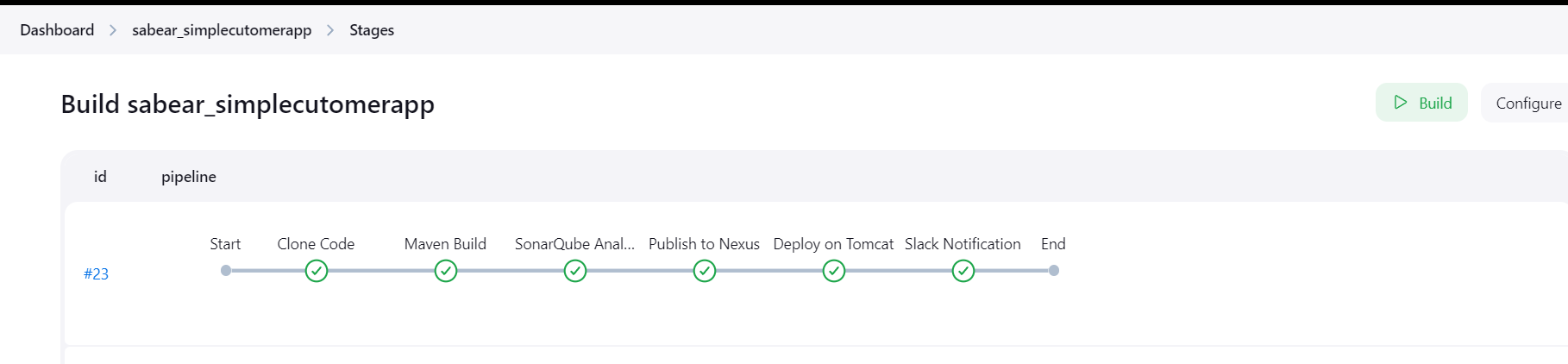
)

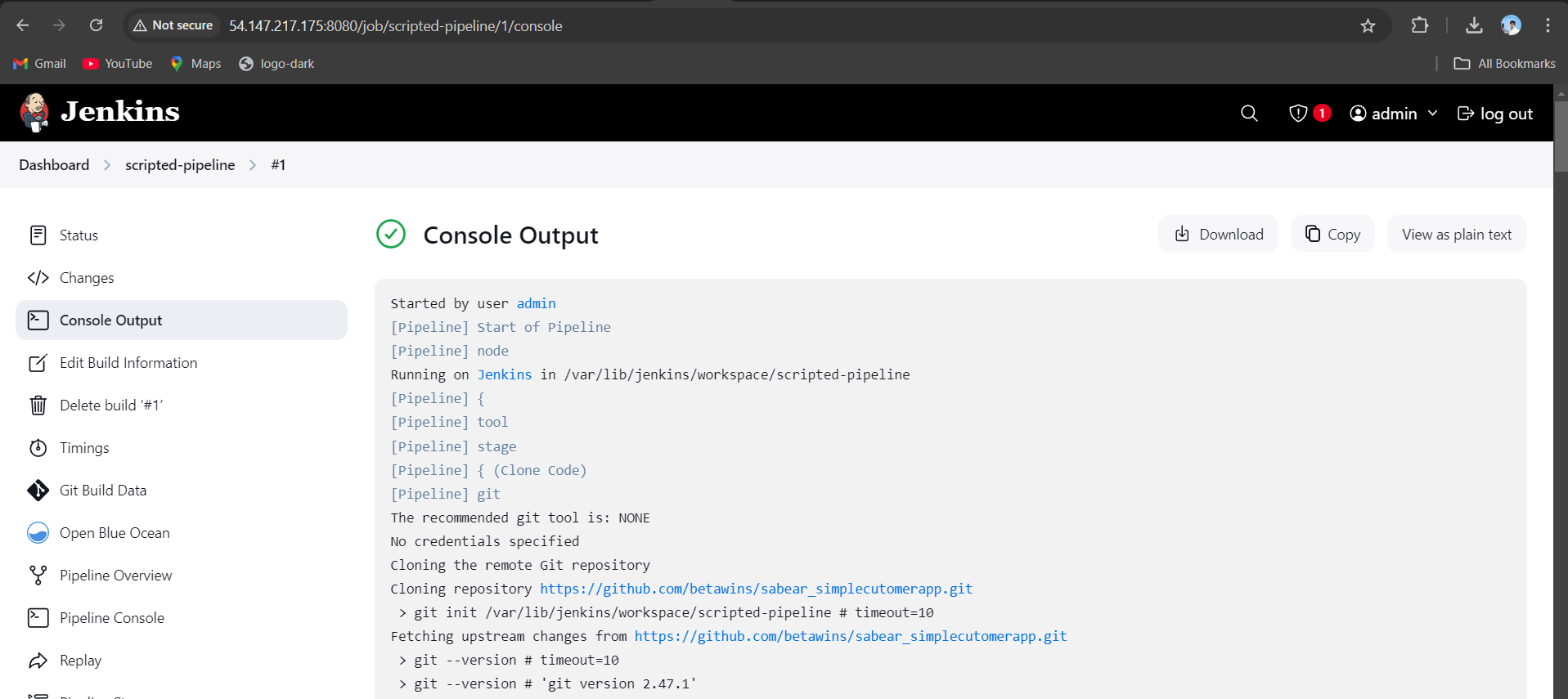
}

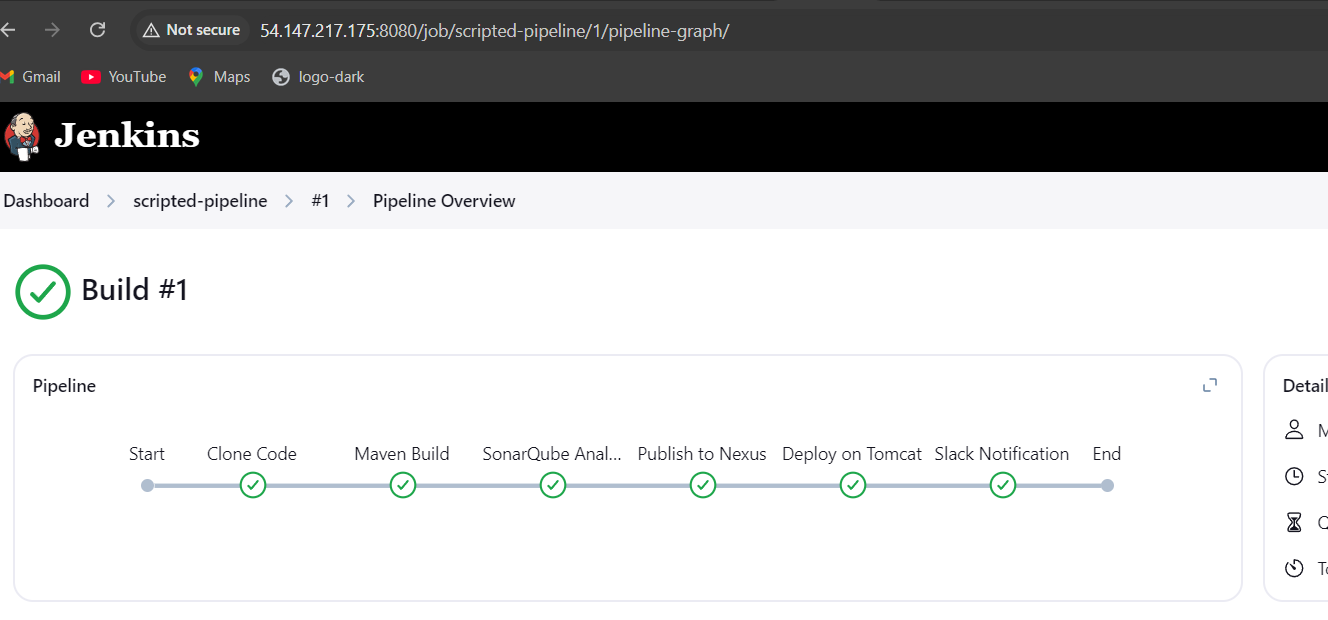
}

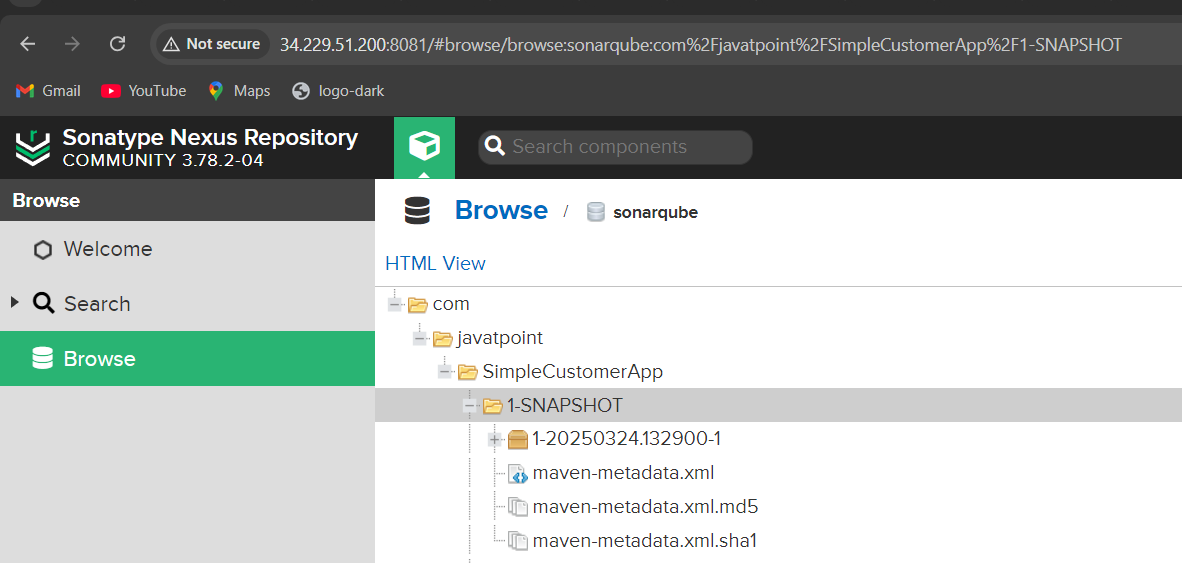
}

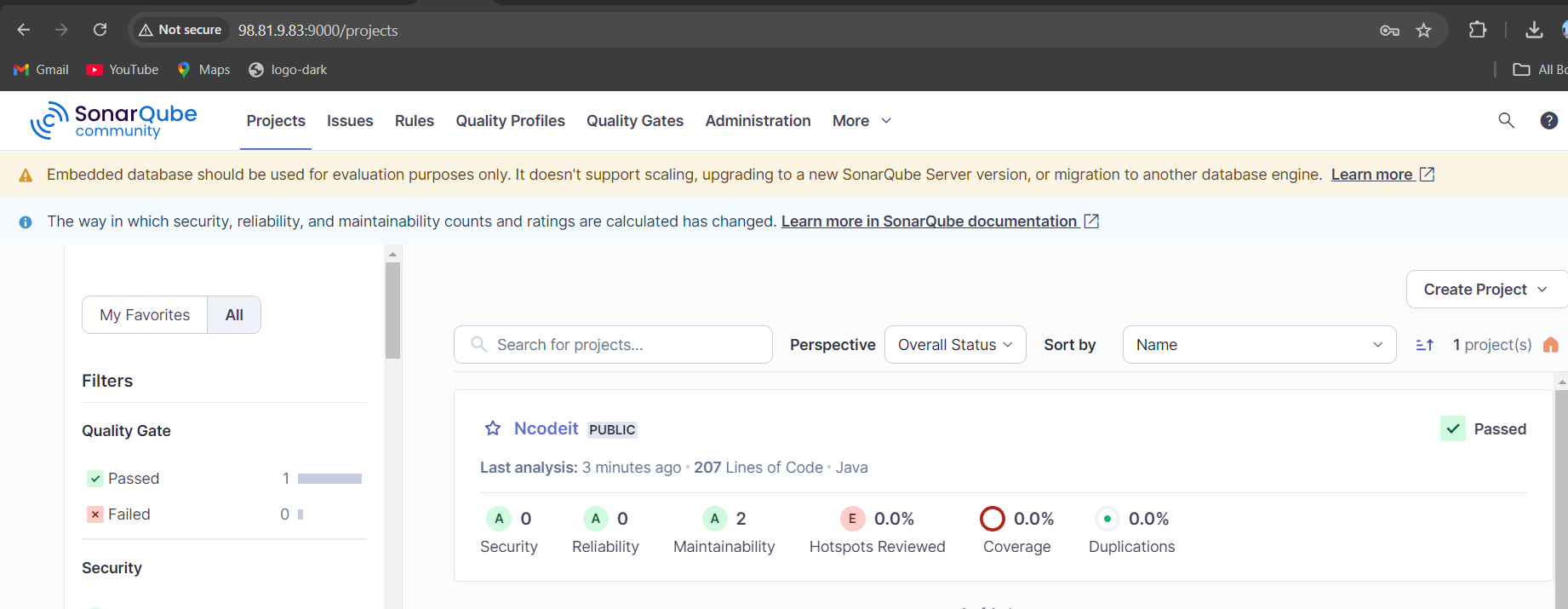


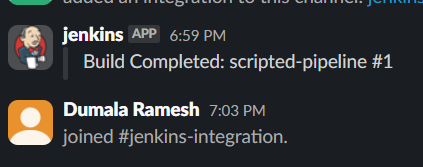




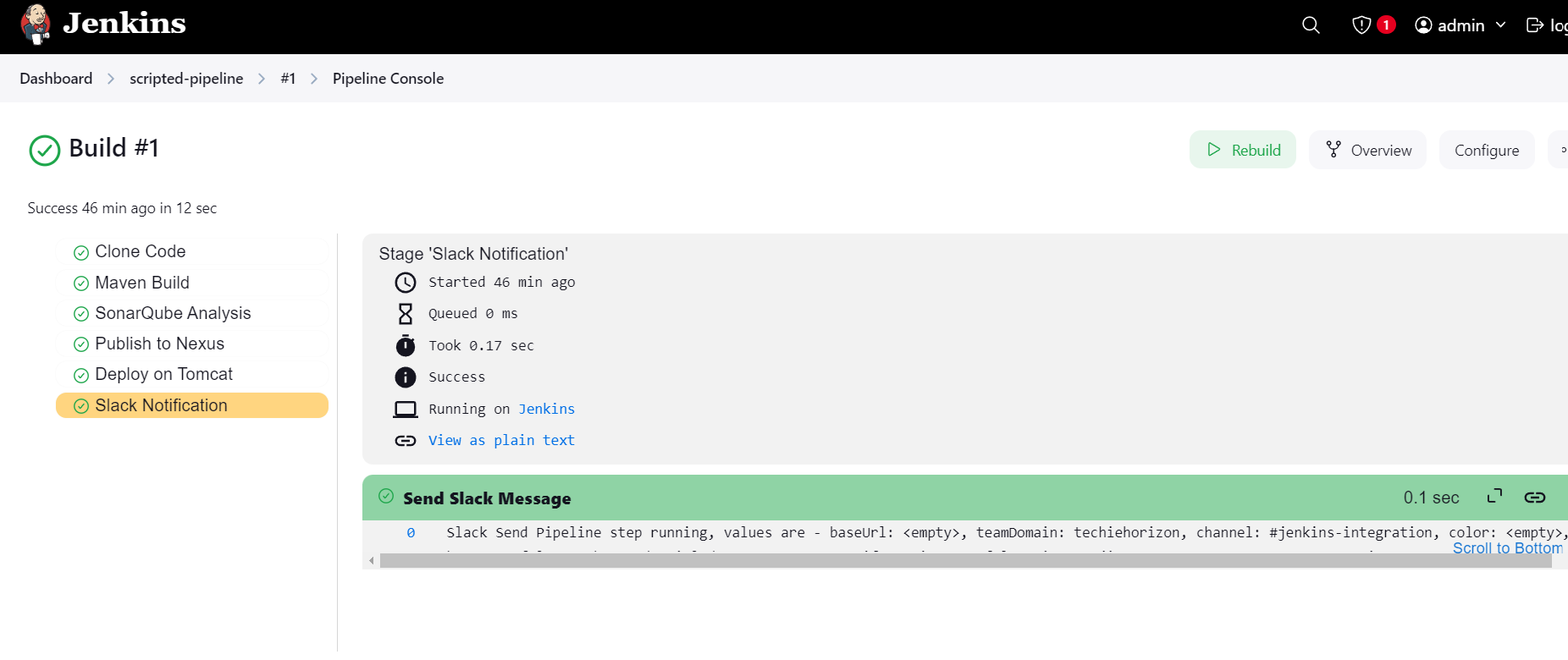


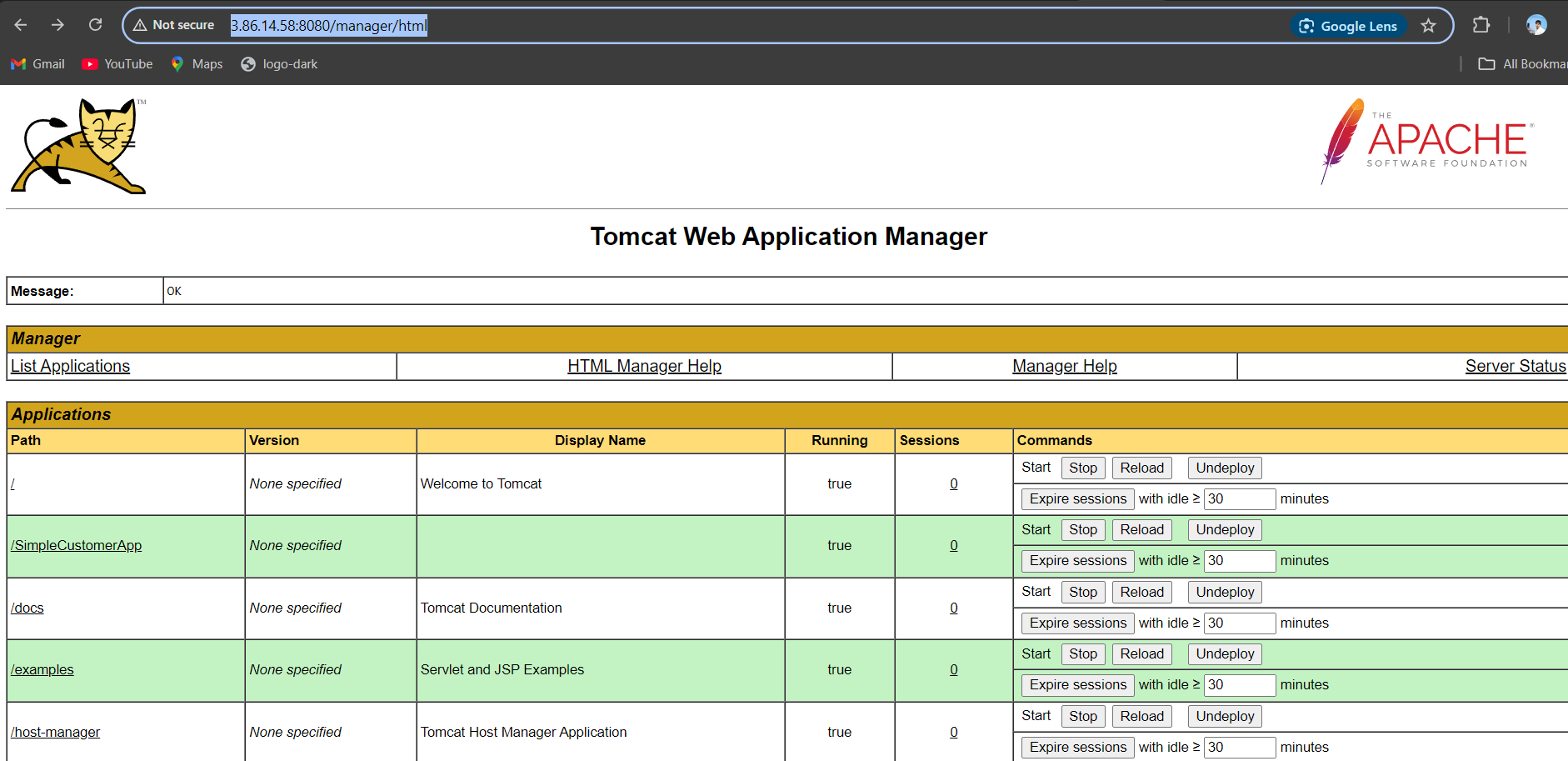


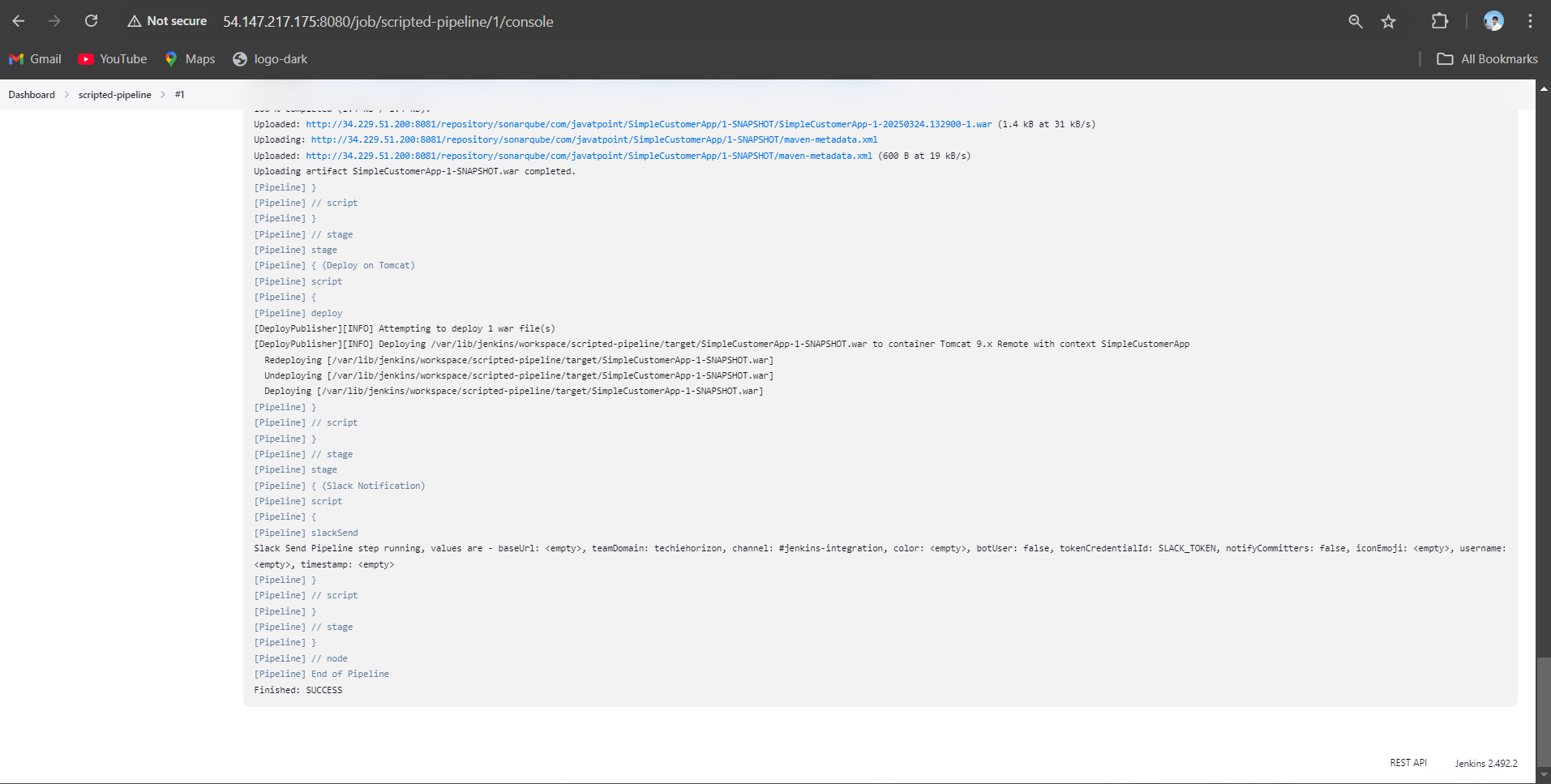












4) Write sample skeleton of pipelines.

pipeline {

agent any

stages {

stage('Checkout') {

steps {

echo 'Cloning repository...'

git url: 'https://github.com/your-repo/sample-project.git', branch: 'main'

}

}

stage('Build') {

steps {

echo 'Building the project...'

sh 'mvn clean package'

}

}

stage('Test') {

steps {

echo 'Running tests...'

sh 'mvn test'

}

}

stage('Deploy') {

steps {

echo 'Deploying application...'

// Add deployment script here

}

}

}

post {

success {

echo 'Pipeline execution completed successfully!'

}

failure {

echo 'Pipeline execution failed!'

}

}

}

node {

// Define environment variables

def SONARQUBE\_URL = 'http://your-sonarqube-server:9000'

def NEXUS\_URL = 'http://your-nexus-server:8081/repository/maven-releases/'

try {

stage('Checkout') {

// Pull latest code from GitHub repository

git url: 'https://github.com/your-repo/sample-project.git', branch: 'main'

}

stage('Build') {

// Compile and package the application using Maven

sh 'mvn clean package'

}

stage('Test') {

// Run unit tests to ensure application stability

sh 'mvn test'

}

stage('Code Analysis') {

// Analyze code quality with SonarQube

sh "mvn sonar:sonar -Dsonar.host.url=${SONARQUBE\_URL} -Dsonar.login=your-token"

}

stage('Artifact Storage') {

// Deploy built artifact (WAR/JAR) to Nexus Repository

sh "mvn deploy -DaltDeploymentRepository=nexus-releases::default::${NEXUS\_URL}"

}

stage('Deploy to Staging') {

// Transfer artifact to staging server using SCP

sh 'scp target/\*.war user@staging-server:/path/to/deploy/'

}

stage('Deploy to Production') {

// Automated deployment to production server

sh 'scp target/\*.war user@production-server:/path/to/deploy/'

// Restart application to reflect new changes

sh 'ssh user@production-server "sudo systemctl restart tomcat"'

}

} catch (Exception e) {

// Handle failures and send notifications if necessary

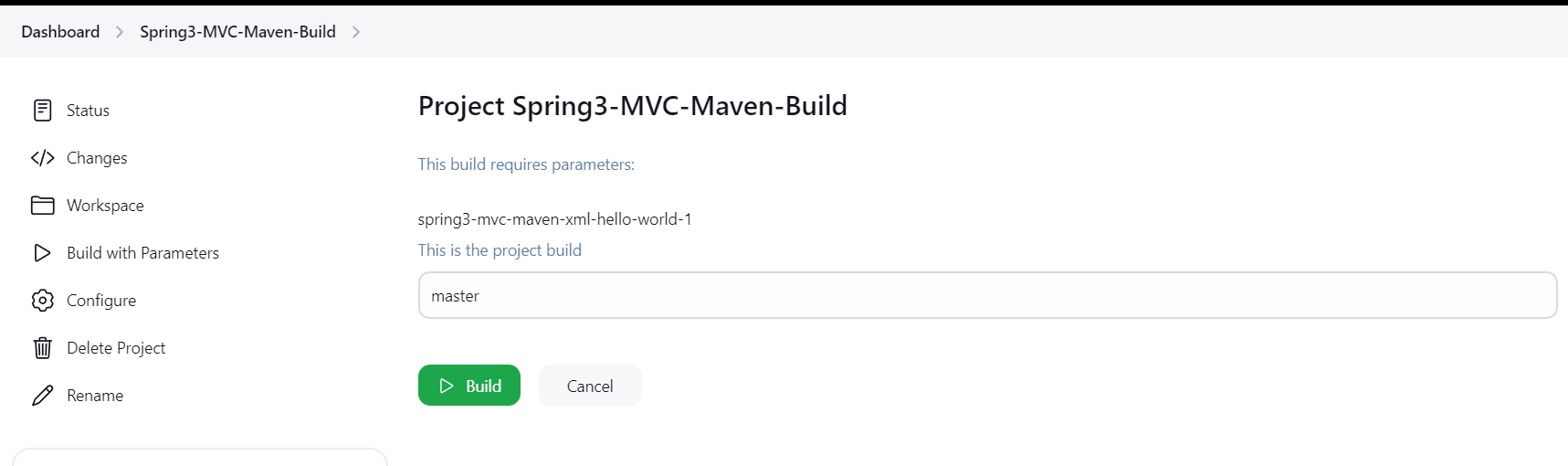
echo "Pipeline failed due to: ${e}"

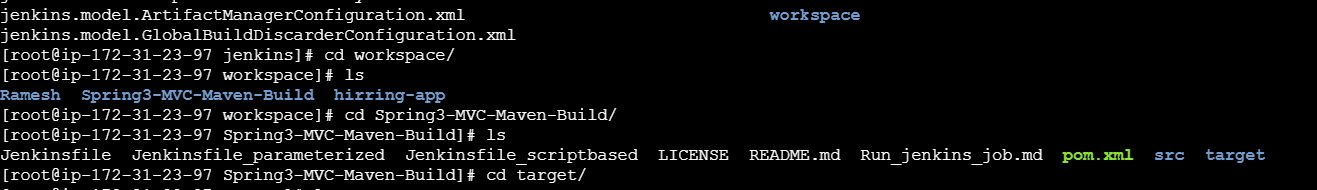
}

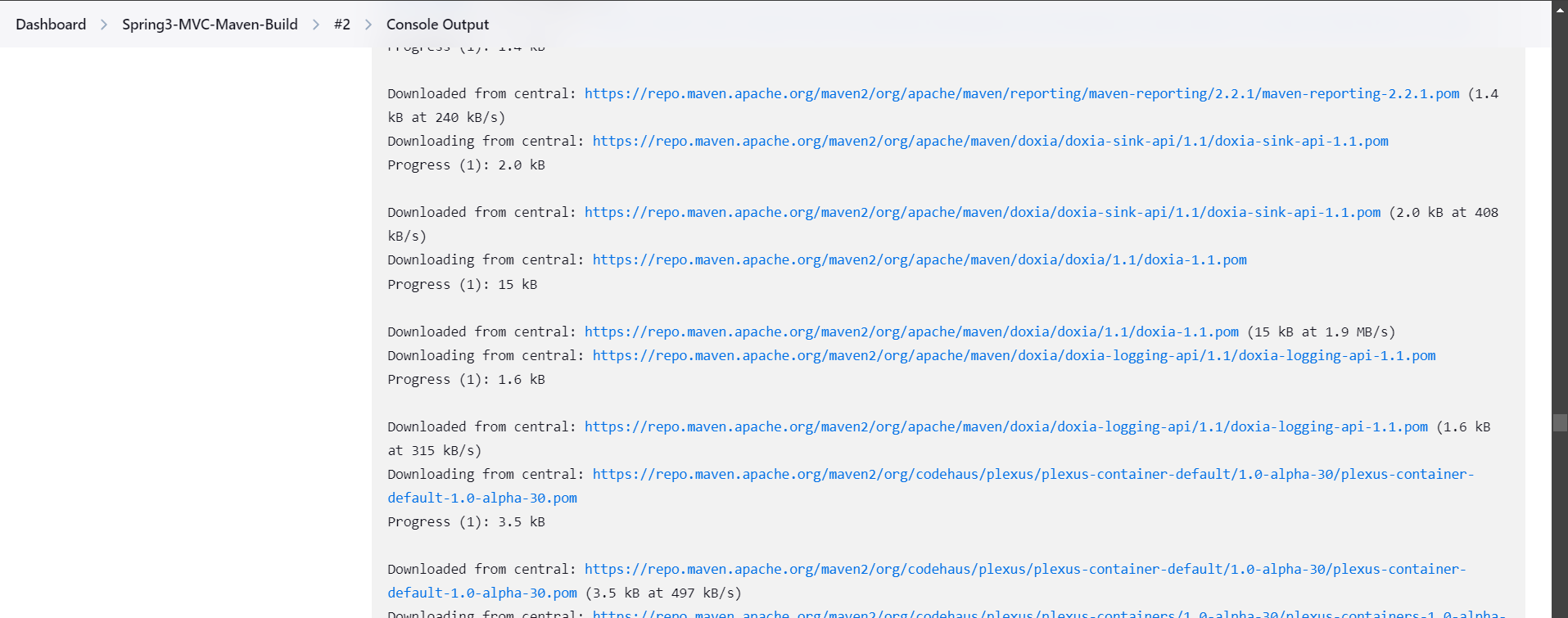
}

5) Create a parameterized job in jenkins.

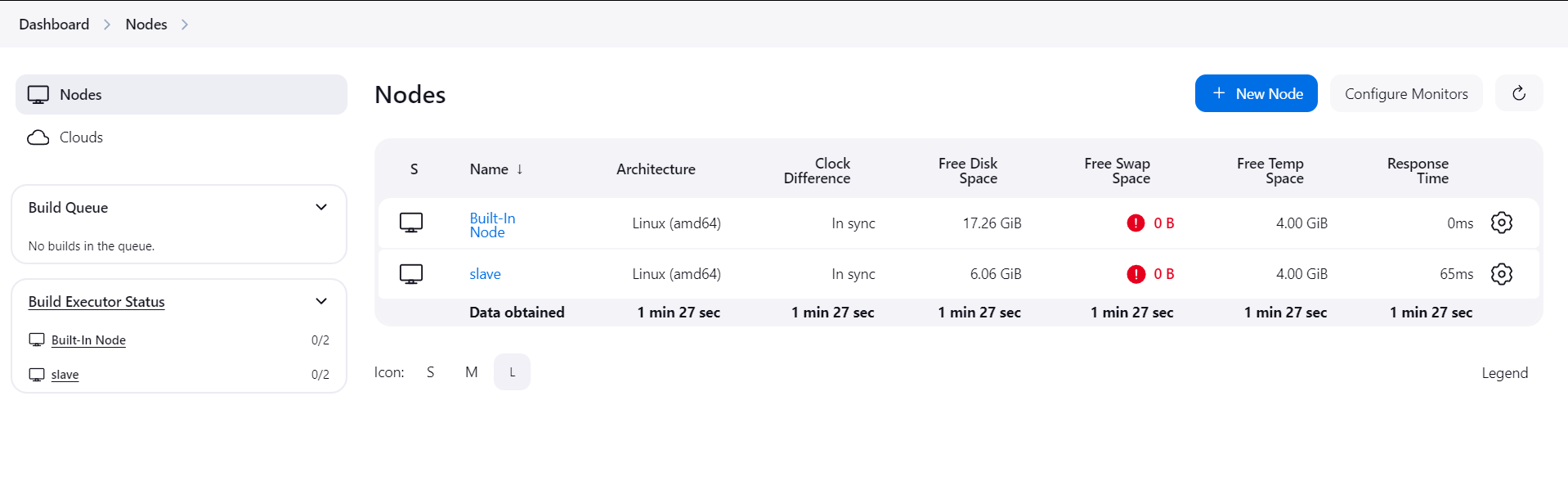
https://github.com/betawins/spring3-mvc-maven-xml-hello-world-1.git

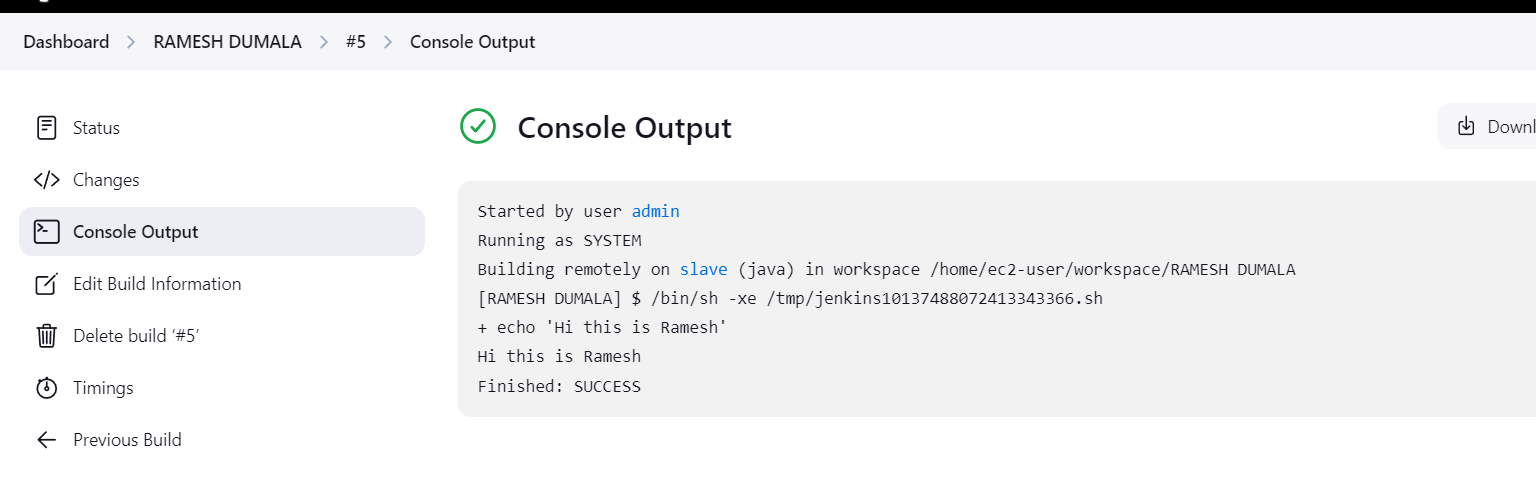


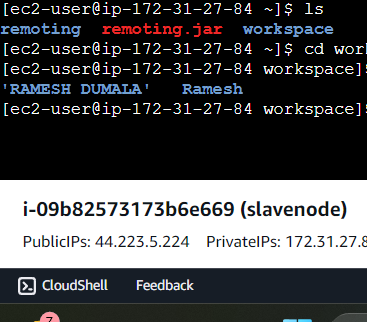


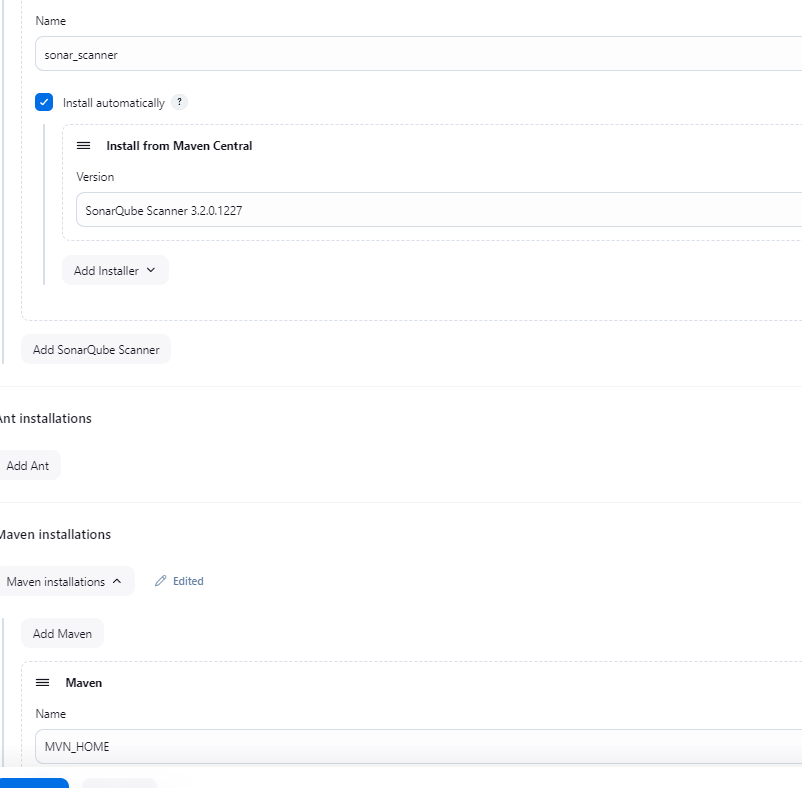


6) Setup one slave machine for jenkins.









pipeline {

agent any

tools {

// Note: this should match with the tool name configured in your jenkins instance (JENKINS\_URL/configureTools/)

maven "MVN\_HOME"

}

environment {

// This can be nexus3 or nexus2

NEXUS\_VERSION = "nexus3"

// This can be http or https

NEXUS\_PROTOCOL = "http"

// Where your Nexus is running

NEXUS\_URL = "3.81.184.134:8081/"

// Repository where we will upload the artifact

NEXUS\_REPOSITORY = "sonarqube"

// Jenkins credential id to authenticate to Nexus OSS

NEXUS\_CREDENTIAL\_ID = "nexus\_keygen"

SCANNER\_HOME = tool 'sonar\_scanner'

}

stages {

stage("clone code") {

steps {

script {

// Let's clone the source

git '<https://github.com/betawins/sabear_simplecutomerapp.git>';

}

}

}

stage("mvn build") {

steps {

script {

// If you are using Windows then you should use "bat" step

// Since unit testing is out of the scope we skip them

sh 'mvn -Dmaven.test.failure.ignore=true clean install'

}

}

}

stage('SonarCloud') {

steps {

withSonarQubeEnv('sonarqube\_server') {

sh '$SCANNER\_HOME/bin/sonar-scanner \

-Dsonar.projectKey=Ncodeit \

-Dsonar.projectName=Ncodeit \

-Dsonar.projectVersion=2.0 \

-Dsonar.sources=/var/lib/jenkins/workspace/$JOB\_NAME/src/ \

-Dsonar.binaries=target/classes/com/visualpathit/account/controller/ \

-Dsonar.junit.reportsPath=target/surefire-reports \

-Dsonar.jacoco.reportPath=target/jacoco.exec \

-Dsonar.java.binaries=src/com/room/sample '

}

}

}

stage("publish to nexus") {

steps {

script {

nexusArtifactUploader artifacts: [[artifactId: 'SimpleCustomerApp', classifier: '', file: '/var/lib/jenkins/workspace/pipeline/target/SimpleCustomerApp-${BUILD\_NUMBER}-SNAPSHOT.war', type: 'war']], credentialsId: 'nexus\_keygen', groupId: 'com.javatpoint', nexusUrl: '3.81.184.134:8081', nexusVersion: 'nexus3', protocol: 'http', repository: 'sonarqube', version: '${BUILD\_NUMBER}-SNAPSHOT'

}

}

}

}

post {

success {

slackSend(

channel: '#jenkins-integration',

message: ":white_check_mark: Build Successful! :rocket:",

teamDomain: 'techiehorizon',

tokenCredentialId: 'raghu'

)

}

failure {

slackSend(

channel: '#jenkins-integration',

message: ":x: Build Failed! Check Jenkins logs.",

teamDoman: 'techiehorizon',

tokenCredentialId: 'raghu'

)

}

}

}