2) Setup a jenkins CICD pipeline using Declarative pipeline using feature-1.1 branch. <https://github.com/betawins/sabear_simplecutomerapp/tree/feature-1.1>

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'

}

}

}

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 deployer:deployer \

http://3.94.89.70:8080/manager/text/deploy?path=/SimpleCustomerApp&update=true

"""

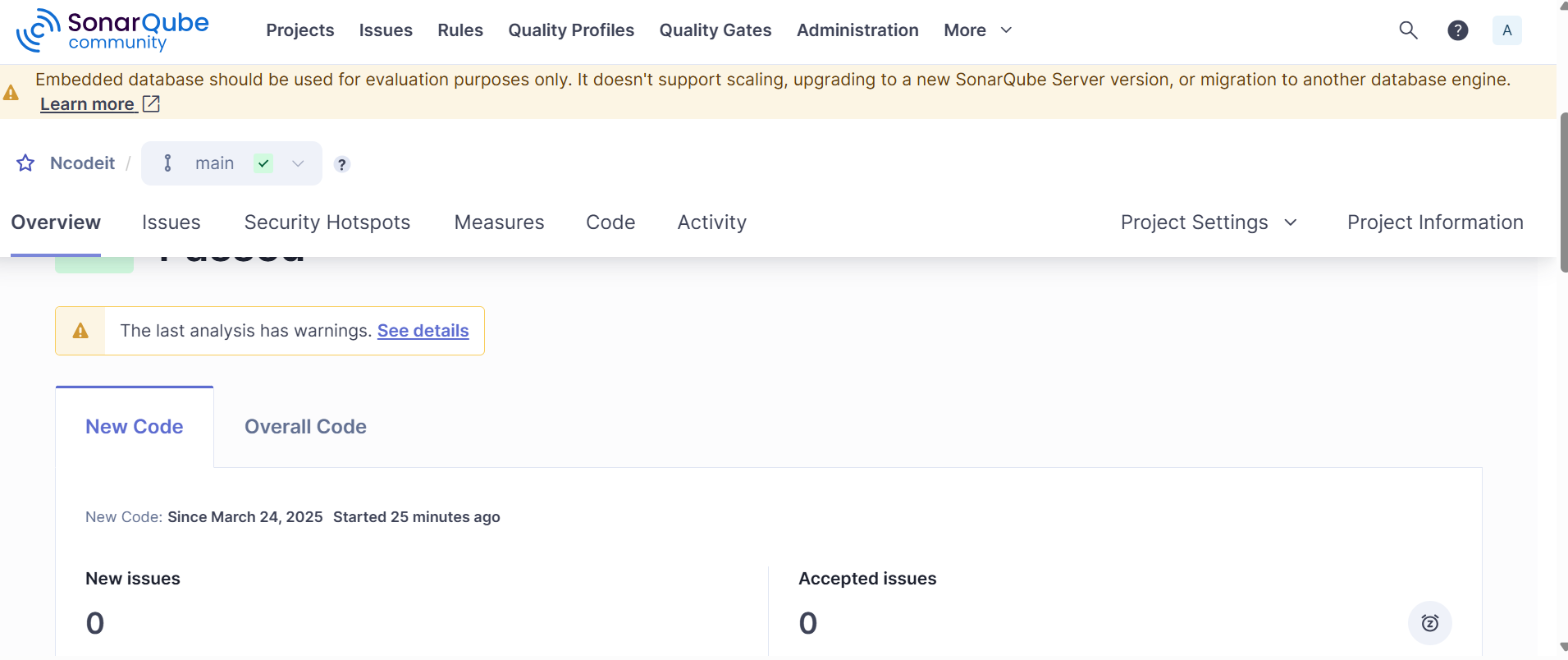
}

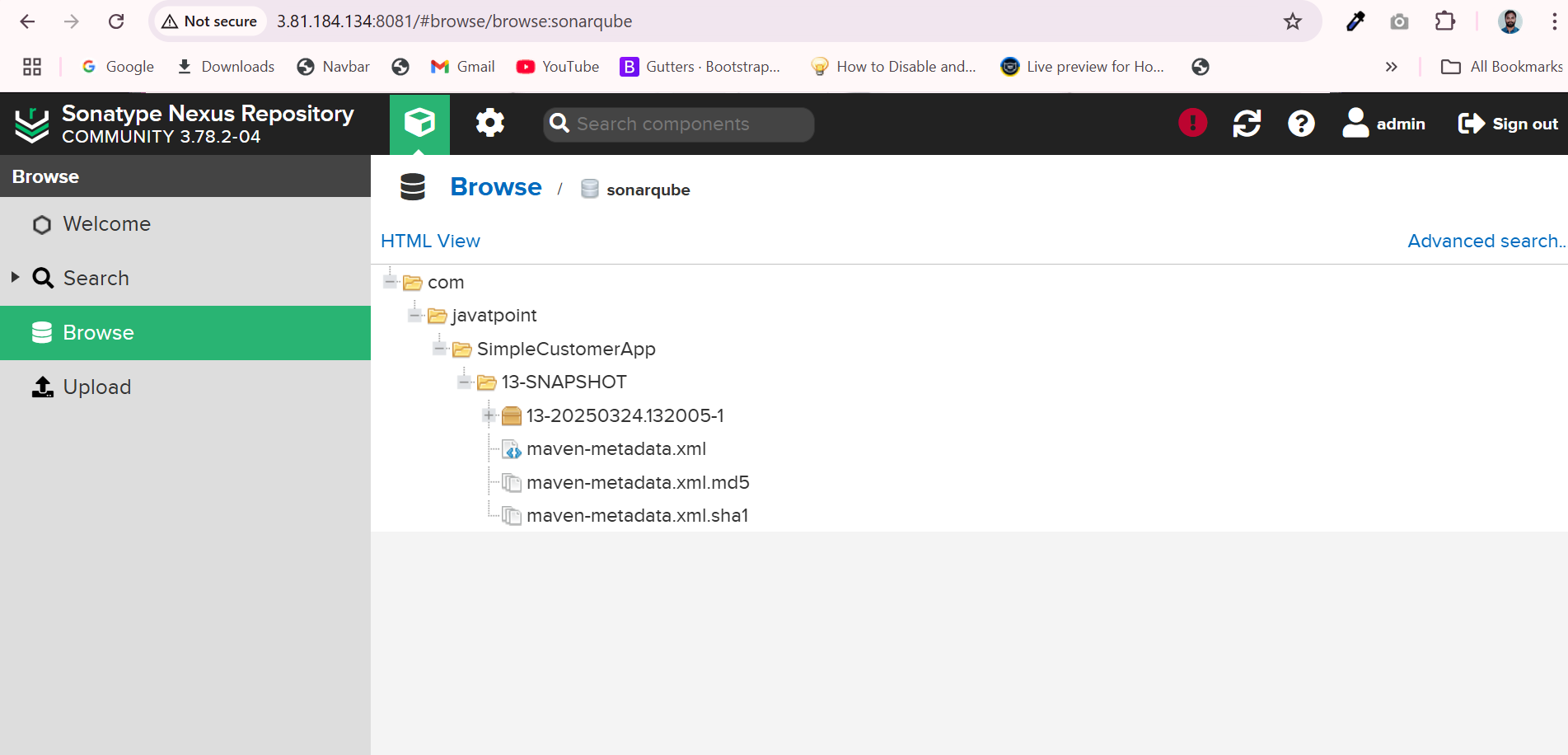
}

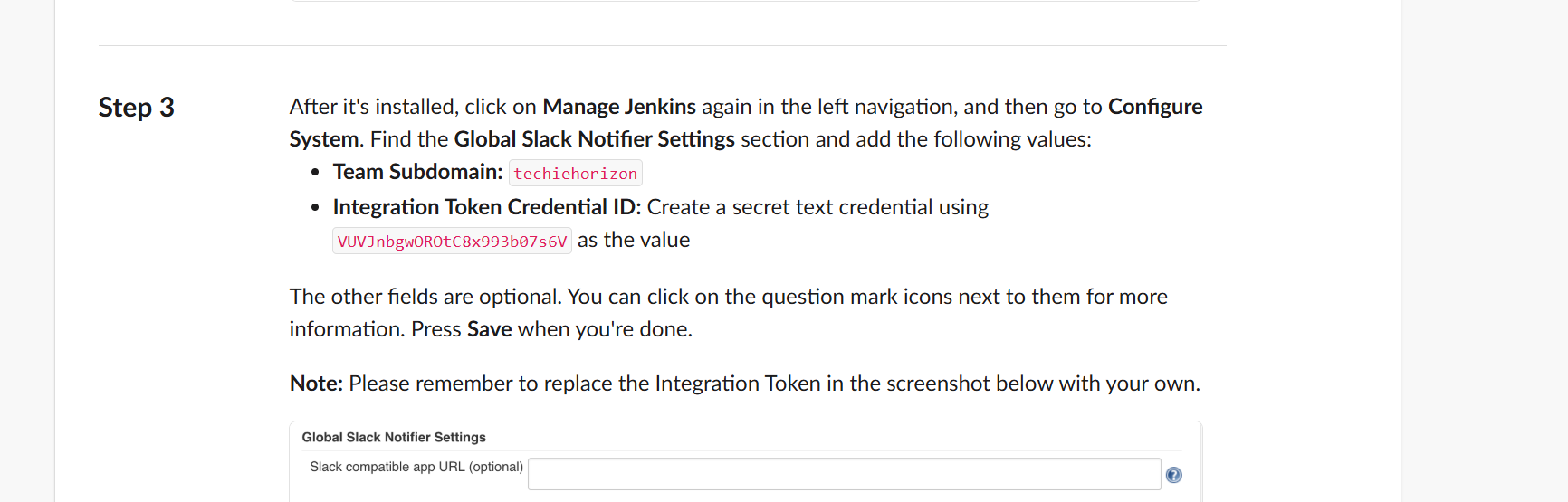
}

}

}







stages: 1) Git Clone

stages {

stage("clone code") {

steps {

script {

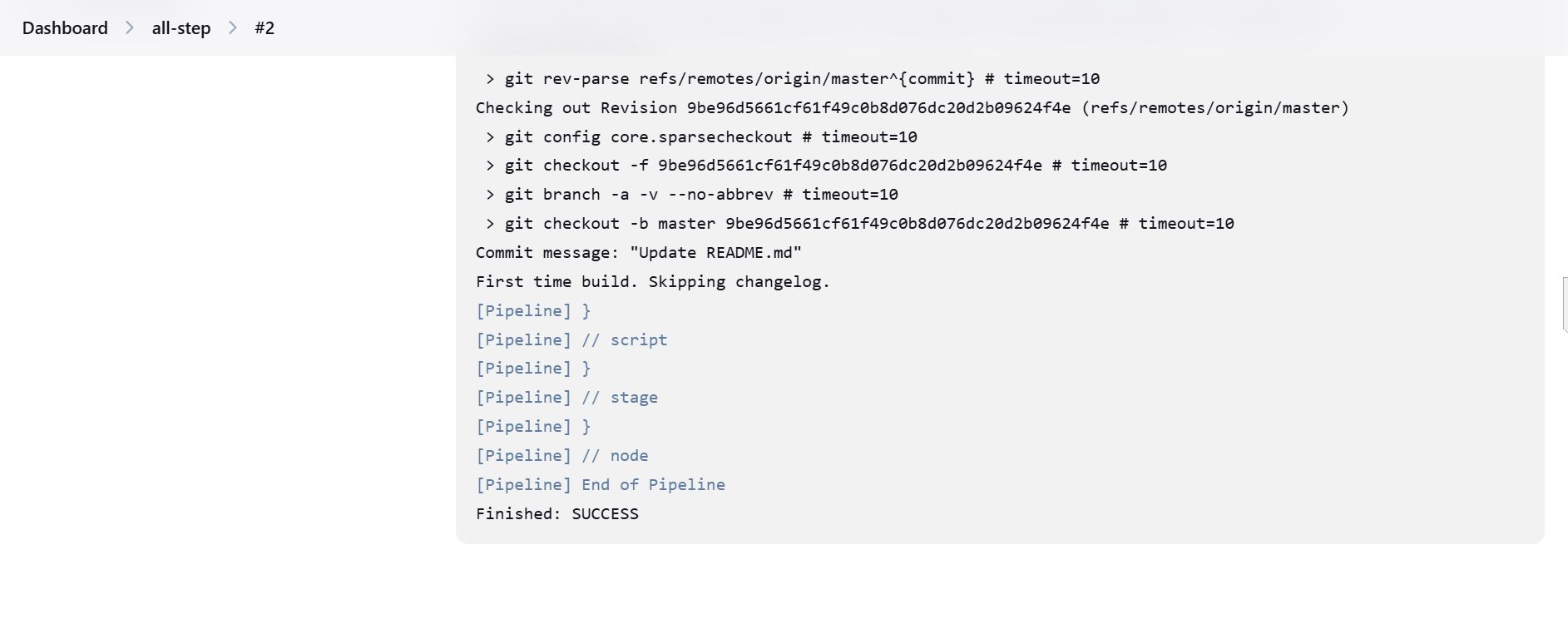
// Let's clone the source

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

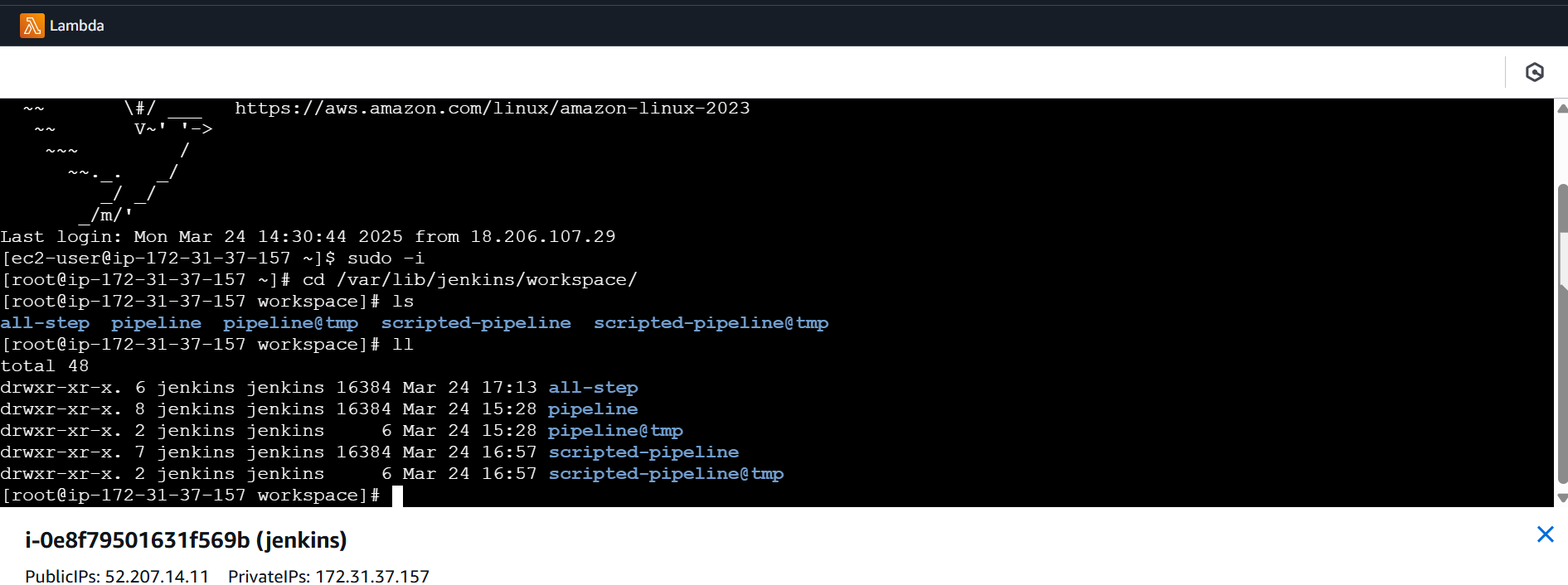
}

}

}



Output: only clone git



2) Sonarqube Integration

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 \

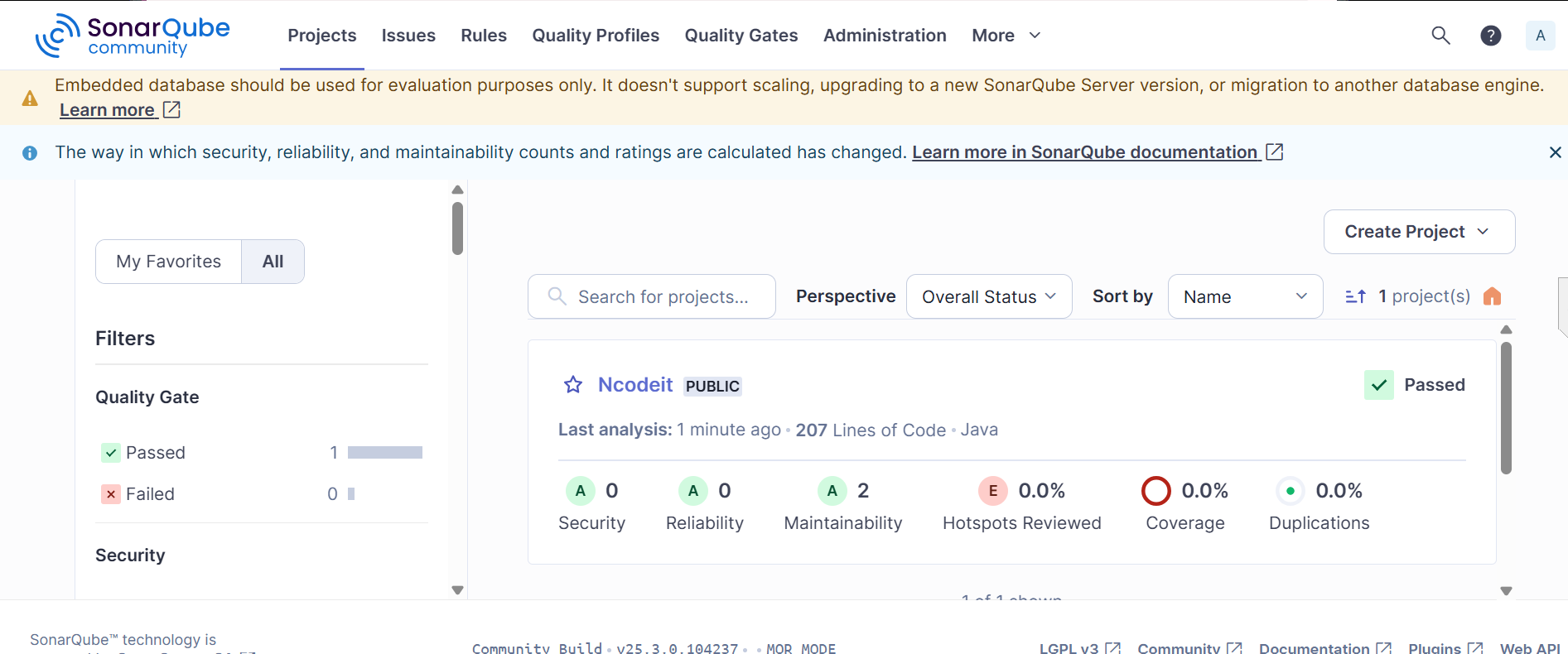
                -Dsonar.jacoco.reportPath=target/jacoco.exec \

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

             }

        }

        }



3) Maven Compilation

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('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("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

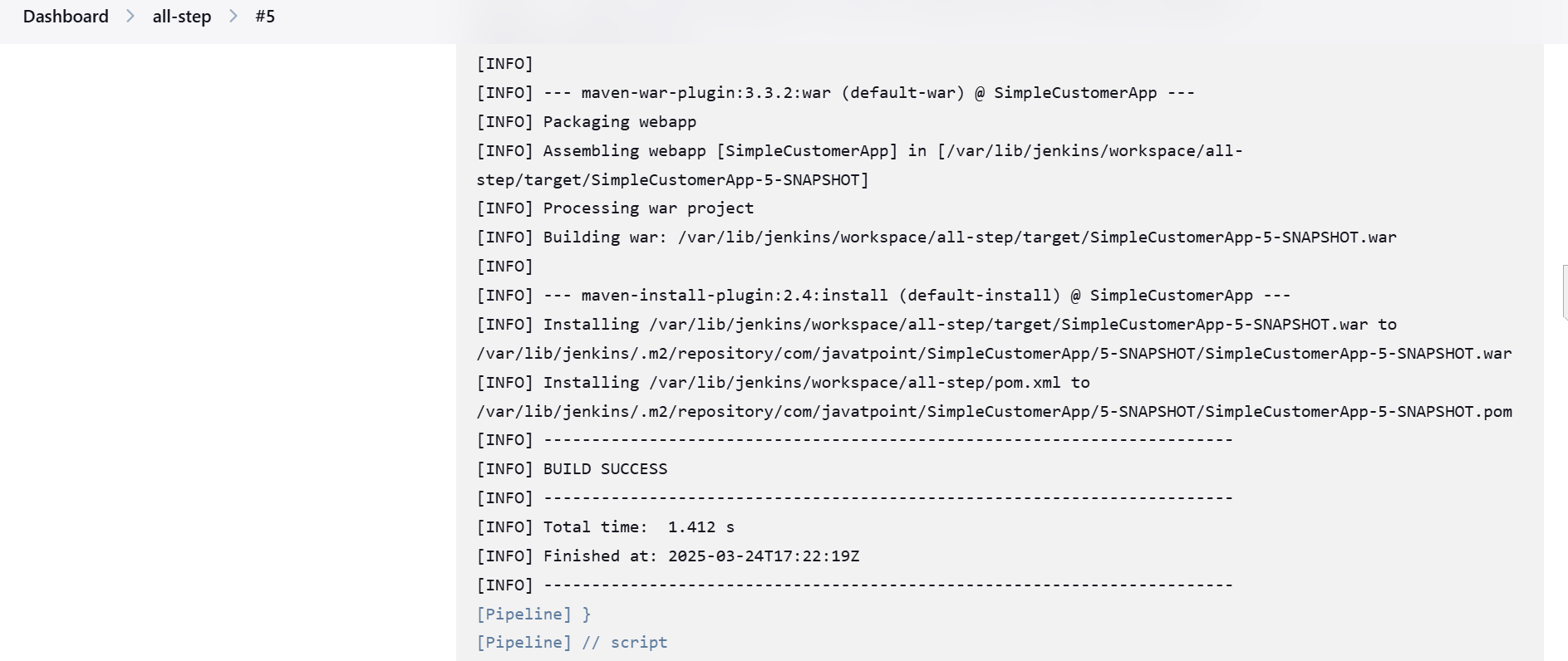
sh 'mvn -Dmaven.test.failure.ignore=true clean install'

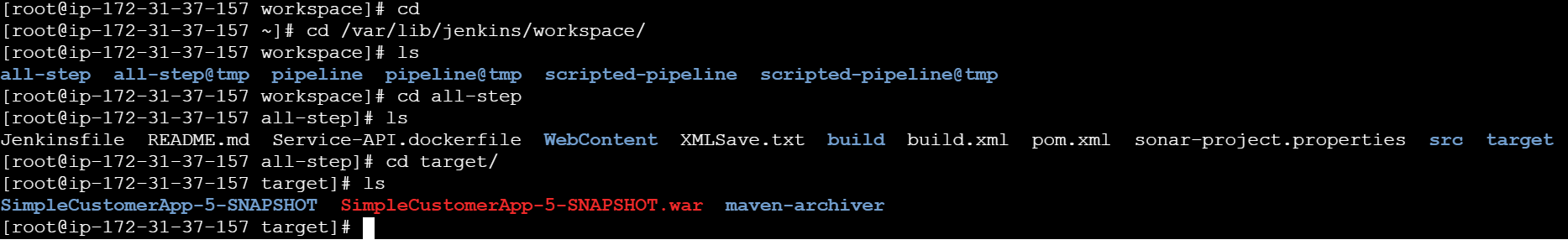
}

}

}

}





4) Nexus Artifactory

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 '

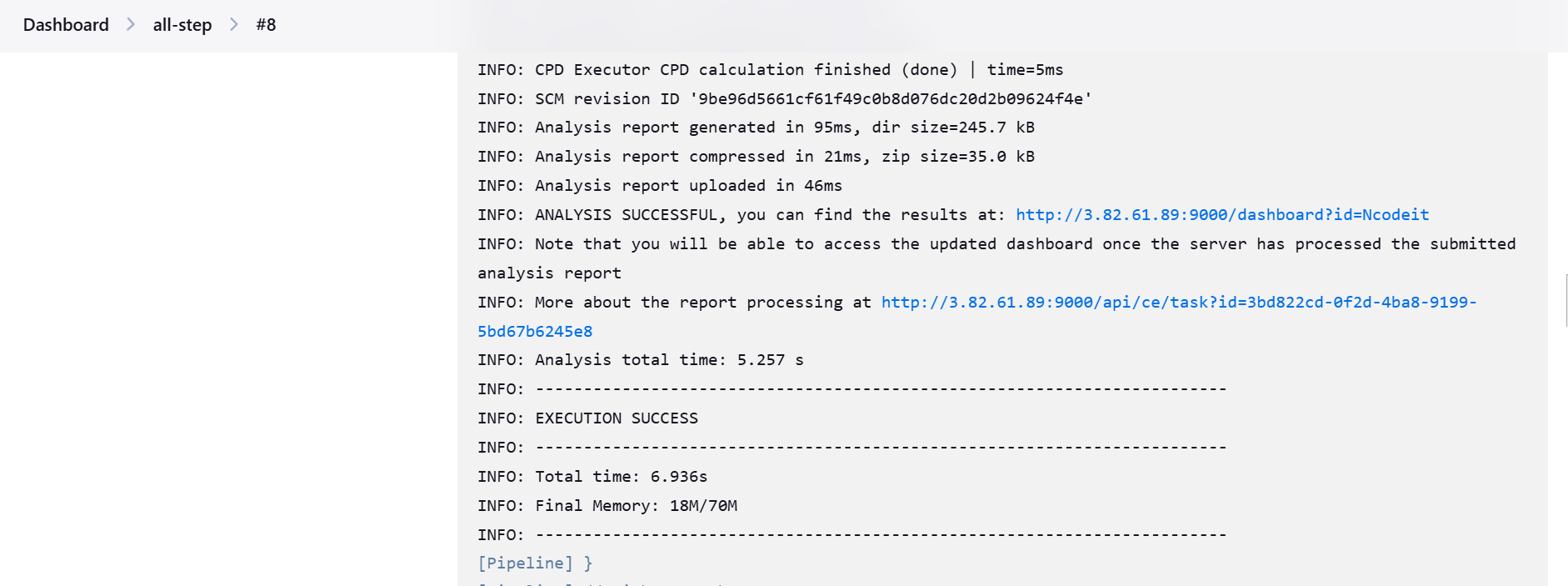
}

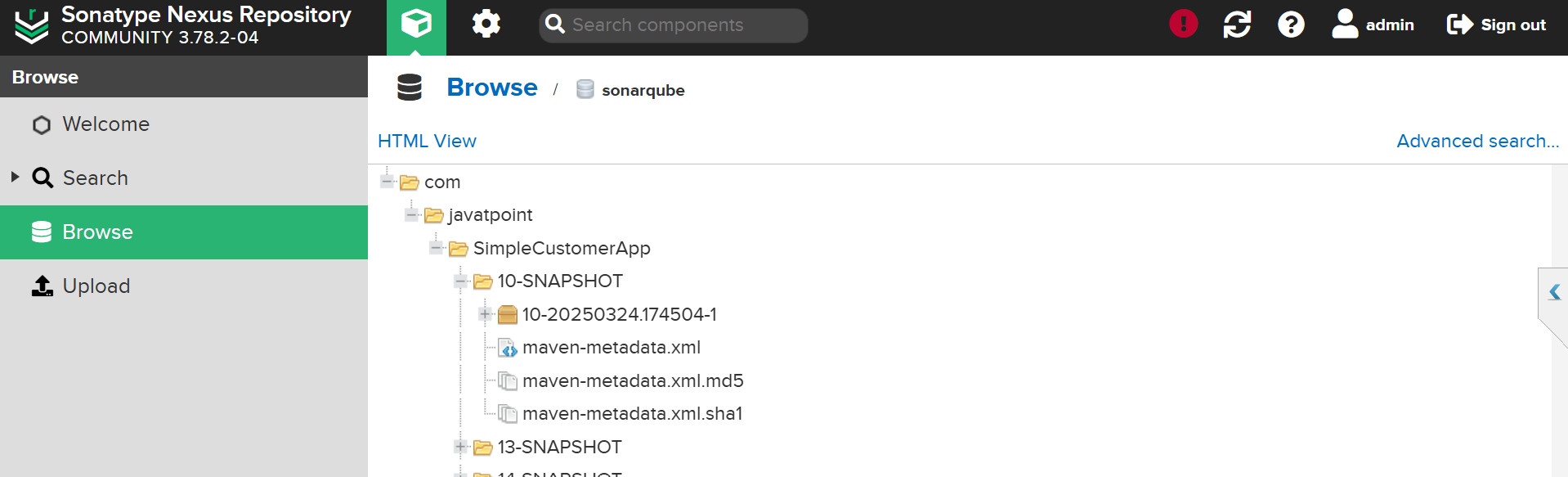
}

}

}

}





5) Slack Notification

post {

success {

slackSend(

channel: '#jenkins-integration',

message: ":white\_check\_mark: Build Successful! Ramesh & Raghu declarative pipeline testing :rocket:",

teamDomain: 'techiehorizon',

tokenCredentialId: 'raghu'

)

}

failure {

slackSend(

channel: '#jenkins-integration',

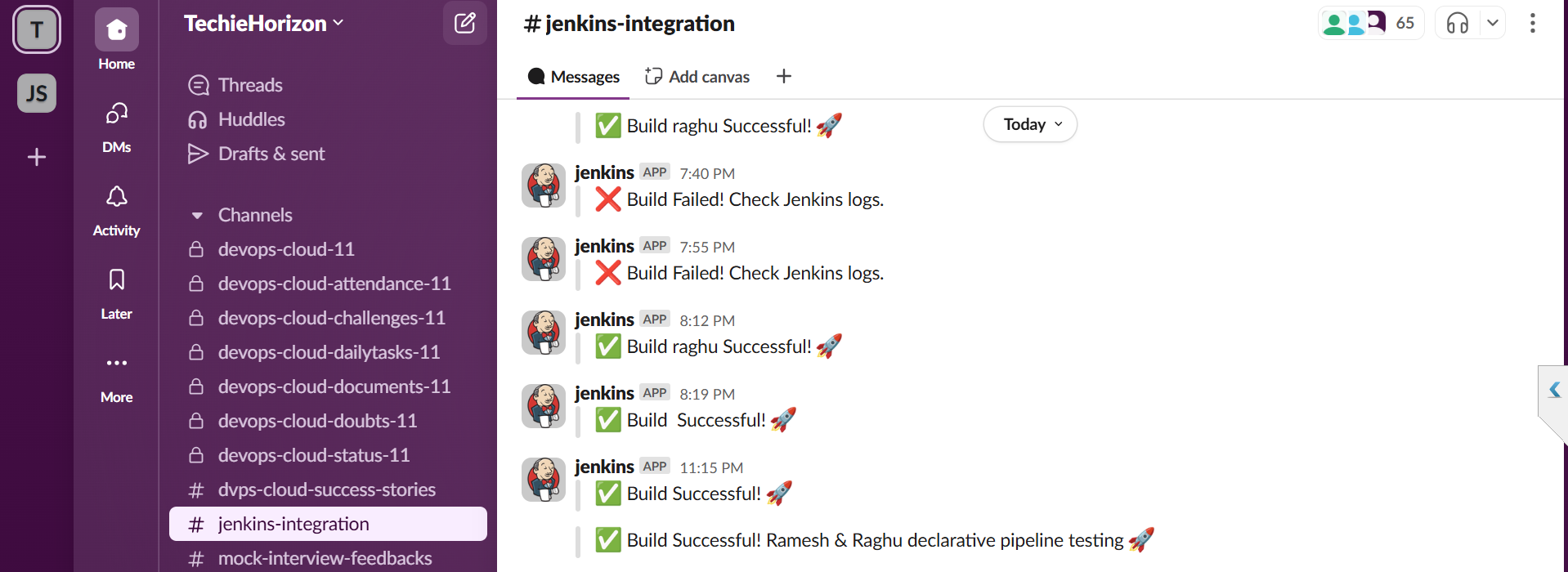
message: ":x: Build Failed! Check Jenkins logs.",

teamDomain: 'techiehorizon',

tokenCredentialId: 'raghu'

)

}



1. Deploy On tomcat

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 deployer:deployer \

http://3.94.89.70:8080/manager/text/deploy?path=/SimpleCustomerApp&update=true

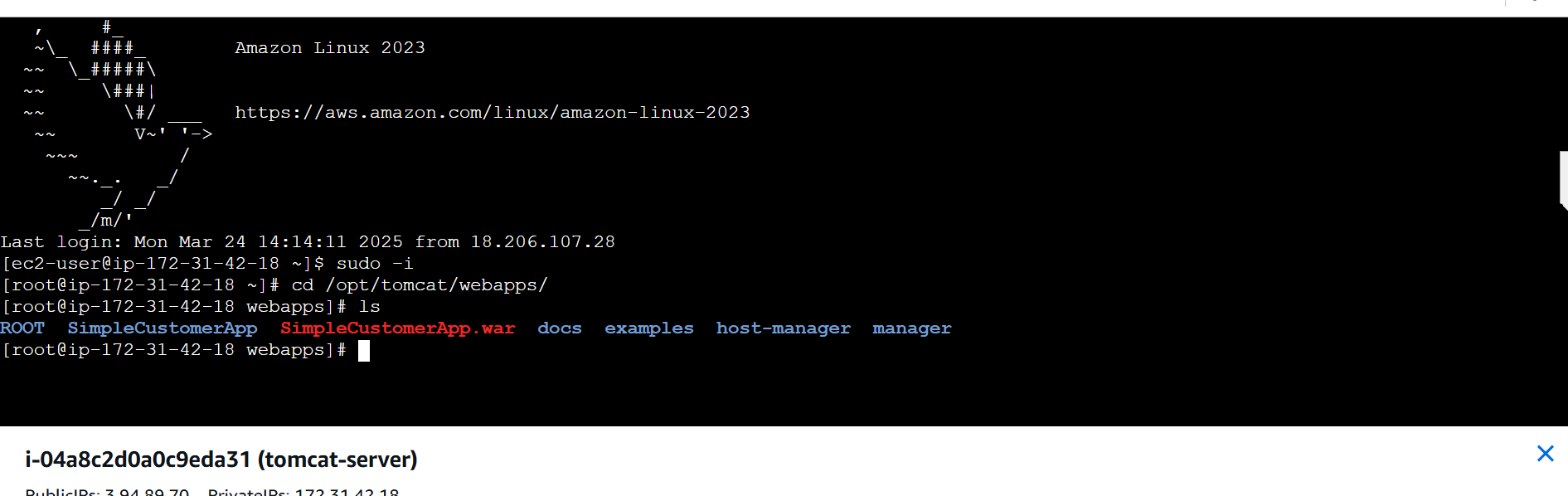
"""

}

}

}

}



1. Git install plugin in Jenkins GUI and also Jenkins master machine
2. Provision one server for sonarwube and install sonarqube via docker

Using below commands

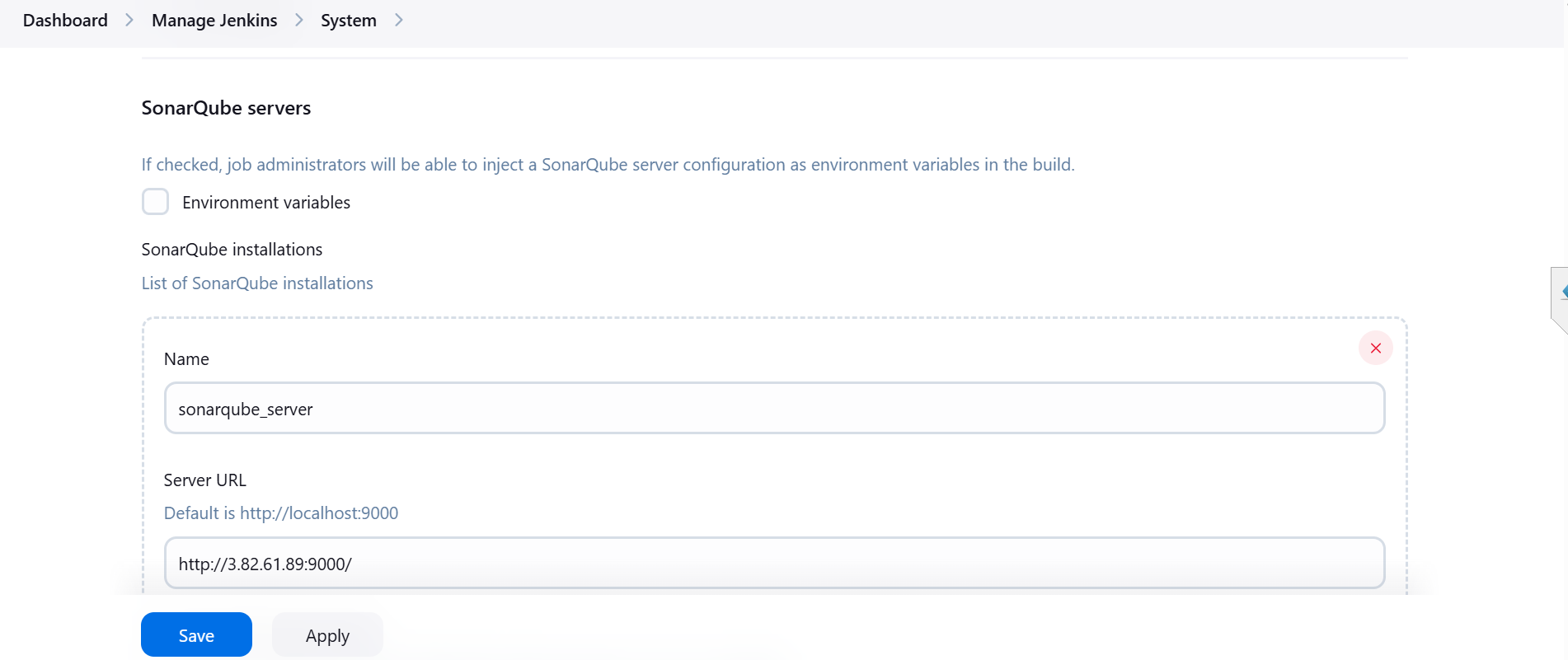
docker pull sonarqube – install sonarqube

By default, the server running within the container will listen on port 9000. You can expose the container port 9000 to the host port 9000 with the -p 9000:9000 argument to docker run, like the command below:

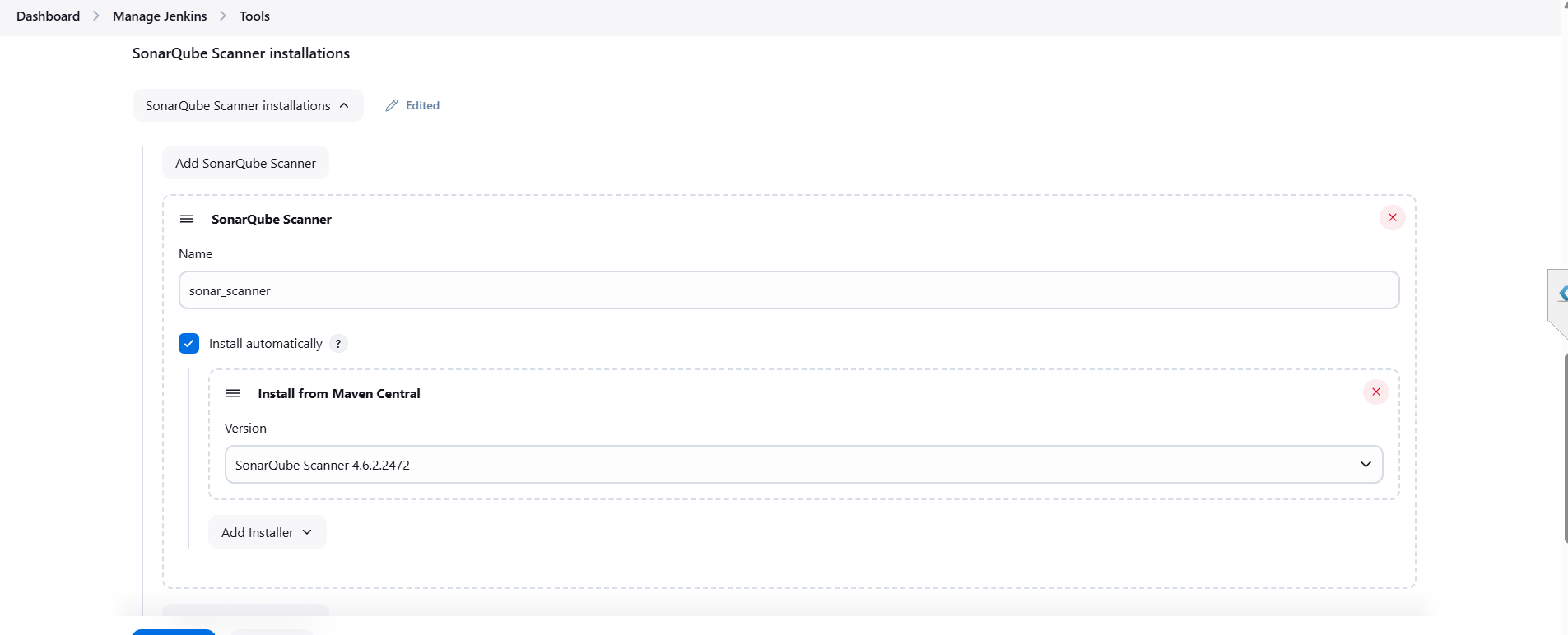
docker run --name sonarqube-custom -p 9000:9000 sonarqube:community

You can then browse to http://localhost:9000 or http://host-ip:9000 in your web browser to access the web interface.

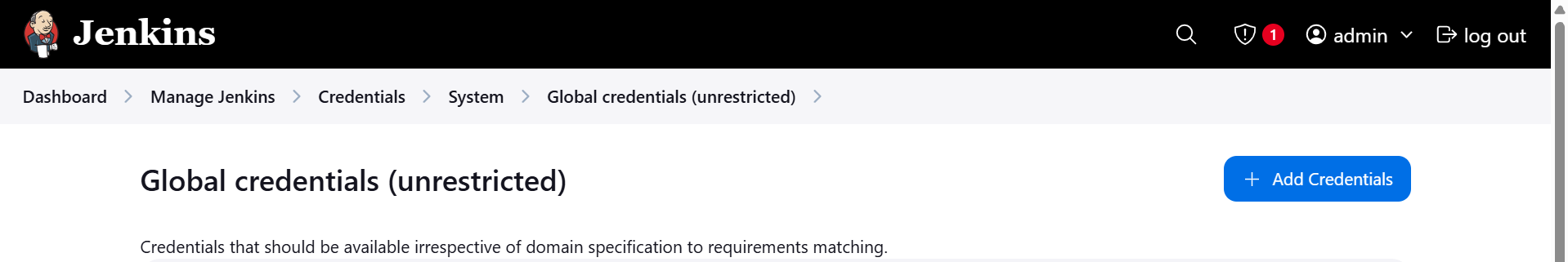
Configure part: first go to manage Jenkins system & paste the sonar server URL

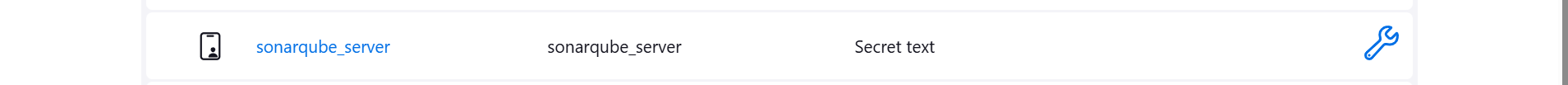


Goto Tools and sonar scanner installation



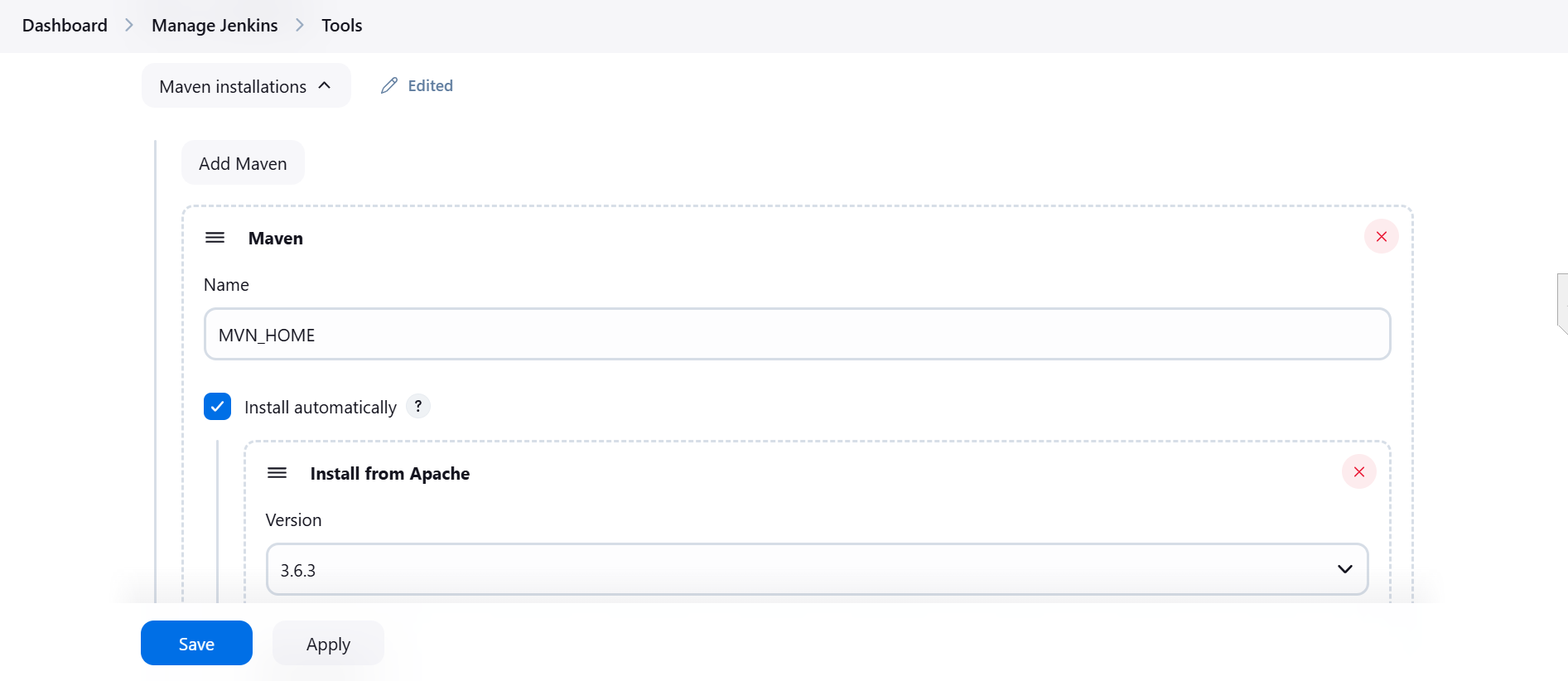
Go to credentials and add the sonar qube details 🡪 account and token generation

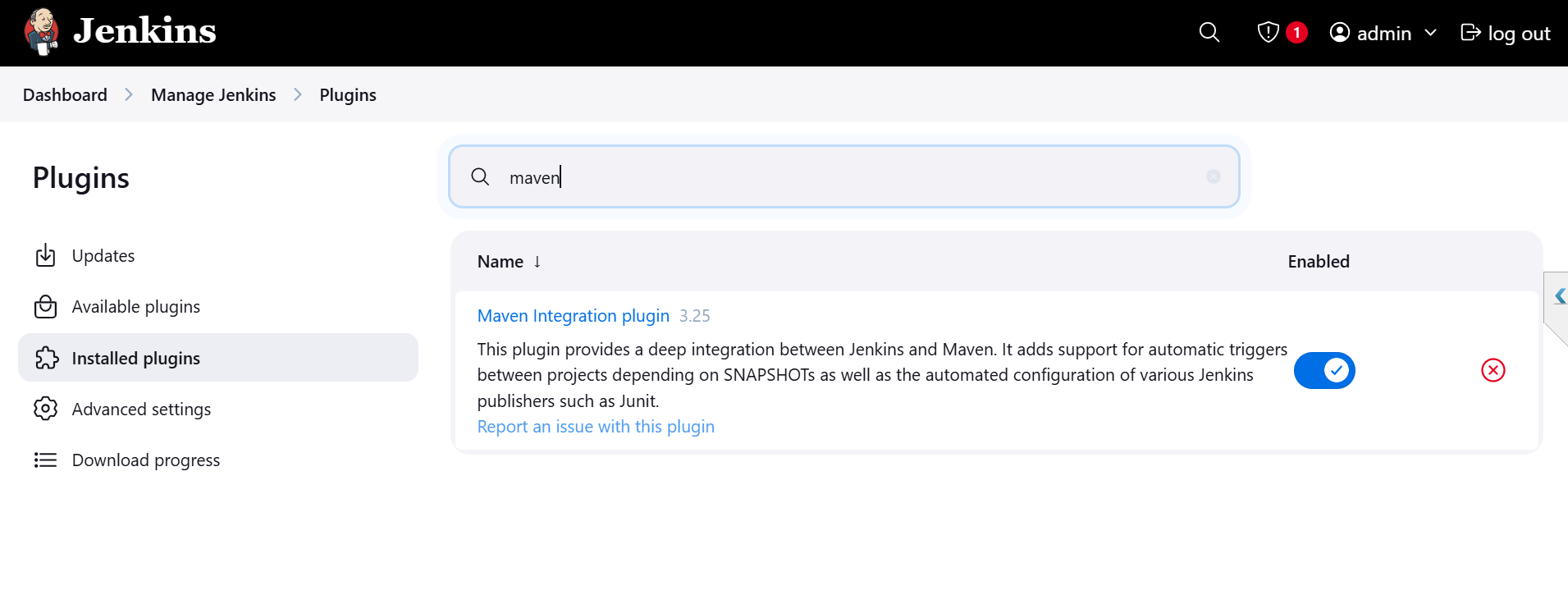




Maven configuration:

Install plugin





Launch a new instance with a name called nexus

First we need to install docker :

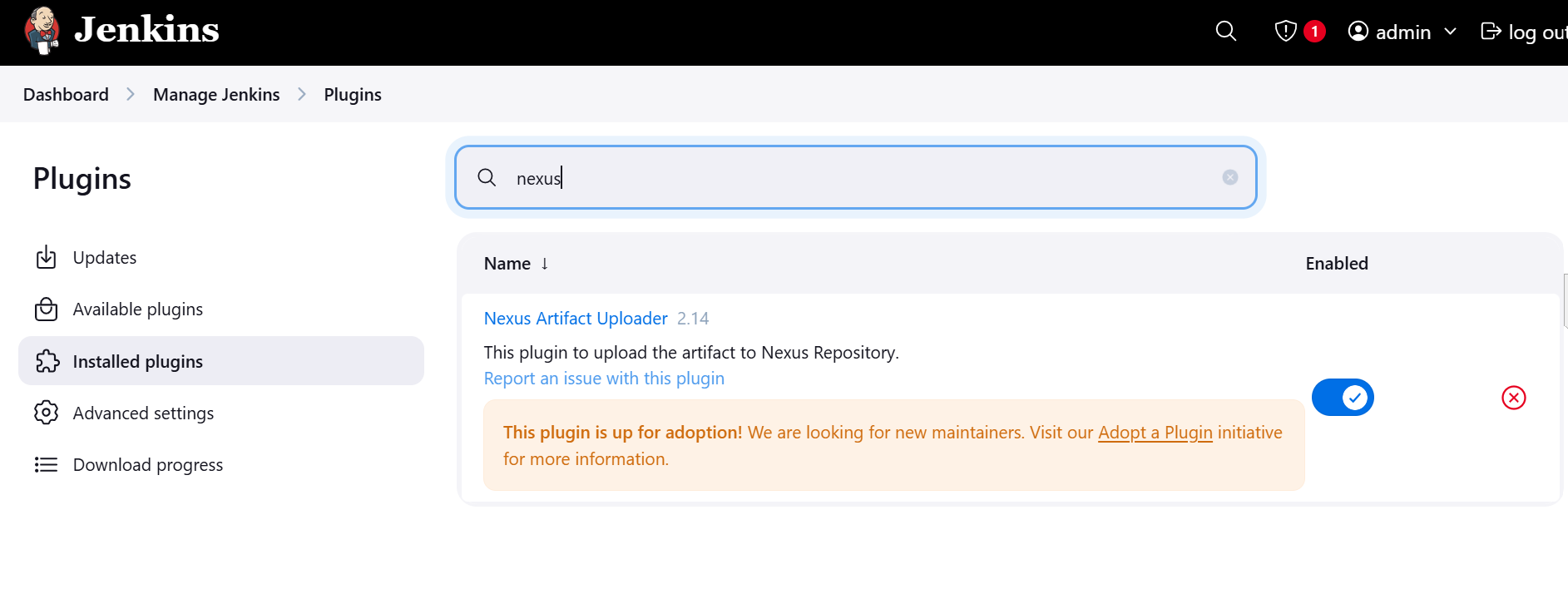
Yum install docekr

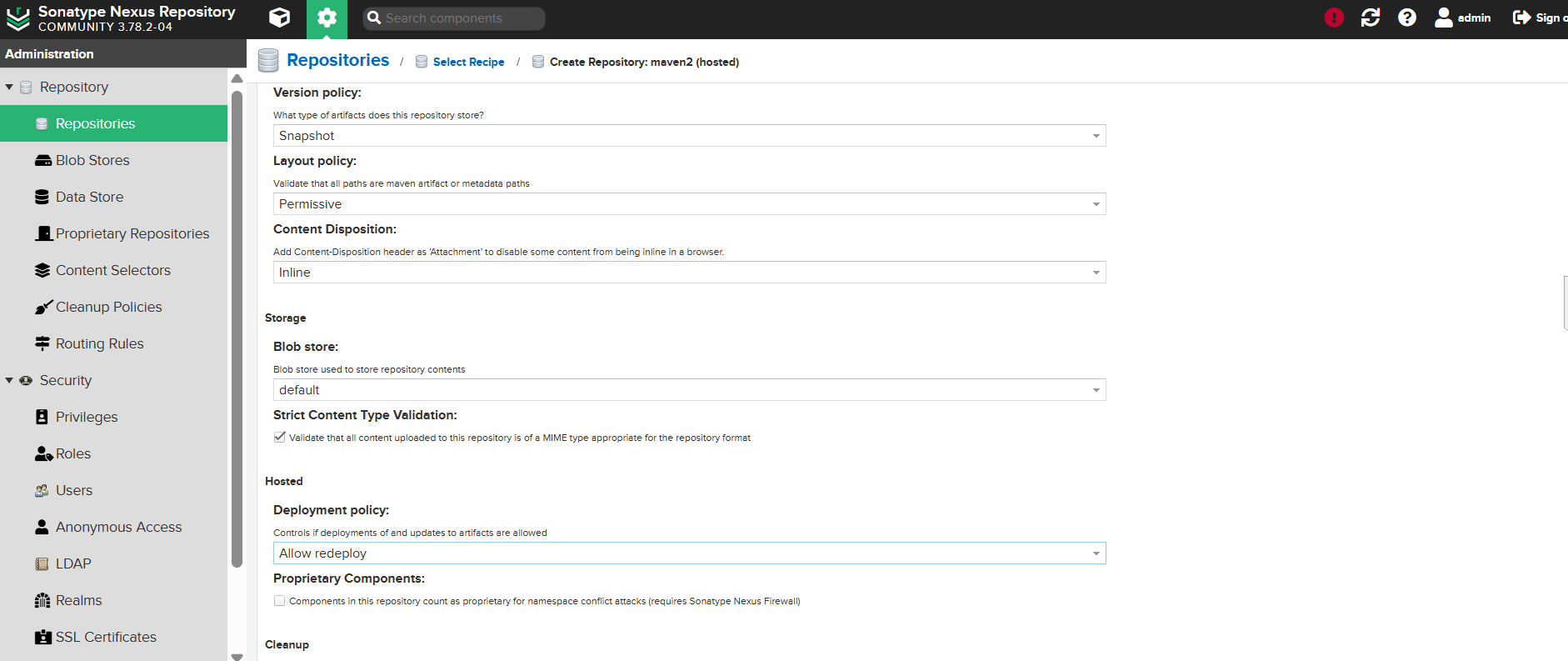
Systemctl start docker

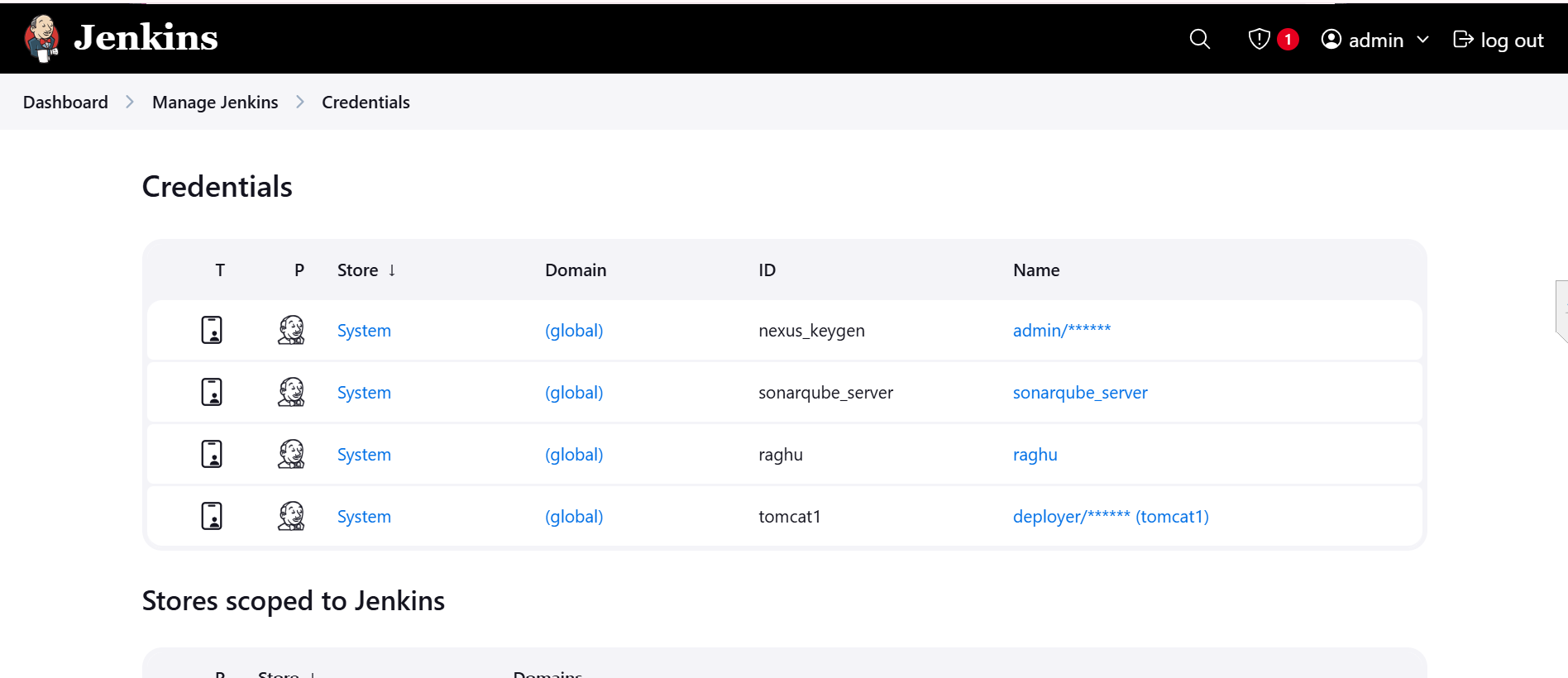
docker pull sonatype/nexus3 ----------installation

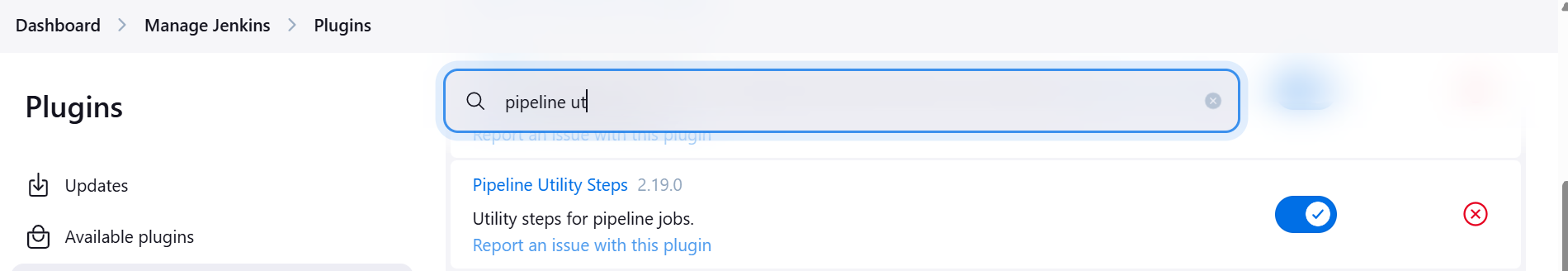
$ docker run -d -p 8081:8081 --name nexus sonatype/nexus3 ---- to start command

Docker ps to check the status of the nexus container

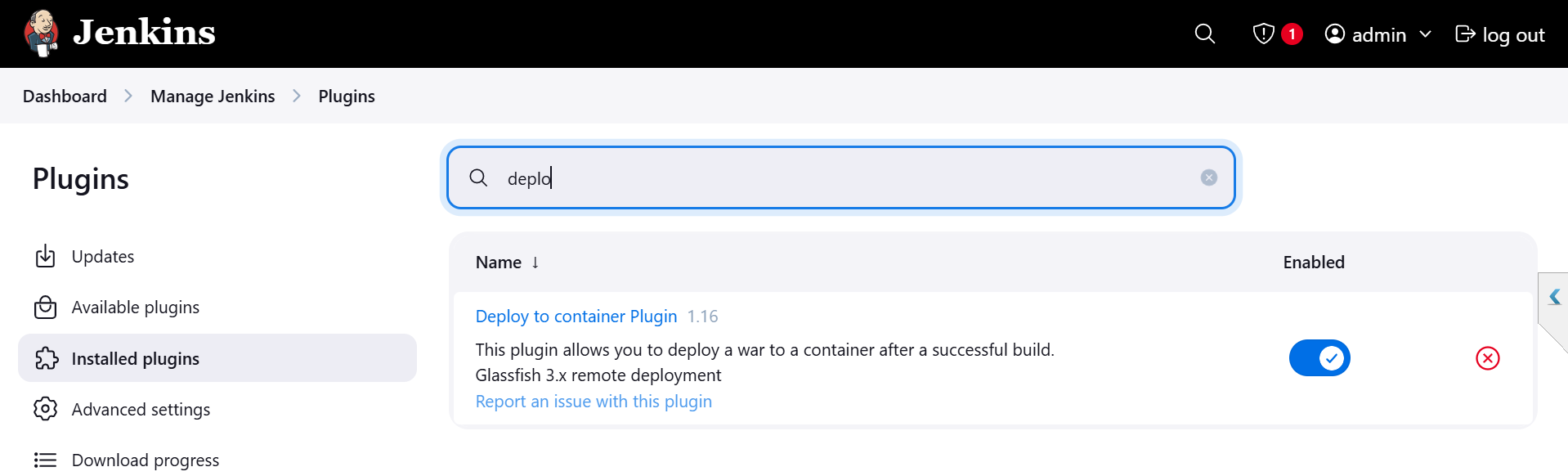




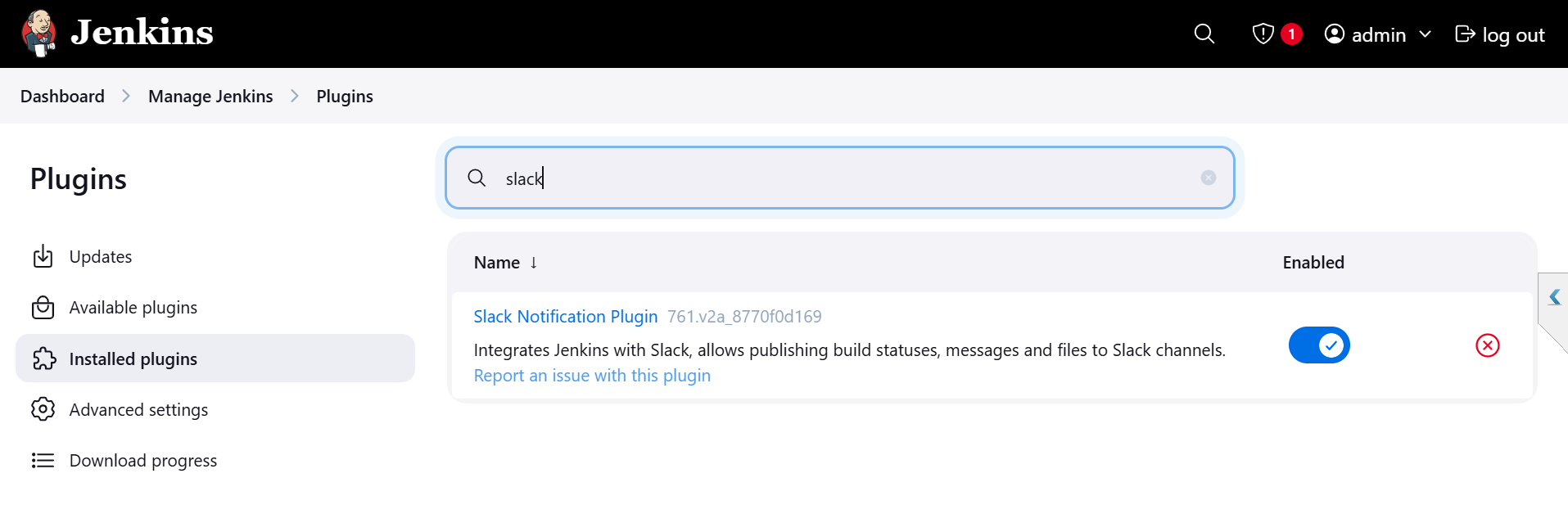


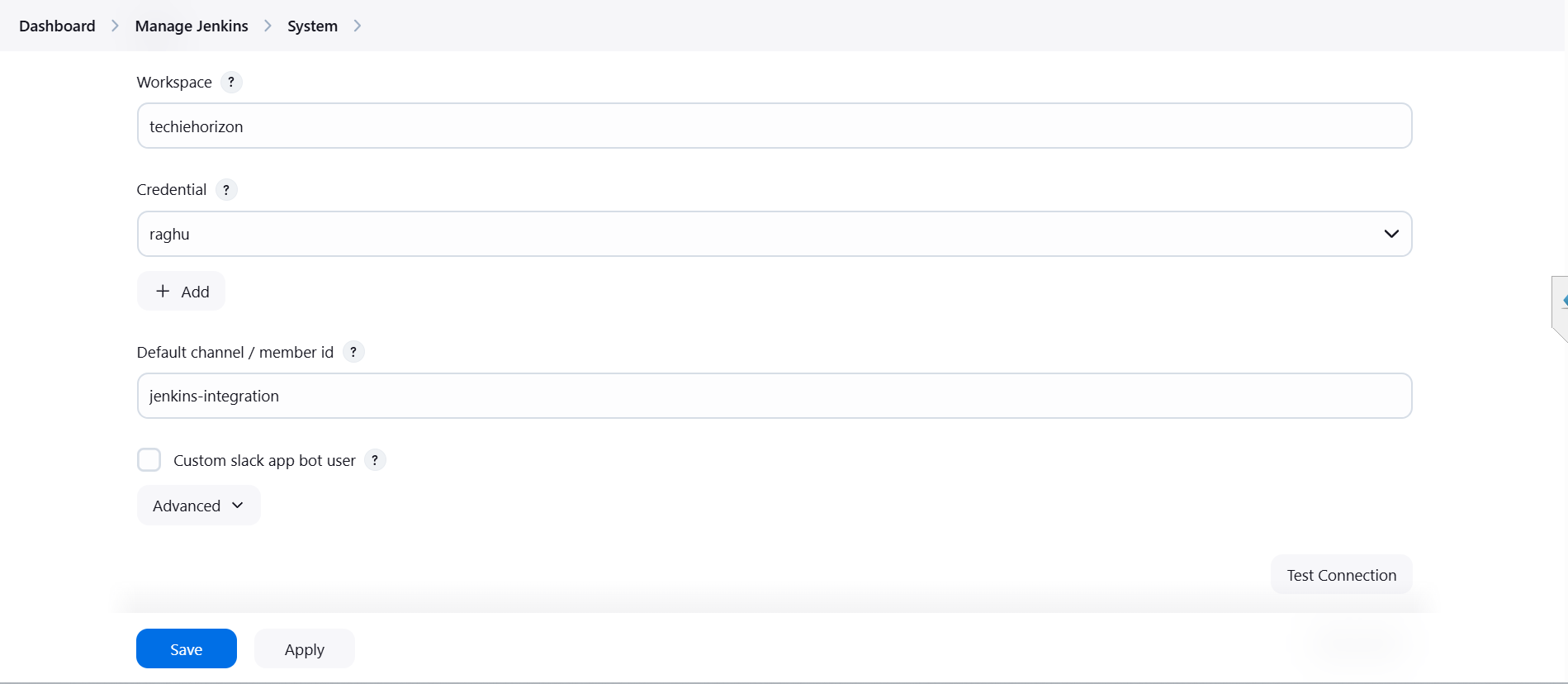


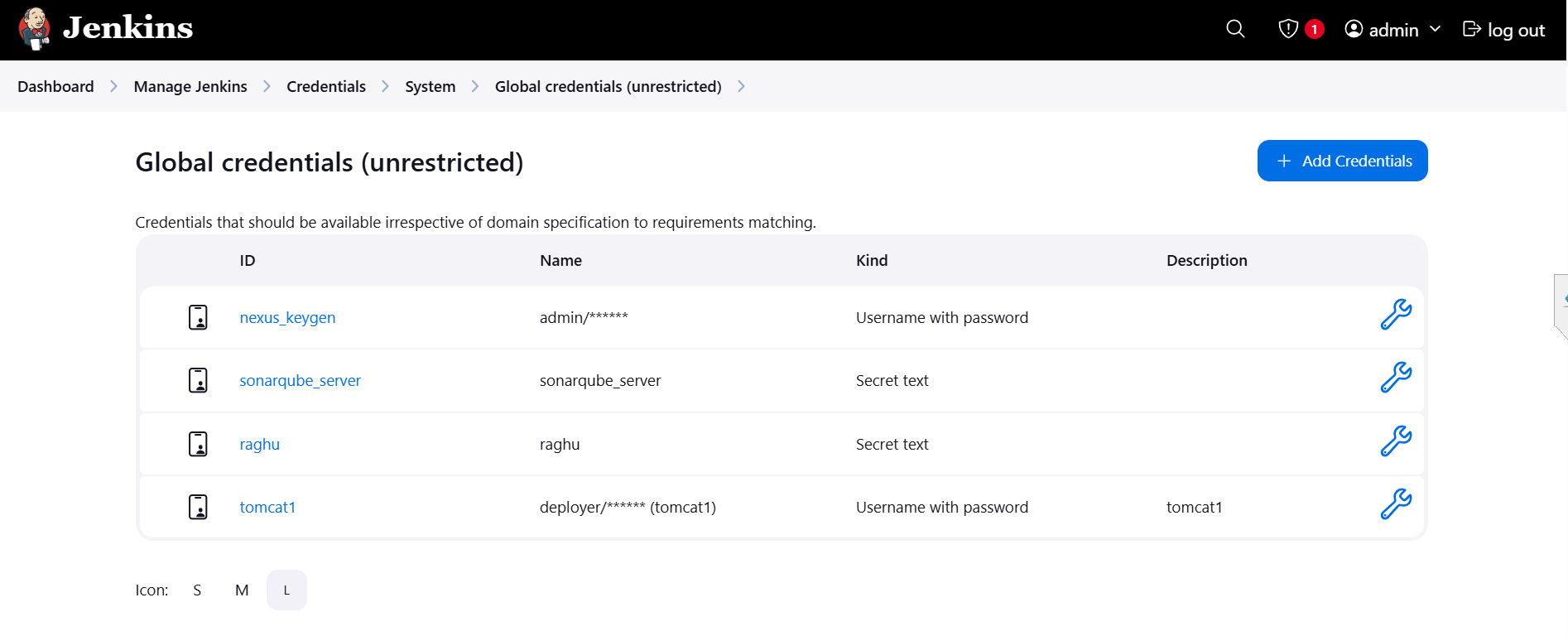
Tomcat installation



Slack notification:

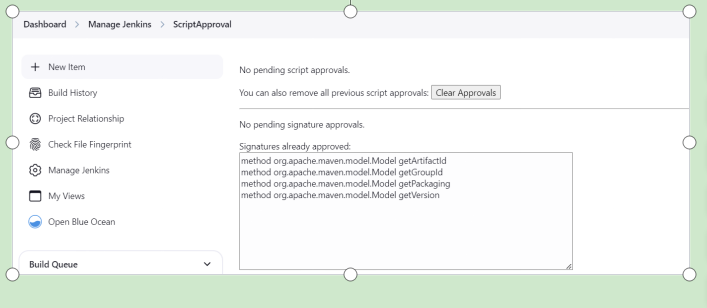






Errors:

1. Nexus while cerating a repository we are taken relase only but in code given the snapshot .



After approve only our code will be executed.