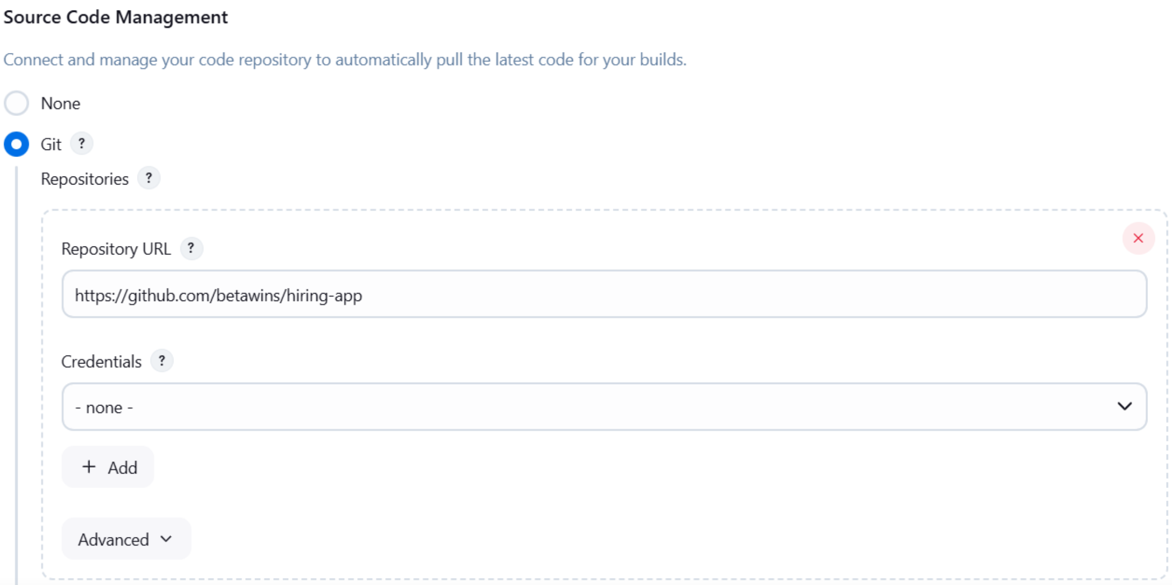
Jenkins 04:

1) Setup jenkins CICD pipeline using freestyle job using Docker containers using below code.

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

Stages:

1) Git Clone:



Source Code Management

Connect and manage your code repository to automatically pull the latest code for your builds.

○ None

● Git ?

Repositories ?

Repository URL ?

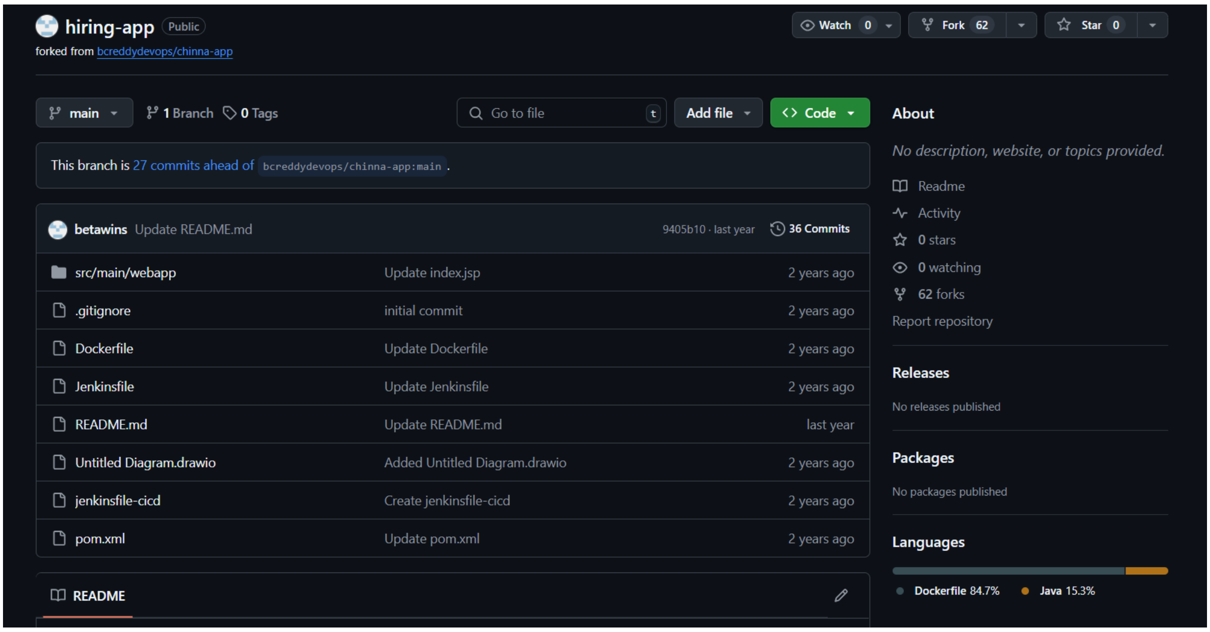
https://github.com/betawins/hiring-app

Credentials ?

- none -

+ Add

Advanced



hiring-app Public

forked from bcreddydevops/chinna-app

Watch 0 | Fork 62 | Star 0

main | 1 Branch | 0 Tags | Go to file | Add file | Code

This branch is 27 commits ahead of bcreddydevops/chinna-app:main .

betawins Update README.md — 9405b10 · last year — 36 Commits

src/main/webapp	Update index.jsp	2 years ago
.gitignore	initial commit	2 years ago
Dockerfile	Update Dockerfile	2 years ago
Jenkinsfile	Update Jenkinsfile	2 years ago
README.md	Update README.md	last year
Untitled Diagram.drawio	Added Untitled Diagram.drawio	2 years ago
jenkinsfile-cicd	Create jenkinsfile-cicd	2 years ago
pom.xml	Update pom.xml	2 years ago

README

About

No description, website, or topics provided.

Readme

Activity

0 stars

0 watching

62 forks

Report repository

Releases

No releases published

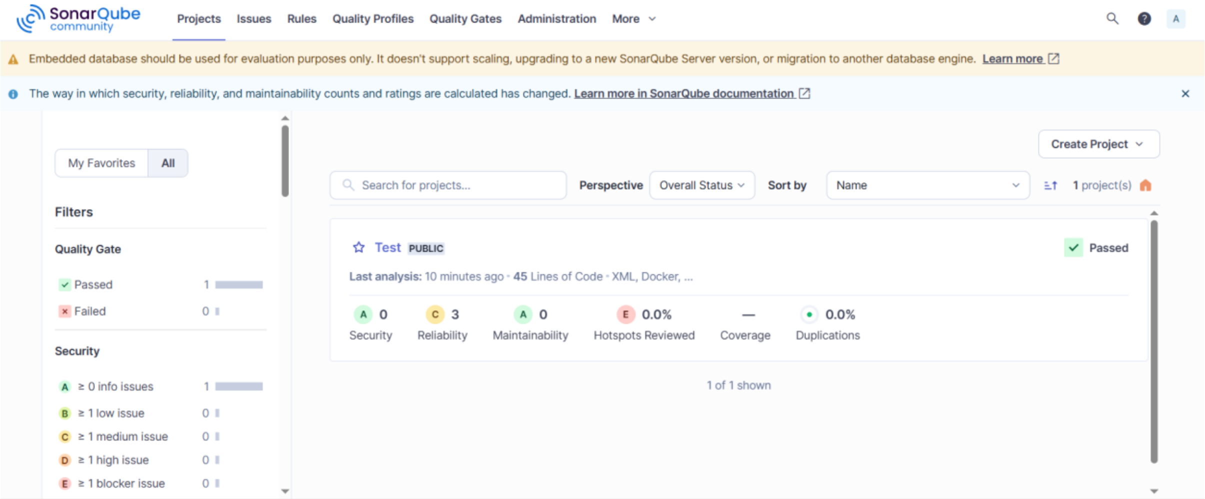
Packages

No packages published

Languages

Dockerfile 84.7% Java 15.3%

2) Sonarqube Integration:



Embedded database should be used for evaluation purposes only. It doesn't support scaling, upgrading to a new SonarQube Server version, or migration to another database engine. Learn more

The way in which security, reliability, and maintainability counts and ratings are calculated has changed. Learn more in SonarQube documentation

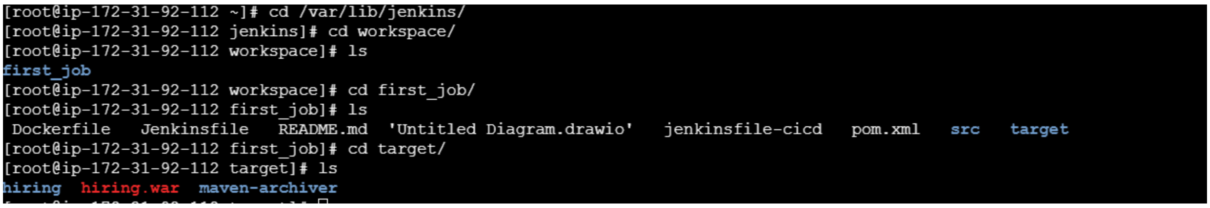
Test PUBLIC — Passed

Last analysis: 10 minutes ago · 45 Lines of Code · XML, Docker, ...

A 0	C 3	A 0	E 0.0%	—	● 0.0%
Security	Reliability	Maintainability	Hotspots Reviewed	Coverage	Duplications

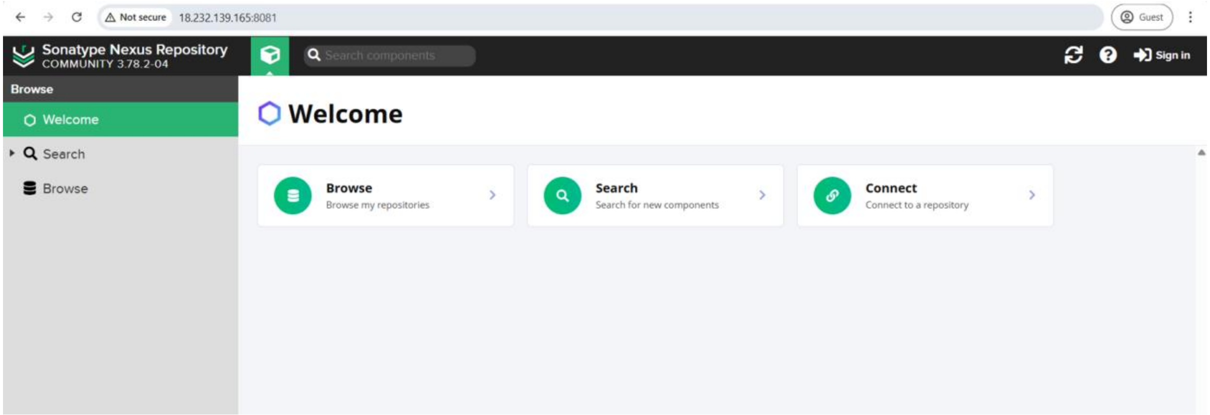
1 of 1 shown

3) Maven Compilation:

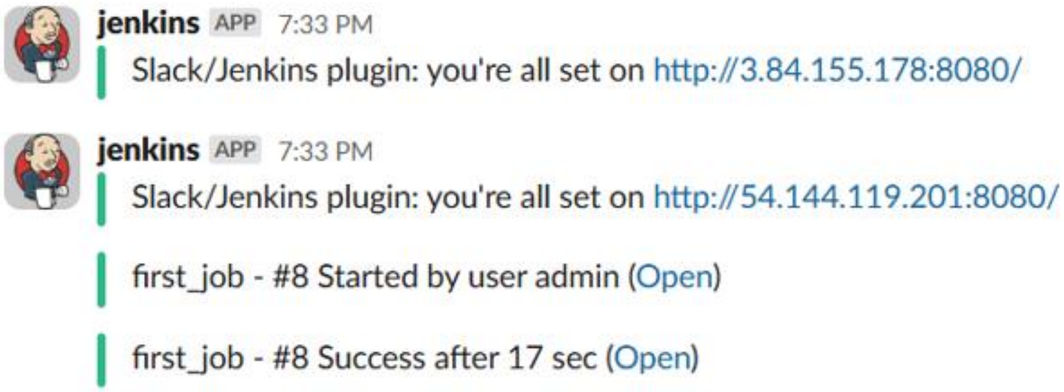


```
[root@ip-172-31-92-112 ~]# cd /var/lib/jenkins/
[root@ip-172-31-92-112 jenkins]# cd workspace/
[root@ip-172-31-92-112 workspace]# ls
first_job
[root@ip-172-31-92-112 workspace]# cd first_job/
[root@ip-172-31-92-112 first_job]# ls
 Dockerfile   Jenkinsfile   README.md  'Untitled Diagram.drawio'   jenkinsfile-cicd   pom.xml   src   target
[root@ip-172-31-92-112 first_job]# cd target/
[root@ip-172-31-92-112 target]# ls
hiring  hiring.war  maven-archiver
```

4) Nexus Artifactory



5) Slack Notification



jenkins APP 7:33 PM

Slack/Jenkins plugin: you're all set on http://3.84.155.178:8080/

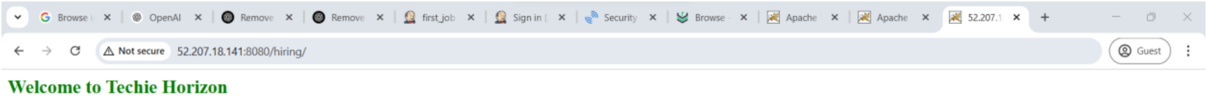
jenkins APP 7:33 PM

Slack/Jenkins plugin: you're all set on http://54.144.119.201:8080/

first_job - #8 Started by user admin (Open)

first_job - #8 Success after 17 sec (Open)

6) Deploy On tomcat



52.207.18.141:8080/hiring/

Welcome to Techie Horizon

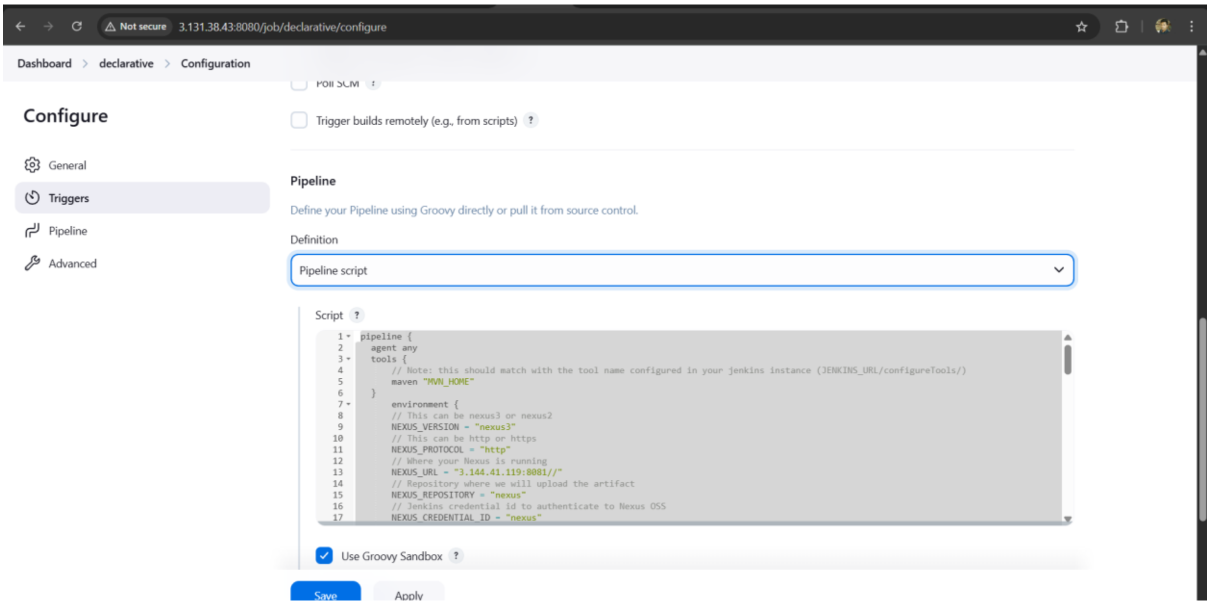
2) Setup a jenkins CICD pipeline using Declarative pipeline using feature-1.1 branch. https://github.com/betawins/sabear_simplecutomerapp/tree/feature-1.1

→Declarative Pipeline:

```
   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 = "3.144.41.119:8081//"
    // Repository where we will upload the artifact
    NEXUS_REPOSITORY = "nexus"
    // Jenkins credential id to authenticate to Nexus OSS
    NEXUS_CREDENTIAL_ID = "nexus"
   SCANNER_HOME = tool 'sonarqube_server'
}
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') {
                                    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: "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 {
         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://http://18.222.112.27/:8080/manager/text/deploy?path=/SimpleCustomerApp&update=true
         """
       }
     }
    }
  }
```

}



3.131.38.43:8080/job/declarative/configure

Dashboard > declarative > Configuration

Configure

Poll SCM

□ Trigger builds remotely (e.g., from scripts)

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition: Pipeline script

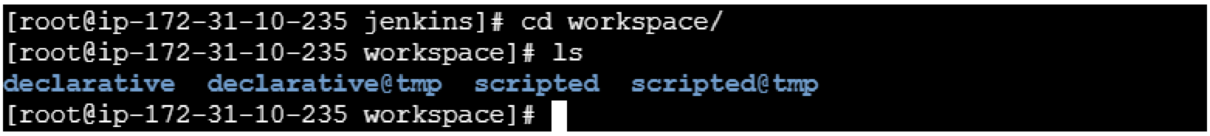
Script

```
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 = "3.144.41.119:8081//"
      // Repository where we will upload the artifact
      NEXUS_REPOSITORY = "nexus"
      // Jenkins credential id to authenticate to Nexus OSS
      NEXUS_CREDENTIAL_ID = "nexus"
```

☑ Use Groovy Sandbox

stages:

1) Git Clone

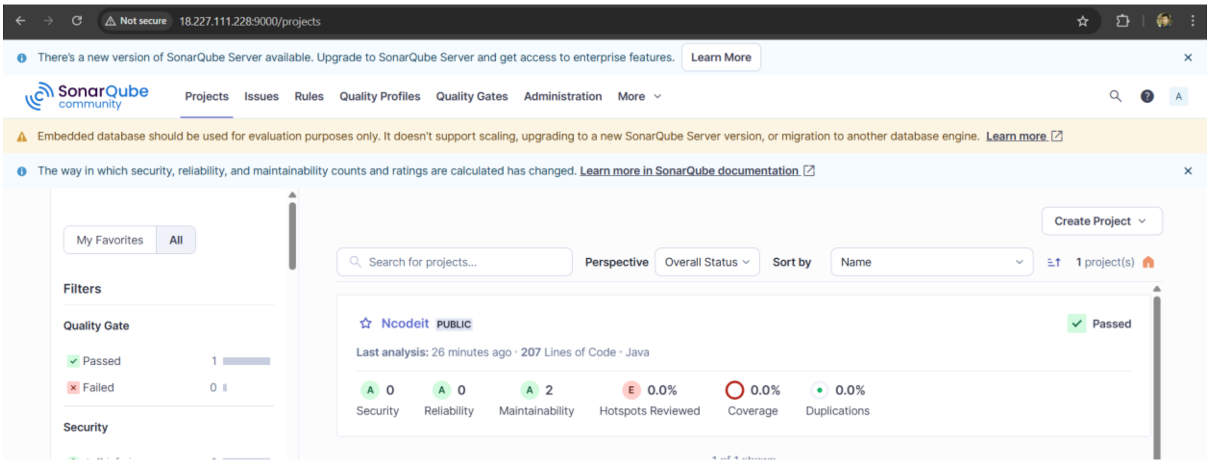


```
[root@ip-172-31-10-235 jenkins]# cd workspace/
[root@ip-172-31-10-235 workspace]# ls
declarative  declarative@tmp  scripted  scripted@tmp
[root@ip-172-31-10-235 workspace]#
```

i-0240b88b38eb8d8d2 (jenkins-master)

PublicIPs: 3.131.38.43   PrivateIPs: 172.31.10.235

2) Sonarqube Integration



18.227.111.228:9000/projects

There's a new version of SonarQube Server available. Upgrade to SonarQube Server and get access to enterprise features. Learn More

Embedded database should be used for evaluation purposes only. It doesn't support scaling, upgrading to a new SonarQube Server version, or migration to another database engine. Learn more

The way in which security, reliability, and maintainability counts and ratings are calculated has changed. Learn more in SonarQube documentation

Perspective: Overall Status   Sort by: Name   1 project(s)

Filters

Quality Gate

Passed	1
Failed	0

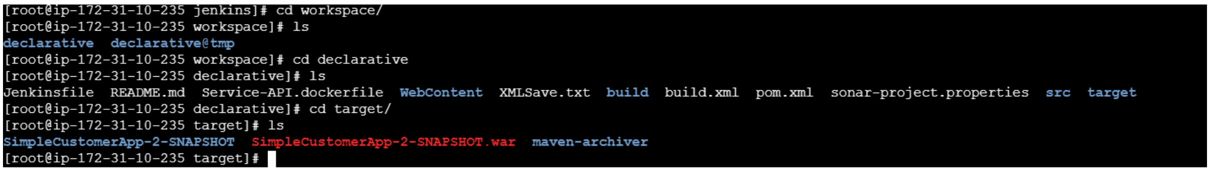
Security

Ncodeit PUBLIC — Passed

Last analysis: 26 minutes ago · 207 Lines of Code · Java

A 0	A 0	A 2	E 0.0%	0.0%	0.0%
Security	Reliability	Maintainability	Hotspots Reviewed	Coverage	Duplications

3) Maven Compilation

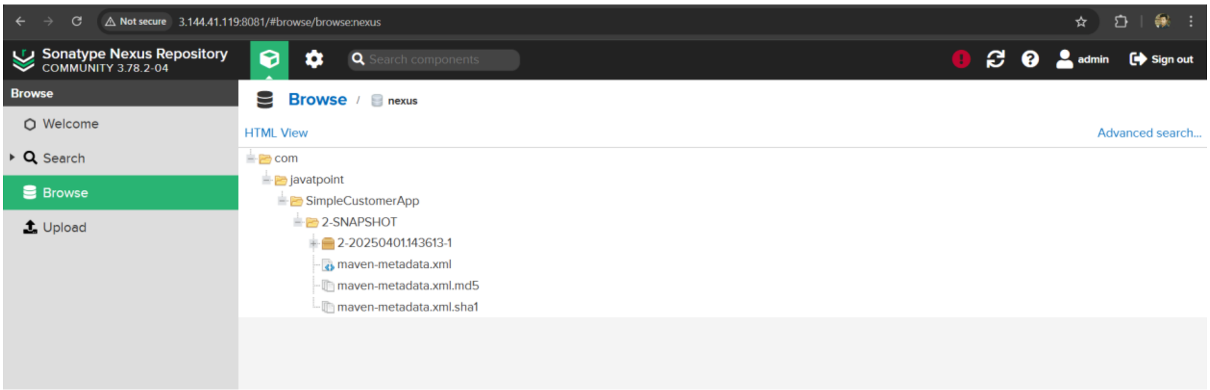


```
[root@ip-172-31-10-235 jenkins]# cd workspace/
[root@ip-172-31-10-235 workspace]# ls
declarative  declarative@tmp
[root@ip-172-31-10-235 workspace]# cd declarative
[root@ip-172-31-10-235 declarative]# ls
Jenkinsfile  README.md  Service-API.dockerfile  WebContent  XMLSave.txt  build  build.xml  pom.xml  sonar-project.properties  src  target
[root@ip-172-31-10-235 declarative]# cd target/
[root@ip-172-31-10-235 target]# ls
SimpleCustomerApp-2-SNAPSHOT  SimpleCustomerApp-2-SNAPSHOT.war  maven-archiver
[root@ip-172-31-10-235 target]# 
```

i-0240b88b38eb8d8d2 (jenkins-master)

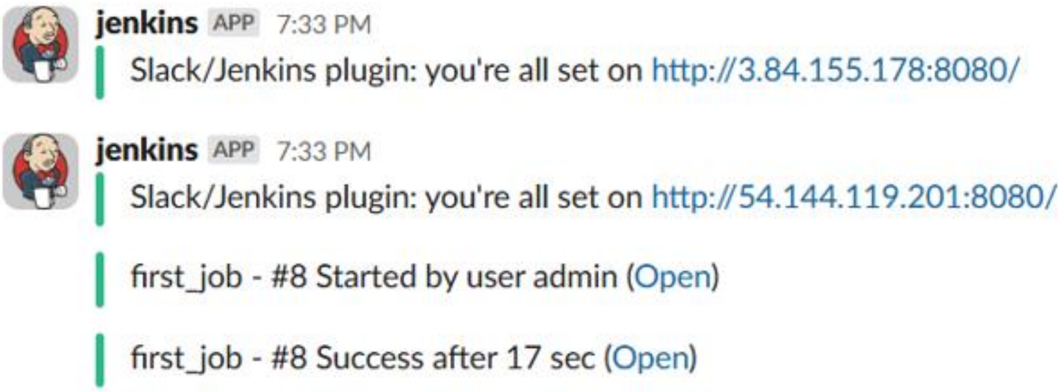
PublicIPs: 3.131.38.43 PrivateIPs: 172.31.10.235

4) Nexus Artifactory



```
com
  javatpoint
    SimpleCustomerApp
      2-SNAPSHOT
        2-20250401.143613-1
        maven-metadata.xml
        maven-metadata.xml.md5
        maven-metadata.xml.sha1
```

5)Slack Notification



jenkins APP 7:33 PM

Slack/Jenkins plugin: you're all set on http://3.84.155.178:8080/

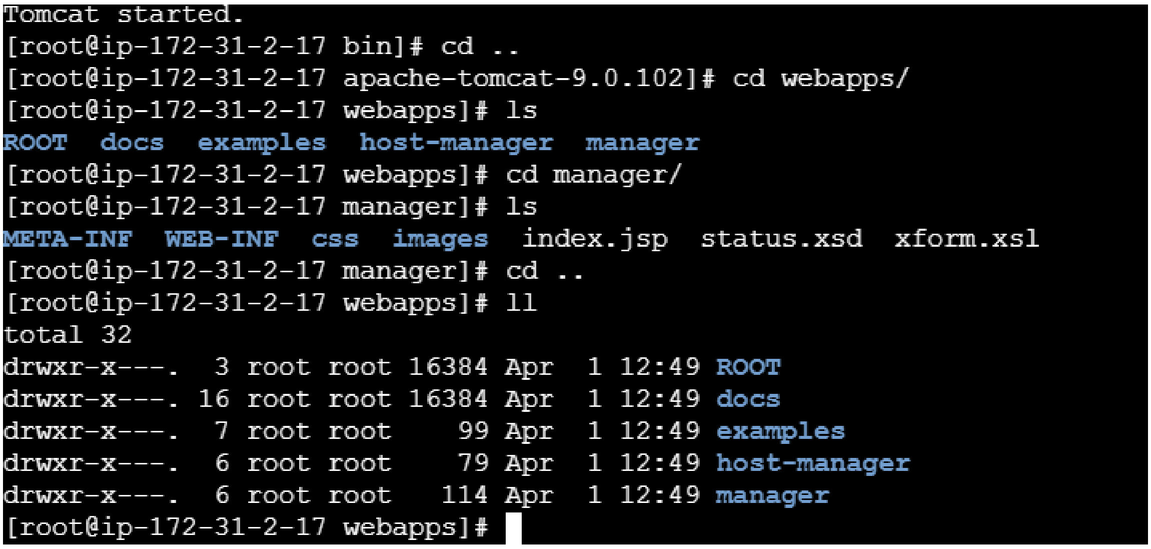
jenkins APP 7:33 PM

Slack/Jenkins plugin: you're all set on http://54.144.119.201:8080/

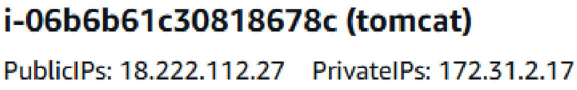
first_job - #8 Started by user admin (Open)

first_job - #8 Success after 17 sec (Open)

6) Deploy On tomcat

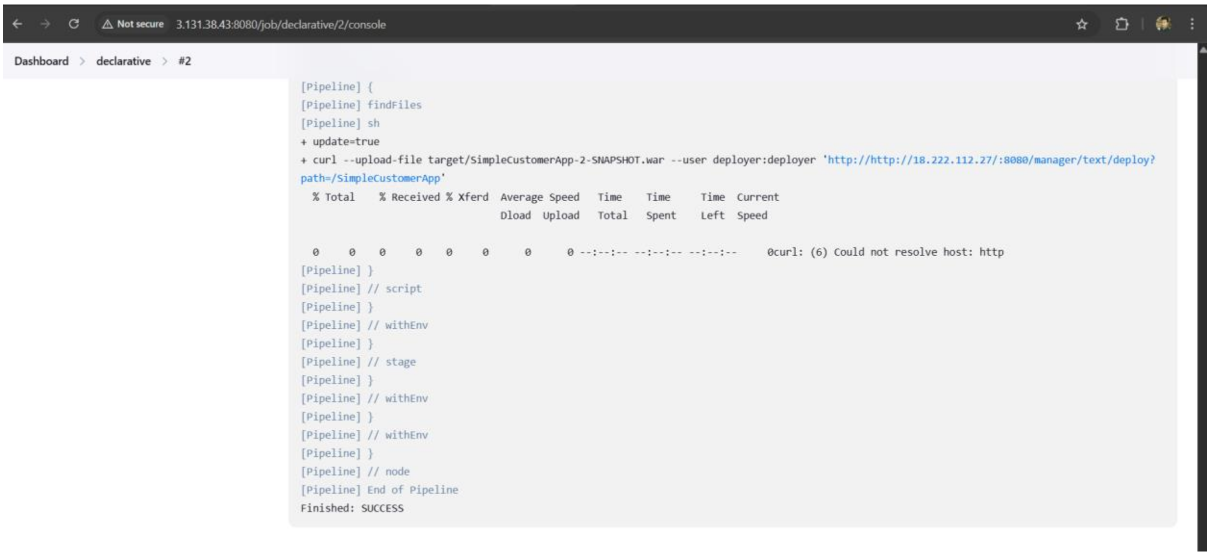


```
Tomcat started.
[root@ip-172-31-2-17 bin]# cd ..
[root@ip-172-31-2-17 apache-tomcat-9.0.102]# cd webapps/
[root@ip-172-31-2-17 webapps]# ls
ROOT  docs  examples  host-manager  manager
[root@ip-172-31-2-17 webapps]# cd manager/
[root@ip-172-31-2-17 manager]# ls
META-INF  WEB-INF  css  images  index.jsp  status.xsd  xform.xsl
[root@ip-172-31-2-17 manager]# cd ..
[root@ip-172-31-2-17 webapps]# ll
total 32
drwxr-x---.  3 root root 16384 Apr  1 12:49 ROOT
drwxr-x---. 16 root root 16384 Apr  1 12:49 docs
drwxr-x---.  7 root root    99 Apr  1 12:49 examples
drwxr-x---.  6 root root    79 Apr  1 12:49 host-manager
drwxr-x---.  6 root root   114 Apr  1 12:49 manager
[root@ip-172-31-2-17 webapps]# 
```



i-06b6b61c30818678c (tomcat)

PublicIPs: 18.222.112.27   PrivateIPs: 172.31.2.17



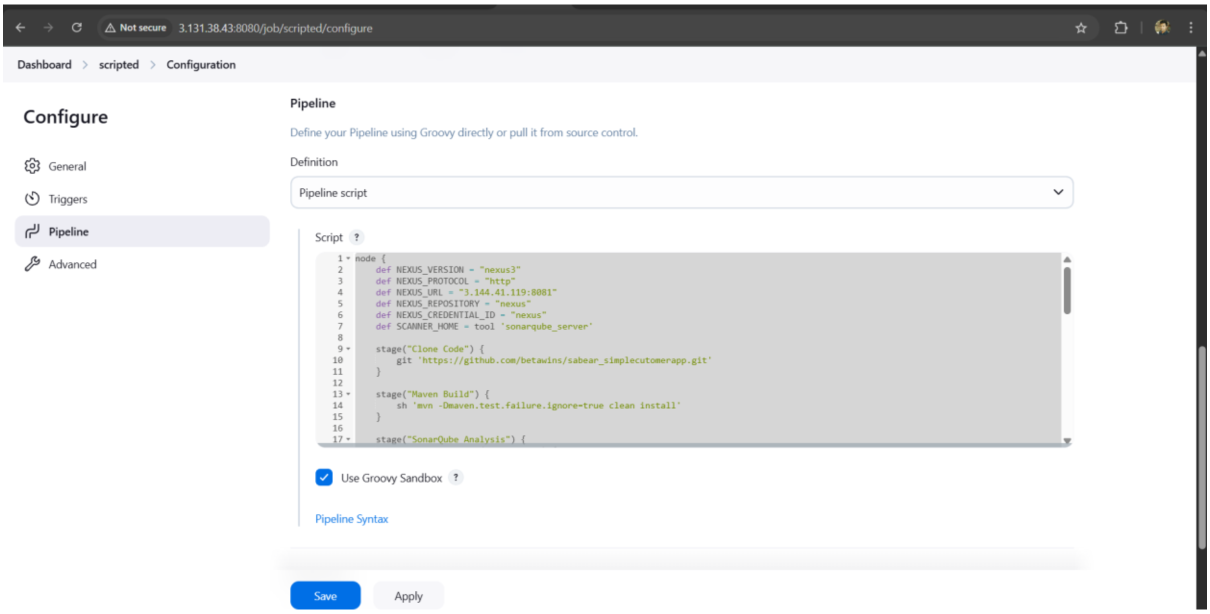
Not secure
3.131.38.43:8080/job/declarative/2/console
Dashboard > declarative > #2

```
[Pipeline] {
[Pipeline] findFiles
[Pipeline] sh
+ update=true
+ curl --upload-file target/SimpleCustomerApp-2-SNAPSHOT.war --user deployer:deployer 'http://http://18.222.112.27/:8080/manager/text/deploy?
path=/SimpleCustomerApp'
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed

  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0curl: (6) Could not resolve host: http
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

3)Setup a jenkins CICD pipeline using Scripted pipeline using feature-1.1 branch. https://github.com/betawins/sabear_simplecutomerapp/tree/feature-1.1

→Scripted pipeline:

```
node {
   def NEXUS_VERSION = "nexus3"
   def NEXUS_PROTOCOL = "http"
   def NEXUS_URL = "3.144.41.119:8081"
   def NEXUS_REPOSITORY = "nexus"
   def NEXUS_CREDENTIAL_ID = "nexus"
   def SCANNER_HOME = tool 'sonarqube_server'

   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') {
           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: 'tomcat1', path: '', url:
   'http://18.222.112.27/')],
         contextPath: 'SimpleCustomerApp',
         war: "target/SimpleCustomerApp-${BUILD_NUMBER}-SNAPSHOT.war"
       }
     }


   }
```



Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition

Pipeline script

Script ?

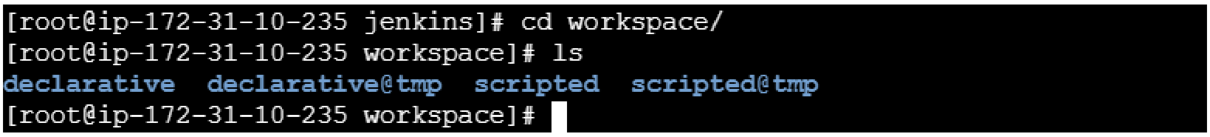
```
node {
    def NEXUS_VERSION = "nexus3"
    def NEXUS_PROTOCOL = "http"
    def NEXUS_URL = "3.144.41.119:8081"
    def NEXUS_REPOSITORY = "nexus"
    def NEXUS_CREDENTIAL_ID = "nexus"
    def SCANNER_HOME = tool 'sonarqube_server'

    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") {
```

☑ Use Groovy Sandbox ?

Pipeline Syntax

stages:

1) Git Clone

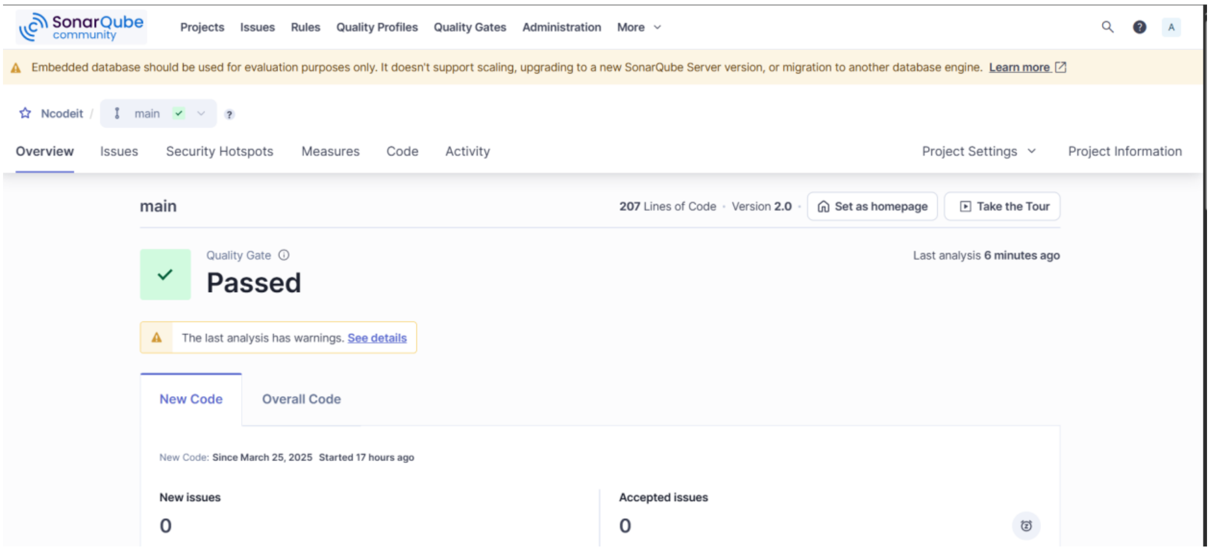


```
[root@ip-172-31-10-235 jenkins]# cd workspace/
[root@ip-172-31-10-235 workspace]# ls
declarative  declarative@tmp  scripted  scripted@tmp
[root@ip-172-31-10-235 workspace]#
```

i-0240b88b38eb8d8d2 (jenkins-master)

PublicIPs: 3.131.38.43   PrivateIPs: 172.31.10.235

2) Sonarqube Integration



Embedded database should be used for evaluation purposes only. It doesn't support scaling, upgrading to a new SonarQube Server version, or migration to another database engine. Learn more

main

207 Lines of Code · Version 2.0

Quality Gate

Passed

Last analysis 6 minutes ago

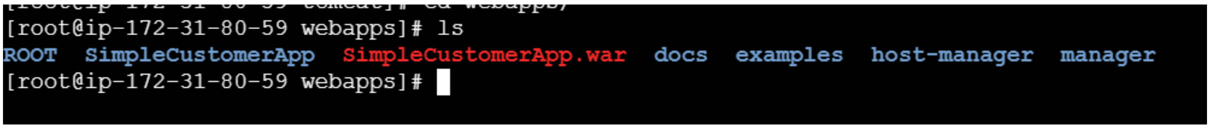
The last analysis has warnings. See details

New Code: Since March 25, 2025 Started 17 hours ago

New issues
0

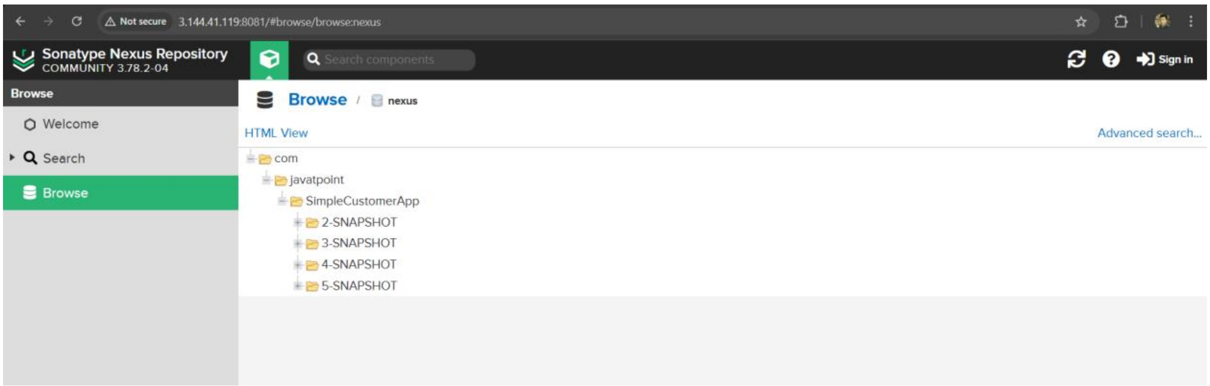
Accepted issues
0

3) Maven Compilation



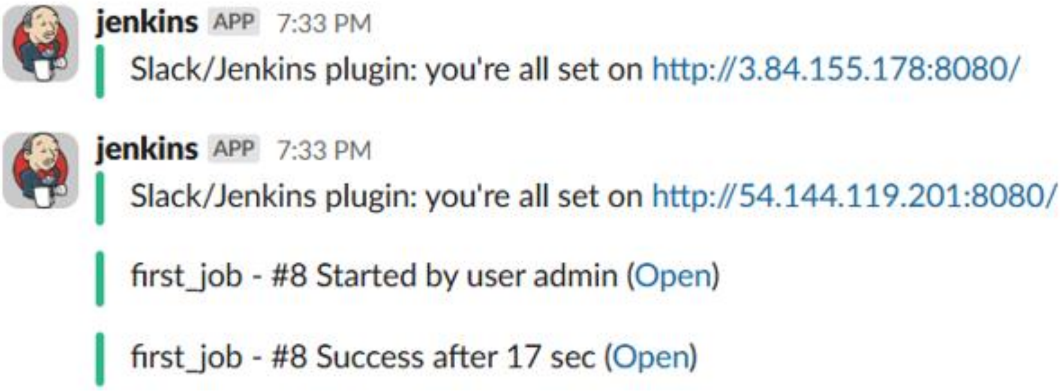
```
[root@ip-172-31-80-59 tomcat]# cd webapps/
[root@ip-172-31-80-59 webapps]# ls
ROOT  SimpleCustomerApp  SimpleCustomerApp.war  docs  examples  host-manager  manager
[root@ip-172-31-80-59 webapps]#
```

4) Nexus Artifactory



```
com
  javatpoint
    SimpleCustomerApp
      2-SNAPSHOT
      3-SNAPSHOT
      4-SNAPSHOT
      5-SNAPSHOT
```

5) Slack Notification



jenkins APP 7:33 PM

Slack/Jenkins plugin: you're all set on http://3.84.155.178:8080/

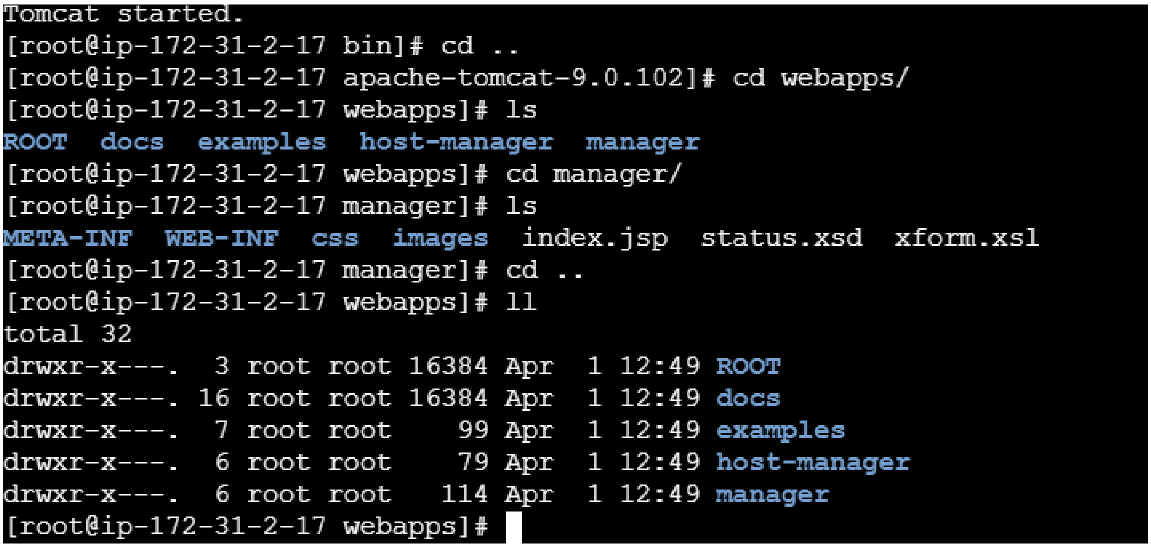
jenkins APP 7:33 PM

Slack/Jenkins plugin: you're all set on http://54.144.119.201:8080/

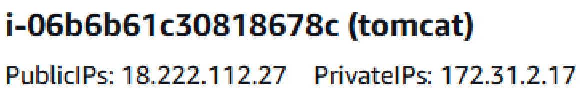
first_job - #8 Started by user admin (Open)

first_job - #8 Success after 17 sec (Open)

6) Deploy On tomcat

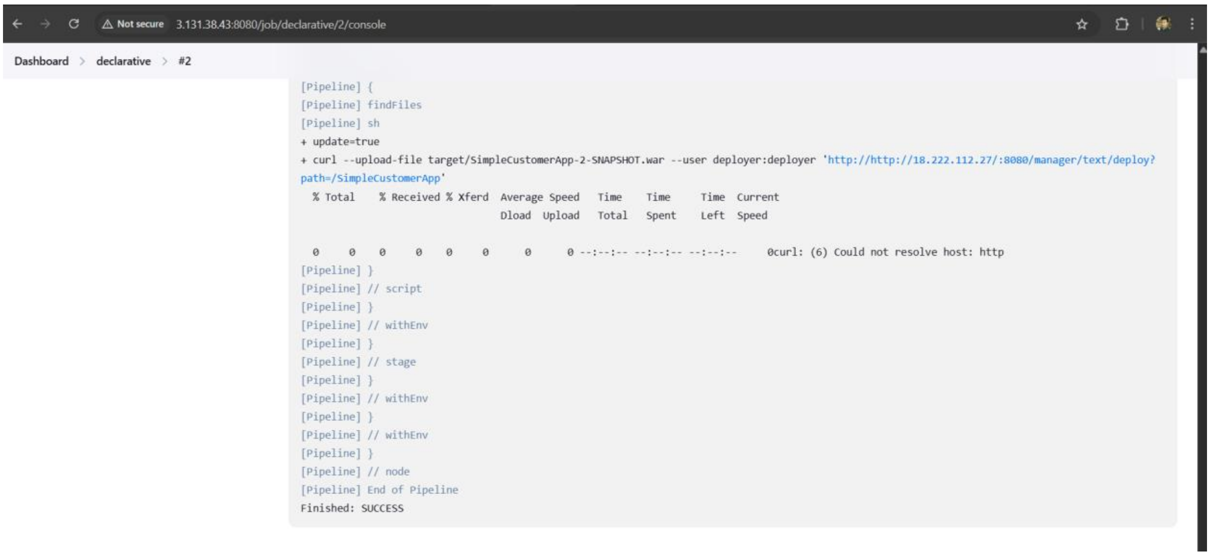


```
Tomcat started.
[root@ip-172-31-2-17 bin]# cd ..
[root@ip-172-31-2-17 apache-tomcat-9.0.102]# cd webapps/
[root@ip-172-31-2-17 webapps]# ls
ROOT  docs  examples  host-manager  manager
[root@ip-172-31-2-17 webapps]# cd manager/
[root@ip-172-31-2-17 manager]# ls
META-INF  WEB-INF  css  images  index.jsp  status.xsd  xform.xsl
[root@ip-172-31-2-17 manager]# cd ..
[root@ip-172-31-2-17 webapps]# ll
total 32
drwxr-x---.  3 root root 16384 Apr  1 12:49 ROOT
drwxr-x---. 16 root root 16384 Apr  1 12:49 docs
drwxr-x---.  7 root root    99 Apr  1 12:49 examples
drwxr-x---.  6 root root    79 Apr  1 12:49 host-manager
drwxr-x---.  6 root root   114 Apr  1 12:49 manager
[root@ip-172-31-2-17 webapps]# 
```



i-06b6b61c30818678c (tomcat)

PublicIPs: 18.222.112.27   PrivateIPs: 172.31.2.17



⚠ Not secure 3.131.38.43:8080/job/declarative/2/console
Dashboard > declarative > #2

```
[Pipeline] {
[Pipeline] findFiles
[Pipeline] sh
+ update=true
+ curl --upload-file target/SimpleCustomerApp-2-SNAPSHOT.war --user deployer:deployer 'http://http://18.222.112.27/:8080/manager/text/deploy?path=/SimpleCustomerApp'
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed

  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0curl: (6) Could not resolve host: http
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

4) Write sample skeleton of pipelines:

→skeleton of a Declarative Pipeline in Jenkins:

```
pipeline {
  agent any

  environment {
    // Define environment variables here
    MY_ENV_VAR = 'value'
}

stages {
    stage('Build') {
        steps {
            script {
                // Code for the build stage (e.g., compiling code, running tests)
                echo 'Building the project...'
                // Example build command
                sh './build.sh'
            }
        }
    }

    stage('Test') {
        steps {
            script {
                // Code for the test stage (e.g., running unit tests)
                echo 'Running tests...'
                // Example test command
                sh './run_tests.sh'
            }
        }
    }

    stage('Deploy') {
        steps {
            script {
                // Code for the deploy stage (e.g., deploying the app)
                echo 'Deploying the application...'
                // Example deploy command
                sh './deploy.sh'
            }
        }
    }
}

post {
    always {
        // Actions that will always run after the pipeline finishes
```

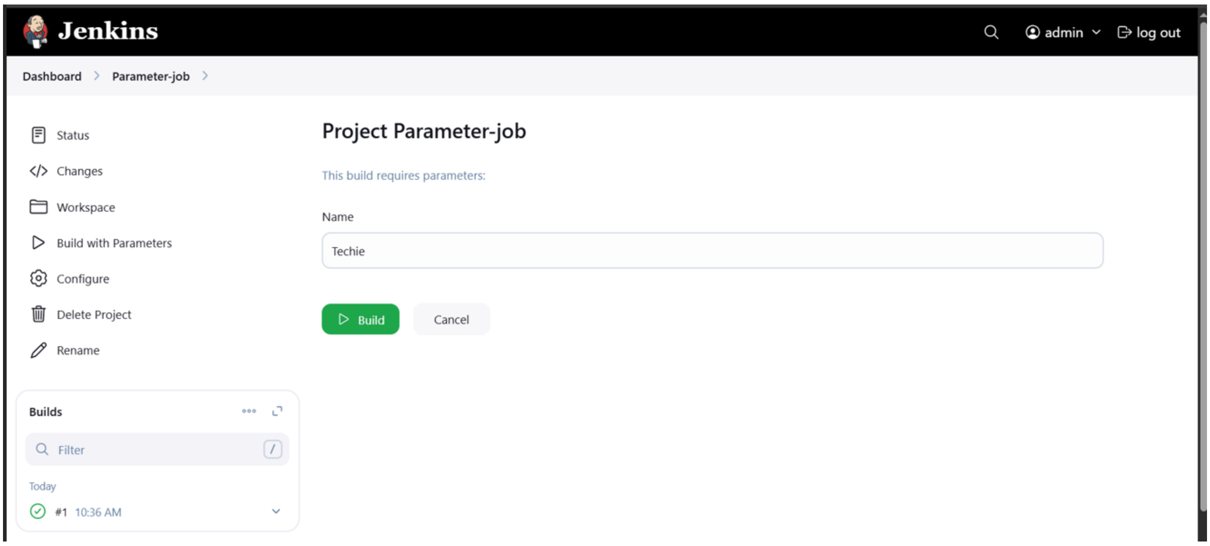
```
            echo 'Pipeline finished'
        }

        success {
            // Actions that will run only if the pipeline is successful
            echo 'Pipeline succeeded'
        }

        failure {
            // Actions that will run only if the pipeline fails
            echo 'Pipeline failed'
        }
    }
}
```

→ skeleton of a Scripted Pipeline in Jenkins:

```
node {
    try {
        stage('Checkout') {
            // Clone the repository
            checkout scm
        }

        stage('Build') {
            // Build commands
            echo "Building the project..."
        }

        stage('Test') {
            // Run tests
            echo "Running tests..."
        }

        stage('Deploy') {
            // Deployment steps
            echo "Deploying application..."
        }
    } catch (Exception e) {
        currentBuild.result = 'FAILURE'
        echo "Pipeline failed: ${e.getMessage()}"
    } finally {
        stage('Cleanup') {
            // Cleanup steps
            echo "Cleaning up..."
        }
    }
}
```

5)Create a parametirized job in jenkins. https://github.com/betawins/spring3-mvc-maven-xml-hello-world-1.git

→**String parameter:**



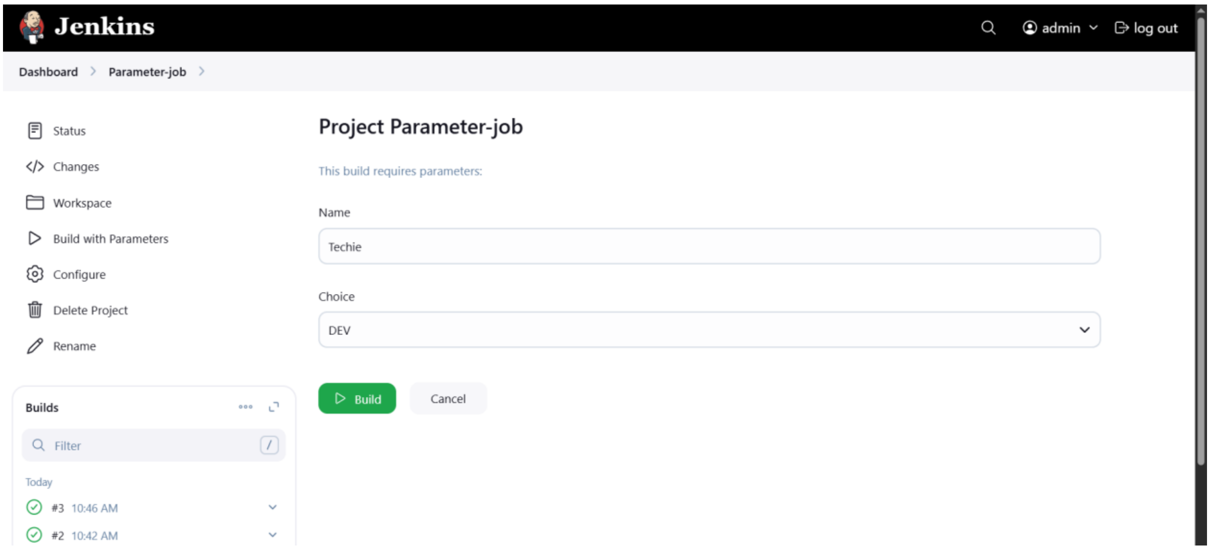
Project Parameter-job

This build requires parameters:

Name

Techie

→ Choice parameter:



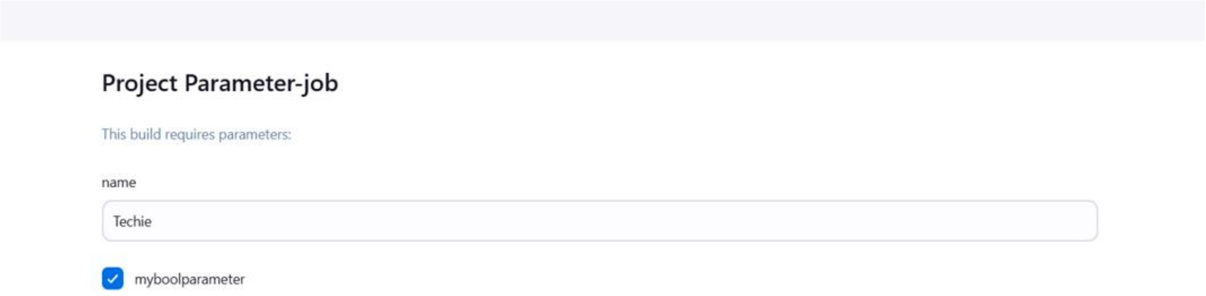
Project Parameter-job

This build requires parameters:

Name: Techie

Choice: DEV

→ Boolean Parameter:



Project Parameter-job

This build requires parameters:

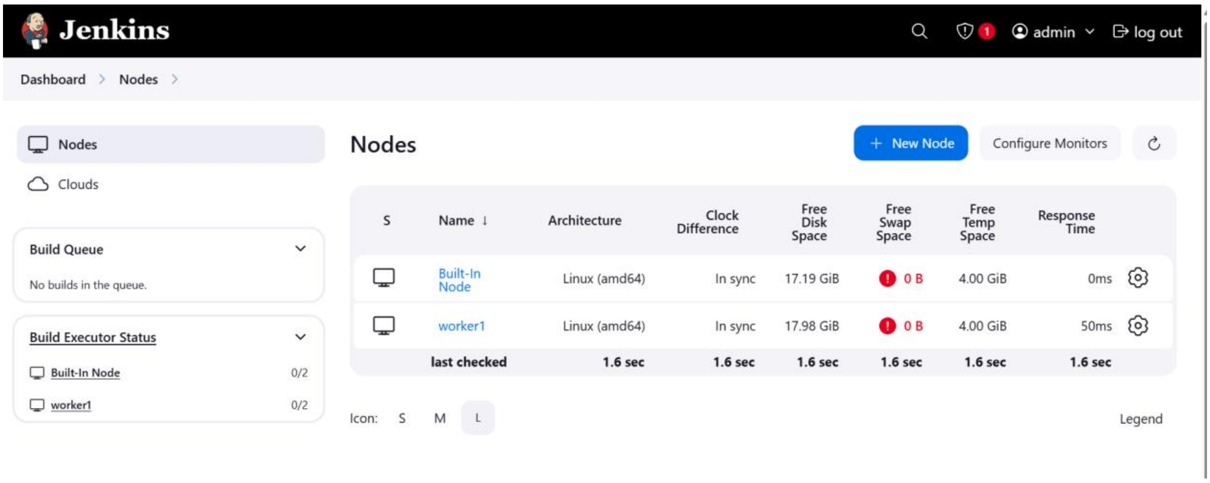
name: Techie

☑ myboolparameter

→ Password parameter:

→ Multi-line string parameter

6)Setup one slave machine for jenkins.



No builds in the queue.

Built-In Node 0/2

worker1 0/2

Nodes

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	17.19 GiB	0 B	4.00 GiB	0ms
	worker1	Linux (amd64)	In sync	17.98 GiB	0 B	4.00 GiB	50ms
	last checked	1.6 sec	1.6 sec	1.6 sec	1.6 sec	1.6 sec	1.6 sec