

# 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 ▾

The screenshot shows the GitHub repository page for 'hiring-app' by user 'betawins'. The repository is public and was forked from 'bcreddydevops/chinna-app'. It has 1 branch (main), 0 tags, and 36 commits. The main branch is 27 commits ahead of the parent repository. The repository contains several files and folders, including 'src/main/webapp', '.gitignore', 'Dockerfile', 'Jenkinsfile', 'README.md', 'Untitled Diagram.drawio', 'jenkinsfile-cicd', and 'pom.xml'. The right sidebar shows the repository's activity, including a README, activity, stars, and forks. The 'Languages' section shows that the repository is primarily composed of Dockerfile (84.7%) and Java (15.3%).

File/Folder	Commit Message	Commit Date
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

## 2) Sonarqube Integration:

The screenshot displays the SonarQube web interface. At the top, there's a navigation bar with links: Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, and More. A warning banner states: "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](#)". Below this, a message indicates a change in calculation methods for security, reliability, and maintainability counts and ratings, with a link to the documentation.

The main content area shows a project named "Test PUBLIC" with a status of "Passed". The last analysis was performed 10 minutes ago, resulting in 45 Lines of Code (XML, Docker, ...). The project's quality metrics are displayed as follows:

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

On the left, there are filters for Quality Gate (Passed: 1, Failed: 0) and Security (A: ≥ 0 info issues, B: ≥ 1 low issue, C: ≥ 1 medium issue, D: ≥ 1 high issue, E: ≥ 1 blocker issue, all with counts of 0).

## 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

The screenshot shows the Sonatype Nexus Repository web interface. The browser address bar indicates the URL is 18.232.139.165:8081. The page header includes the Sonatype logo, the text "Sonatype Nexus Repository COMMUNITY 3.78.2-04", a search bar, and a "Sign in" button.

The main content area is titled "Welcome" and features three primary actions:

- Browse**: Browse my repositories
- Search**: Search for new components
- Connect**: Connect to a repository

A left sidebar contains navigation links for "Browse", "Search", and "Browse".

## 5) Slack Notification



jenkins APP 7:33 PM

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



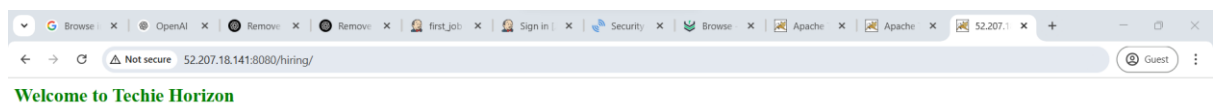
jenkins APP 7:33 PM

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



## 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)

### →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'
                ]],

```

```

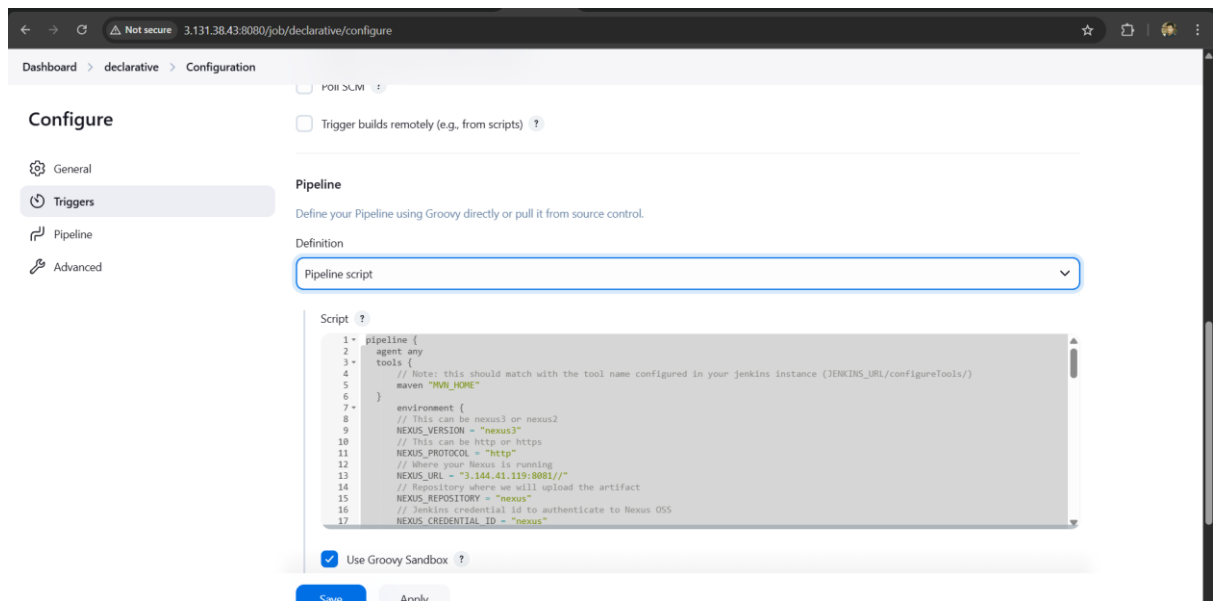
        credentialsId: 'nexus',
        groupId: 'com.javatpoint',
        nexusUrl: "${NEXUS_URL}",
        nexusVersion: "${NEXUS_VERSION}",
        protocol: "${NEXUS_PROTOCOL}",
        repository: "${NEXUS_REPOSITORY}",
        version: "${BUILD_NUMBER}-SNAPSHOT"
    )
}
}
}

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&u
pdate=true
            """
        }
    }
}
}
}
}

```

}



stages:

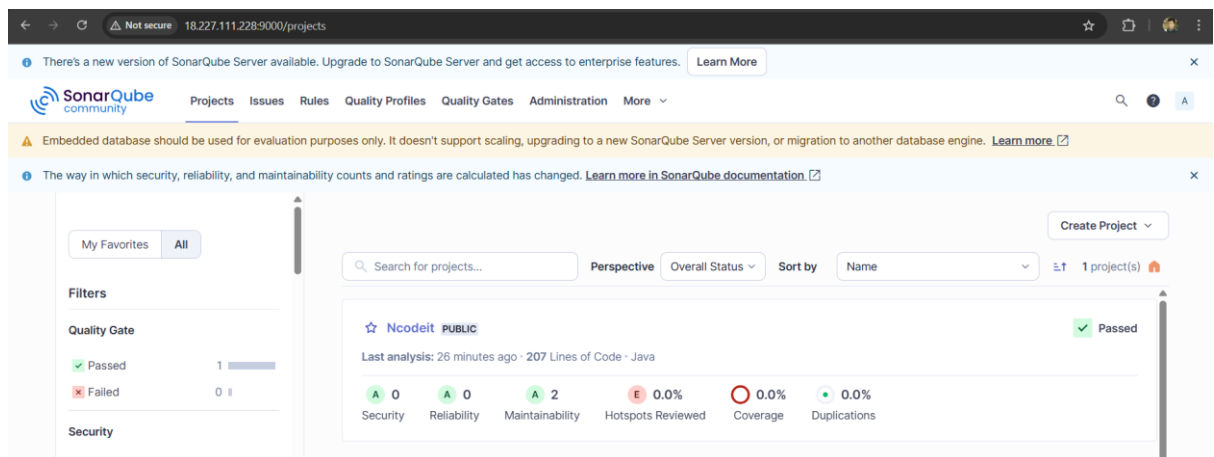
## 1) Git Clone

```
[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



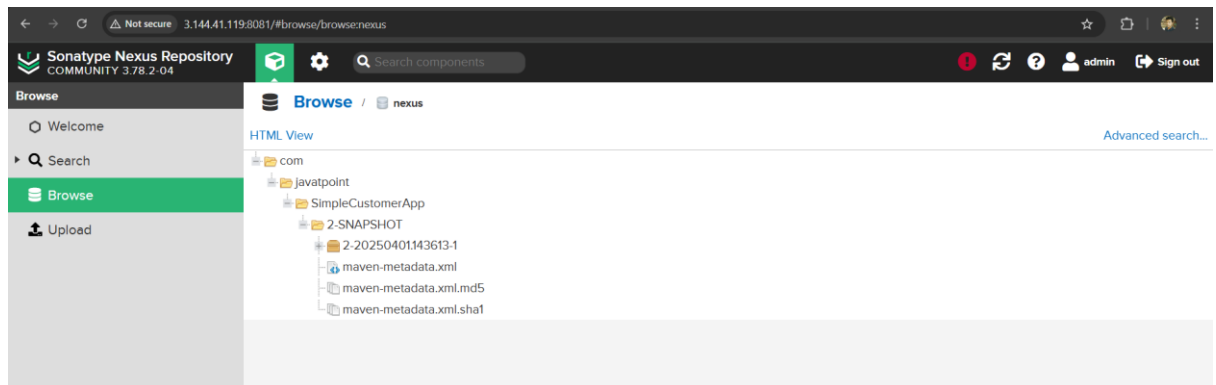
### 3) Maven Compilation

```
[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



### 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)

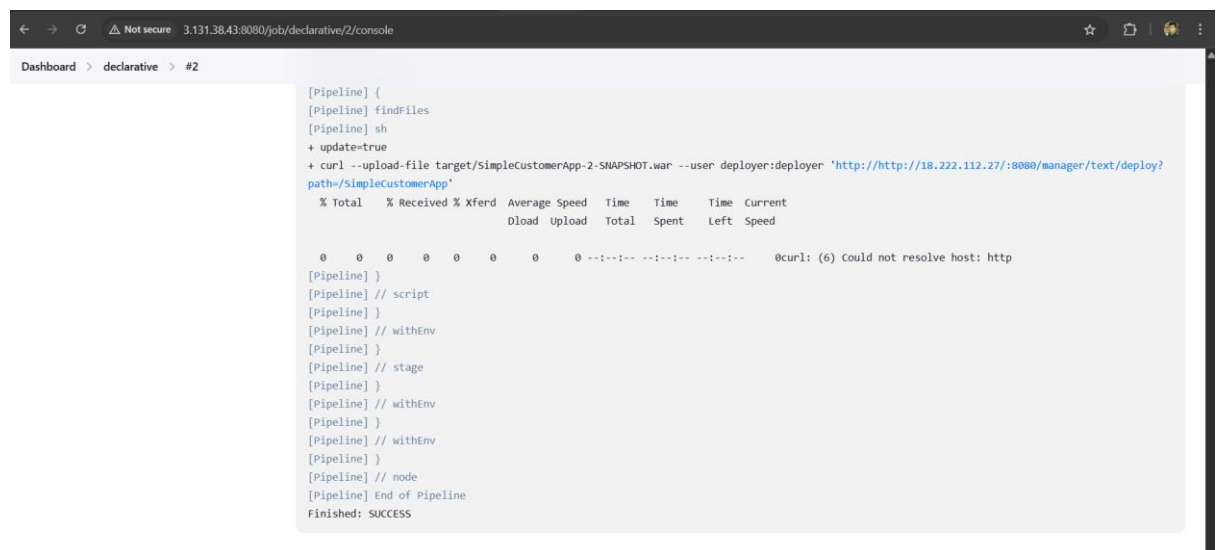


## 6) Deploy On tomcat

```
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.xml
[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



```
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] // withEnv
[Pipeline] // withEnv
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

## 3) Setup a Jenkins CI/CD 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)

→ 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"
    }
}

}

```

Dashboard > scripted > Configuration

### Configure

- General
- Triggers
- Pipeline**
- Advanced

### Pipeline

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

Definition

Pipeline script

Script

```

1 node {
2   def NEXUS_VERSION = "nexus3"
3   def NEXUS_PROTOCOL = "http"
4   def NEXUS_URL = "3.144.41.119:8081"
5   def NEXUS_REPOSITORY = "nexus"
6   def NEXUS_CREDENTIAL_ID = "nexus"
7   def SCANNER_HOME = tool 'sonarqube_server'
8
9   stage("Clone Code") {
10    git 'https://github.com/betawins/sabear_simplecustomerapp.git'
11  }
12
13  stage("Maven Build") {
14    sh 'mvn -Dmaven.test.failure.ignore=true clean install'
15  }
16
17  stage("SonarQube Analysis") {

```

☒ Use Groovy Sandbox

[Pipeline Syntax](#)

Save Apply

stages:

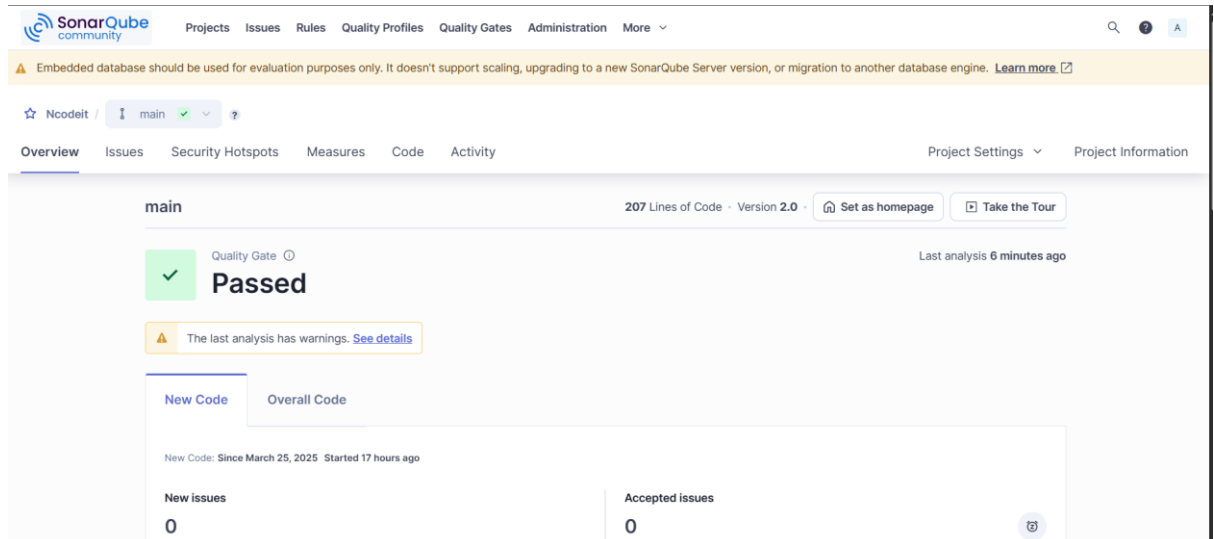
## 1) Git Clone

```
[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

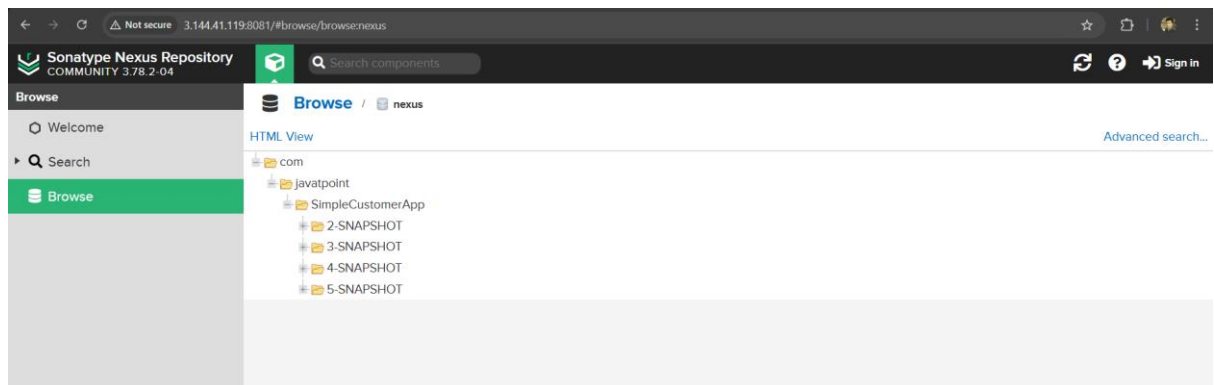


The screenshot displays the SonarQube web interface for a project named 'Ncodeit'. The top navigation bar includes links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, and More. A warning banner at the top states: 'Embedded database should be used for evaluation purposes only. It doesn't support scaling, upgrading to a new SonarQube Server version, or migration to another database engine. [Learn more](#)'. The main content area shows the 'main' branch with a 'Passed' status for the Quality Gate. A green checkmark icon is next to the word 'Passed'. Below this, a yellow warning box indicates: 'The last analysis has warnings. [See details](#)'. The interface also shows '207 Lines of Code' and 'Version 2.0'. On the right, there are buttons for 'Set as homepage' and 'Take the Tour'. At the bottom, there are two tabs: 'New Code' and 'Overall Code'. The 'New Code' tab is active, showing 'New issues: 0' and 'Accepted issues: 0'. The 'Overall Code' tab is also visible. The last analysis was performed 6 minutes ago.

## 3) Maven Compilation

```
[root@ip-172-31-80-59 webapps]# 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



## 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](#))

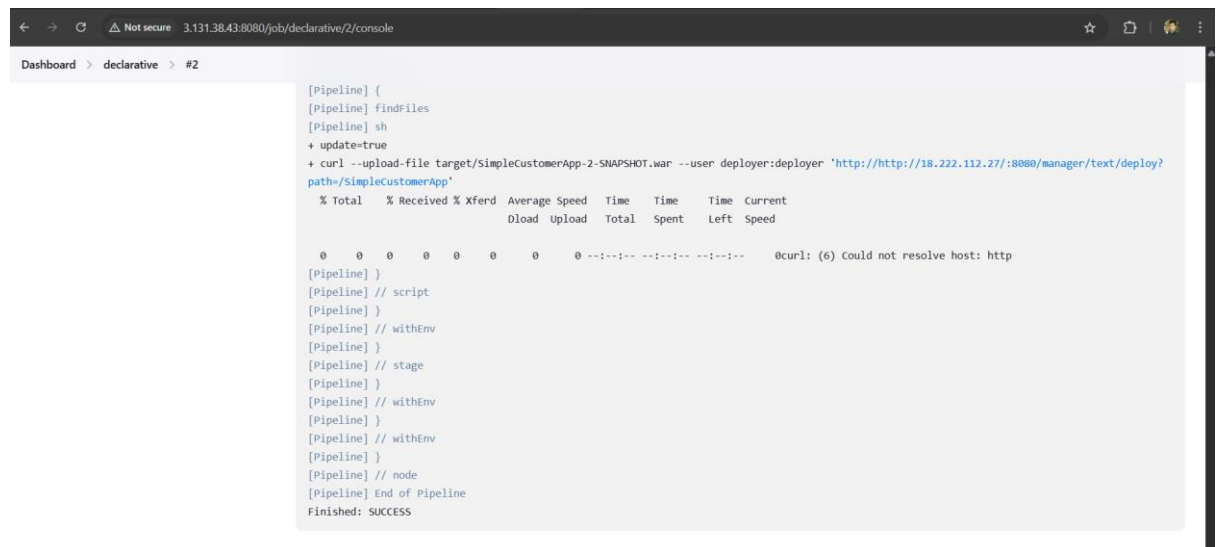
first\_job - #8 Success after 17 sec ([Open](#))

## 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.xml
[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



```
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..."
        }
    }
}

```



```

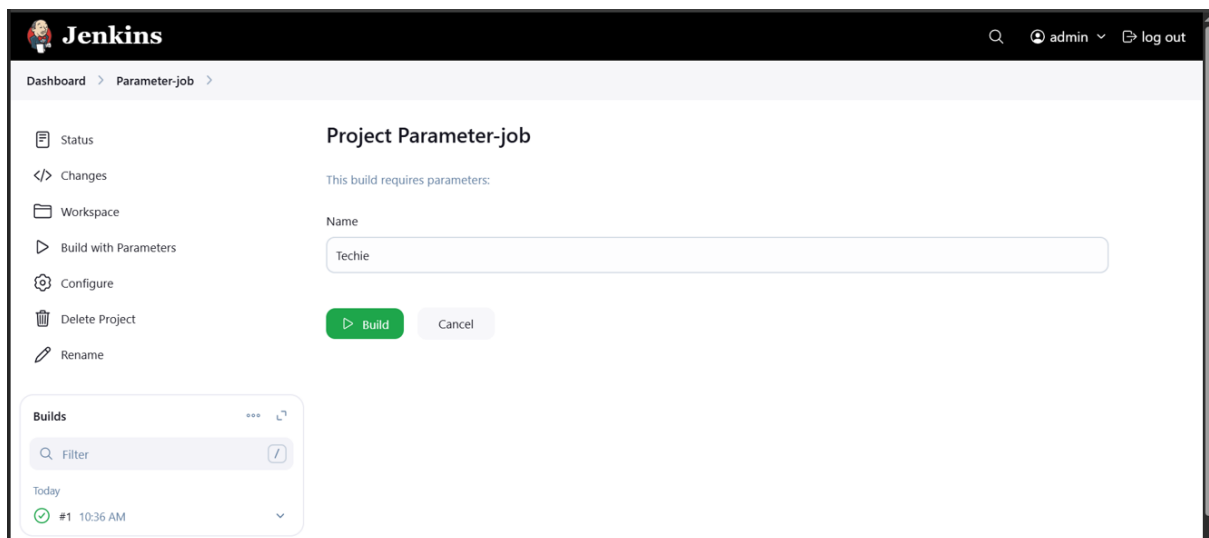
    }
} catch (Exception e) {
    currentBuild.result = 'FAILURE'
    echo "Pipeline failed: ${e.getMessage()}"
} finally {
    stage('Cleanup') {
        // Cleanup steps
        echo "Cleaning up..."
    }
}
}

```

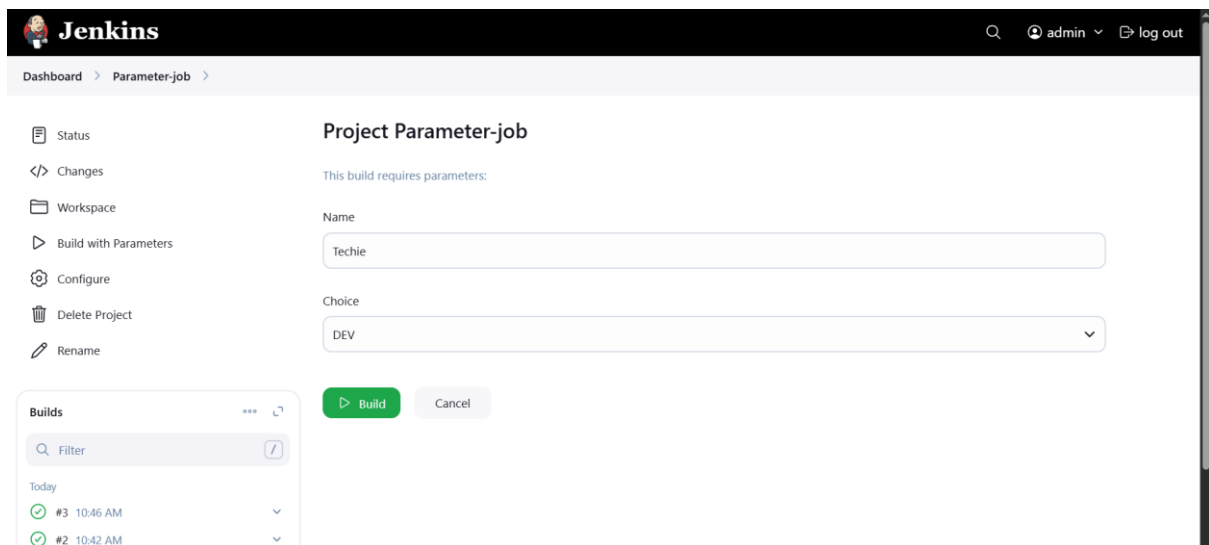
## 5) Create a parametrized job in Jenkins.

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

→String parameter:

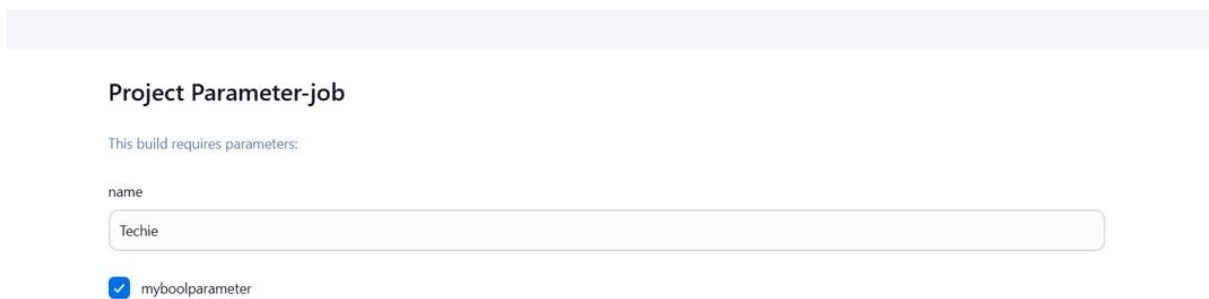


## → Choice parameter:



The screenshot shows the Jenkins web interface for configuring a project parameter job. The top navigation bar includes the Jenkins logo, a search icon, and user information (admin) with a log out link. The breadcrumb trail shows 'Dashboard > Parameter-job >'. On the left sidebar, there are links for Status, Changes, Workspace, Build with Parameters, Configure, Delete Project, and Rename. The main content area is titled 'Project Parameter-job' and includes a note 'This build requires parameters:'. Below this, there are two input fields: 'Name' with the value 'Techie' and 'Choice' with a dropdown menu showing 'DEV'. At the bottom of the configuration area, there are 'Build' and 'Cancel' buttons. On the left, there is a 'Builds' section with a filter input and a list of recent builds, including two successful builds from today.

## → Boolean Parameter:



The screenshot shows the Jenkins web interface for configuring a project parameter job. The top navigation bar includes the Jenkins logo, a search icon, and user information (admin) with a log out link. The breadcrumb trail shows 'Dashboard > Parameter-job >'. On the left sidebar, there are links for Status, Changes, Workspace, Build with Parameters, Configure, Delete Project, and Rename. The main content area is titled 'Project Parameter-job' and includes a note 'This build requires parameters:'. Below this, there is a 'name' input field with the value 'Techie'. Underneath, there is a checkbox labeled 'myboolparameter' which is checked.

## → Password parameter:

## → Multi-line string parameter

## 6) Setup one slave machine for jenkins.

The screenshot shows the Jenkins web interface. At the top is a black header with the Jenkins logo, a search icon, a shield icon with a red '1', a user icon labeled 'admin', and a 'log out' link. Below the header is a breadcrumb trail: 'Dashboard > Nodes >'. The main content area is titled 'Nodes' and includes a '+ New Node' button, a 'Configure Monitors' button, and a refresh icon. On the left sidebar, there are two expandable sections: 'Build Queue' (showing 'No builds in the queue.') and 'Build Executor Status' (showing 'Built-In Node' and 'worker1', both with '0/2' status). The main table lists the 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

Below the table, there is a legend for the 'S' column with icons for 'S', 'M', and 'L'.