

End-to-End CI/CD Pipeline with GitHub Actions, SonarQube, Nexus & Tomcat

1. Introduction

This project demonstrates an **end-to-end CI/CD pipeline** for a Java-based web application using **GitHub Actions** as the automation engine.

The workflow integrates the following key tools:

- **SonarQube** → Code quality analysis
- **Maven** → Build & package WAR files
- **Nexus Repository** → Artifact management
- **Apache Tomcat** → Application deployment

Everything is automated — from **code commit** to **deployment**.

2. CI/CD Pipeline Flow

Developer Commit → GitHub Actions → SonarQube → Maven Build → Nexus → Tomcat

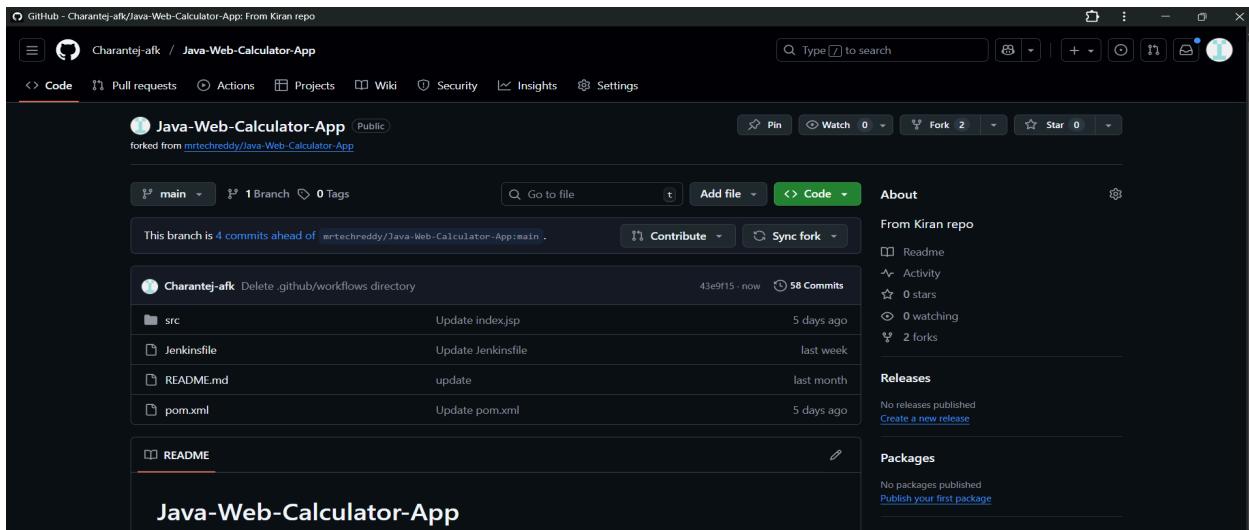
Pipeline Stages

Stage	Description	Tool
SonarQube Analysis	Run static code quality checks	SonarQube
Build Artifact	Create WAR file using Maven	Maven
Upload Artifact	Push WAR to Nexus repo	Nexus
Deploy to Tomcat	Force-clean and deploy new WAR	Apache Tomcat
Verify Deployment	Check if app is reachable	curl

3. Infrastructure Requirements:

1. Go to your GitHub repository.

a. My repo link <https://github.com/Charantej-afk/Java-Web-Calculator-App.git>

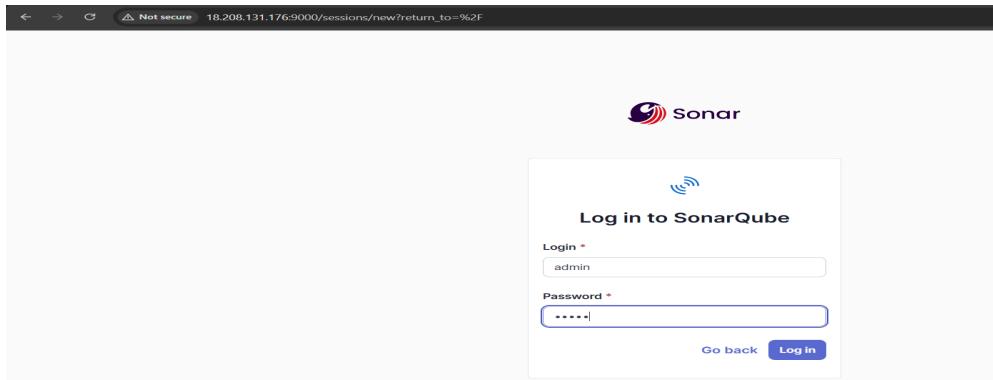


The screenshot shows a GitHub repository page for 'Java-Web-Calculator-App'. The repository is public and has 58 commits. The commit history includes updates to index.jsp, Jenkinsfile, README.md, and pom.xml. The repository has 2 forks and 0 stars. There are sections for About, Releases, and Packages.

2. Create a SonarQube Server, login into it and create a token for authentication.

```
ubuntu@ip-172-31-30-72:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [12
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [
```

```
ubuntu@SonarQube:~$ mkdir jenkins
ubuntu@SonarQube:~$ ls
jenkins sonarqube-24.12.0.100206 sonarqube-24.12.0.100206.zip
ubuntu@SonarQube:~$ ls -a
. .bash_history .bashrc .profile .sudo_as_admin_successful sonarqube-24.12.0.100206
.. .bash_logout .cache .ssh     jenkins sonarqube-24.12.0.100206.zip
ubuntu@SonarQube:~$ cd .ssh/
ubuntu@SonarQube:~/ssh$ ls
authorized_keys
ubuntu@SonarQube:~/ssh$ |
```



Tokens of Administrator

Generate Tokens

Name Expires in
Enter Token Name 30 days Generate

New token "SonarQube-Jenkins" has been created. Make sure you copy it now, you won't be able to see it again!
squ_21586efb87b9b458e4af8102f1dc3e8193d2b837

Name	Type	Project	Last use	Created	Expiration

Close

3. Create a Nexus account for Binary repository and log into to create a repository called maven2 hosted and named maven-releases.

```
ubuntu@nexus:~$ ls
jenkins  nexus-3.85.0-03  nexus-3.85.0-03-linux-x86_64.tar.gz  sonatype-work
ubuntu@nexus:~$ cd nexus-3.85.0-03/
ubuntu@nexus:~/nexus-3.85.0-03$ ls
NOTICE.txt  OSS-LICENSE.txt  bin  deploy  etc  jdk
ubuntu@nexus:~/nexus-3.85.0-03$ cd bin/
ubuntu@nexus:~/nexus-3.85.0-03/bin$ ls
nexus  nexus.vmoptions  sonatype-nexus-repository-3.85.0-03.jar
ubuntu@nexus:~/nexus-3.85.0-03/bin$ ./nexus
Usage: ./nexus {start|stop|run|run-redirect|status|restart|force-reload}
ubuntu@nexus:~/nexus-3.85.0-03/bin$ ./nexus start
Starting nexus
ubuntu@nexus:~/nexus-3.85.0-03/bin$ |
```

Nexus Repository Cloud is now available!

Publish and share components at any scale without the need to manage infrastructure. Take advantage of our limited time promotional offer.

Help us improve your upgrade experience

We want to understand what might be preventing you from upgrading to the latest version of Nexus Repository. Your feedback will help us create better resources and tools.

New Formats Supported

Name	Type	Format	Blob Store	Status	URL	Health check	Firewall Re...
maven-central	proxy	maven2	default	Online - Ready to Co...	<button>copy</button>	0 0	
maven-public	group	maven2	default	Online	<button>copy</button>		
maven-releases	hosted	maven2	default	Online	<button>copy</button>		
maven-snapshots	hosted	maven2	default	Online	<button>copy</button>		
nuget-group	group	nuget	default	Online	<button>copy</button>		
nuget-hosted	hosted	nuget	default	Online	<button>copy</button>		
nuget.org-proxy	proxy	nuget	default	Online - Ready to Co...	<button>copy</button>	0 0	

4. Create a Tomcat server for the deployment of the Java cal-web app and create a tomcat manager user and also login into the server.

```
ubuntu@ip-172-31-30-72:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [12
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [
```

```

<user username="root1" password="<must-be-changed>" roles="root"
-->
<role rolename="manager-gui"/>
<role rolename="manager-script"/>
<user username="tomcat" password="s3cret" roles="manager-gui"/>
</tomcat-users>

```

The screenshot shows the Tomcat Web Application Manager interface. At the top, there's a navigation bar with icons for back, forward, refresh, and a link to '98.91.17.31:8080/manager/html'. Below the bar is a logo of a yellow cat and the text 'THE ASF'. The main title is 'Tomcat Web Application Manager'. A message box says 'Message: OK'. The interface has several sections: 'Manager' (List Applications, HTML Manager Help, Manager Help, Server Status), 'Applications' (a table listing various Tomcat applications like /, /docs, /examples, /host-manager, and /manager with their details and command buttons), and a 'Deploy' section (Deploy directory or WAR file located on server).

4. Credentials Setup

Store all credentials securely as **GitHub Secrets** under
Repository → Settings → Secrets → Actions → New Repository Secret

Secret Name	Description	Example
SONAR_HOST_URL	SonarQube URL	http://52.23.202.186:9000
SONAR_TOKEN	SonarQube Access Token	squ_5309c3edd7a7a5d60d6e51e8121993155f62dad7
NEXUS_USER	Nexus username	admin

NEXUS_PASS	Nexus password	admin123
TOMCAT_USER	Tomcat Manager username	tomcat
TOMCAT_PASS	Tomcat Manager password	s3cret

GitHub - General

Access

- Collaborators
- Moderation options

Code and automation

- Branches
- Tags
- Rules
- Actions
- Models
- Webhooks
- Copilot
- Environments
- Codespaces
- Pages

Preview

Repository name: Java-Web-Calculator-App

Template repository

Require contributors to sign off on web-based commits

Default branch: main

Releases

Enable release immutability

Social preview

Actions secrets and variables

Secrets and variables allow you to manage reusable configuration data. Secrets are **encrypted** and are used for sensitive data. [Learn more about encrypted secrets](#). Variables are shown as plain text and are used for **non-sensitive** data. [Learn more about variables](#).

Anyone with collaborator access to this repository can use these secrets and variables for actions. They are not passed to workflows that are triggered by a pull request from a fork.

Secrets Variables

Environment secrets

This environment has no secrets.

[Manage environment secrets](#)

Repository secrets

This repository has no secrets.

[New repository secret](#)

This environment has no secrets.

[Manage environment secrets](#)

Repository secrets

Name	Last updated	
NEXUS_PASS	2 minutes ago	Edit Delete
NEXUS_USER	4 minutes ago	Edit Delete
SONAR_HOST_URL	6 minutes ago	Edit Delete
SONAR_TOKEN	5 minutes ago	Edit Delete
TOMCAT_PASS	now	Edit Delete
TOMCAT_USER	now	Edit Delete

[New repository secret](#)

5. Repository Folder Structure

Java-Web-Calculator-App/

```
|  
|   └── src/  
|       |   └── main/  
|       |       |   └── java/  
|       |       |   └── webapp/  
|  
|   └── pom.xml  
└── .github/  
    └── workflows/  
        └── ci-cd.yml ← GitHub Actions Pipeline
```



6. GitHub Actions Workflow File

📁 Path: [.github/workflows/ci-cd.yml](#)

```
name: Java CI/CD - SonarQube, Nexus, Tomcat (Full Clean Deploy + Slack Alerts)
```

```
on:
```

```
  push:
```

```
    branches:
```

```
      - main
```

```
  workflow_dispatch:
```

```
jobs:
```

```
  build-deploy:
```

```
    runs-on: ubuntu-latest
```

```
env:
```

```
  SONAR_HOST_URL: ${{ secrets.SONAR_HOST_URL }}
```

```
  SONAR_TOKEN: ${{ secrets.SONAR_TOKEN }}
```

```
  NEXUS_URL: http://18.209.9.207:8081
```

```
  NEXUS_REPO: maven-releases
```

```
  NEXUS_GROUP: com/web/cal
```

```
  NEXUS_ARTIFACT: webapp-add
```

```
  TOMCAT_URL: http://98.81.123.228:8080/manager/text
```

```
  SLACK_WEBHOOK_URL: ${{ secrets.SLACK_WEBHOOK_URL }}
```

```
steps:
```

```
  - name: 📦 Checkout Code
```

```
    uses: actions/checkout@v4
```

```
  - name: ☕ Setup Java 17
```

```
    uses: actions/setup-java@v4
```

```
    with:
```

```
      distribution: 'temurin'
```

```
      java-version: '17'
```

```
  - name: 💾 Cache Maven Dependencies
```

```
    uses: actions/cache@v4
```

```
    with:
```

```

path: ~/m2
key: ${{ runner.os }}-maven-${{ hashFiles('**/pom.xml') }}
restore-keys: |
${{ runner.os }}-maven-

- name: 🔎 Run SonarQube Analysis
run: |
mvn clean verify sonar:sonar \
-DskipTests \
-Dsonar.projectKey=JavaWebCalculator \
-Dsonar.host.url=${{ secrets.SONAR_HOST_URL }} \
-Dsonar.login=${{ secrets.SONAR_TOKEN }}

- name: 🛡 Build WAR File
run: |
mvn clean package -DskipTests
echo "✅ Build completed!"
ls -lh target/*.war

- name: 📤 Upload WAR to Nexus
env:
NEXUS_USER: ${{ secrets.NEXUS_USER }}
NEXUS_PASS: ${{ secrets.NEXUS_PASS }}
run: |
set -e
WAR_FILE=$(ls target/*.war | head -1)
VERSION="0.0.${{ github.run_number }}"
echo "📦 Uploading WAR to Nexus version $VERSION ..."
curl -u ${NEXUS_USER}:${NEXUS_PASS} --upload-file "$WAR_FILE" \
"${{ env.NEXUS_URL }}/repository/${{ env.NEXUS_REPO }}/${{ env.NEXUS_GROUP }}/${{ env.NEXUS_ARTIFACT }}/${VERSION}/${{ env.NEXUS_ARTIFACT }}-${VERSION}.war"
echo "✅ Uploaded successfully."

- name: 🚀 Deploy WAR to Tomcat (Force Clean)
env:
TOMCAT_USER: ${{ secrets.TOMCAT_USER }}
TOMCAT_PASS: ${{ secrets.TOMCAT_PASS }}

```

```

NEXUS_USER: ${ secrets.NEXUS_USER }
NEXUS_PASS: ${ secrets.NEXUS_PASS }

run: |
  set -e
  APP_NAME="webapp-add"
  cd /tmp; rm -f *.war

  echo "🔍 Fetching latest WAR from Nexus..."
  DOWNLOAD_URL=$(curl -s -u ${NEXUS_USER}:${NEXUS_PASS} \
    "${{ env.NEXUS_URL }}/service/rest/v1/search/assets?repository=${{ env.NEXUS_REPO
}}" \
  | grep -oP '"downloadUrl"\s*:\s*"[^"]*webapp-add-[0-9.]+\.\war' | tail -1 | cut -d"" -f4)

  if [ -z "$DOWNLOAD_URL" ]; then
    echo "❌ No WAR found in Nexus!"
    exit 1
  fi

  echo "⬇️ Downloading WAR: $DOWNLOAD_URL"
  curl -u ${NEXUS_USER}:${NEXUS_PASS} -O "$DOWNLOAD_URL"
  WAR_FILE=$(basename "$DOWNLOAD_URL")

  echo "🧹 Undeploying old application..."
  curl -u ${TOMCAT_USER}:${TOMCAT_PASS} "${{ env.TOMCAT_URL
}}/undeploy?path=/${APP_NAME}" || true
  sleep 5

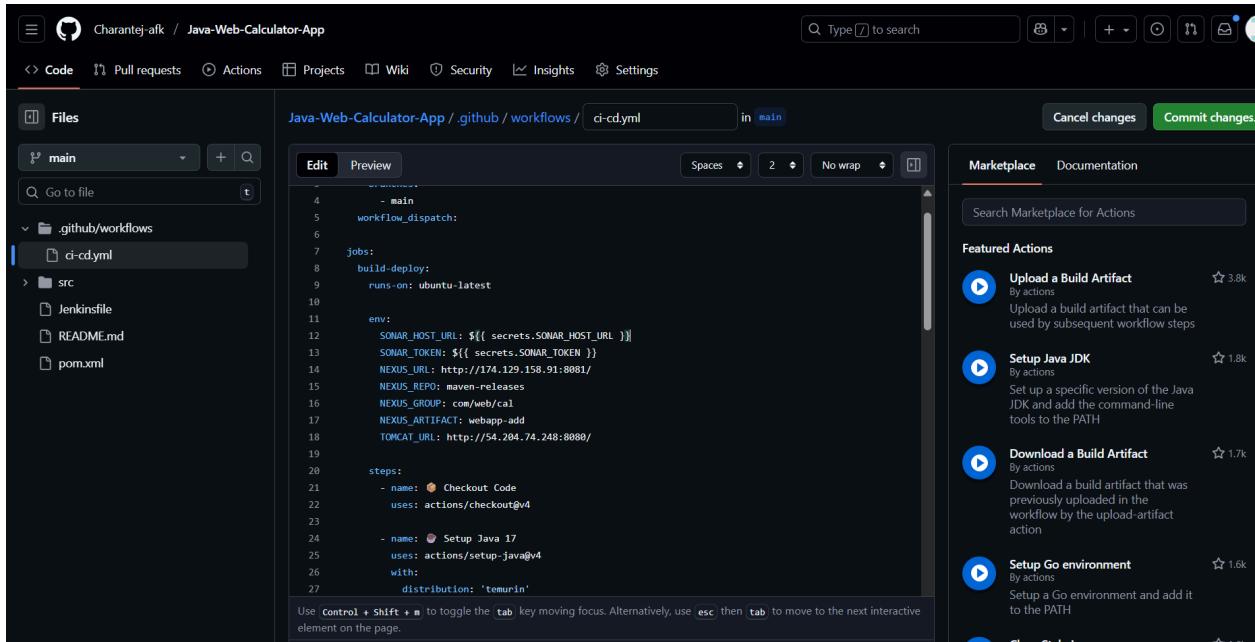
  echo "🧹 Cleaning Tomcat cache directories via manager text..."
  curl -u ${TOMCAT_USER}:${TOMCAT_PASS} "${{ env.TOMCAT_URL
}}/expire?path=/${APP_NAME}" || true
  curl -u ${TOMCAT_USER}:${TOMCAT_PASS} "${{ env.TOMCAT_URL
}}/reload?path=/${APP_NAME}" || true

  echo "🚀 Deploying fresh WAR to Tomcat..."
  curl -u ${TOMCAT_USER}:${TOMCAT_PASS} --upload-file "$WAR_FILE" \
    "${{ env.TOMCAT_URL }}/deploy?path=/${APP_NAME}&update=true"

```

```
echo "✅ Deployment completed successfully!"
```

```
- name: 🔎 Verify Deployment
id: verify
run: |
  sleep 15
  STATUS=$(curl -o /dev/null -s -w "%{http_code}"
  http://98.81.123.228:8080/webapp-add/)
  if [ "$STATUS" -eq 200 ]; then
    echo "✅ App is live with the latest code!"
  else
    echo "⚠️ App returned status $STATUS"
    exit 1
fi
```



The screenshot shows the GitHub Actions workflow editor for a repository named 'Java-Web-Calculator-App'. The workflow file is named 'ci-cd.yml' and is currently being edited. The code in the file is as follows:

```
name: Java-Web-Calculator-App CI/CD
on: [push]
jobs:
  build-deploy:
    runs-on: ubuntu-latest
    env:
      SONAR_HOST_URL: ${{ secrets.SONAR_HOST_URL }}
      SONAR_TOKEN: ${{ secrets.SONAR_TOKEN }}
      NEXUS_URL: http://174.129.158.91:8081/
      NEXUS_REPO: maven-releases
      NEXUS_GROUP: com/web/cal
      NEXUS_ARTIFACT: webapp-add
      TOMCAT_URL: http://54.204.74.248:8080/
    steps:
      - name: 🛒 Checkout Code
        uses: actions/checkout@v4
      - name: 🐕 Setup Java 17
        uses: actions/setup-java@v4
        with:
          distribution: 'temurin'
```

The GitHub interface shows the 'Files' sidebar with 'main' selected. The right side features a 'Marketplace' section with several featured actions: 'Upload a Build Artifact', 'Setup Java JDK', 'Download a Build Artifact', and 'Setup Go environment'. There is also a 'Close stale issues' button.

🧠 7. Triggering the Pipeline

You can trigger the workflow in two ways:

- **Automatic:** On every push to the `main` branch
- **Manual:** Go to **Actions** → **Java CI/CD - SonarQube, Nexus, Tomcat (Full Clean Deploy)** → Click **Run workflow**

The screenshot shows a CI/CD pipeline interface for a Java Web App. At the top, a green checkmark icon indicates a successful job titled "Refactor echo statements for clarity in CI/CD workflow #7". Below this, a sidebar on the left lists "Summary", "Jobs" (with "build-deploy" selected), "Run details", "Usage", and "Workflow file". The main area displays the "build-deploy" job, which succeeded in 1m 10s. The job steps are listed as follows:

- > Set up job
- > Checkout Code
- > Setup Java 17
- > Cache Maven Dependencies
- > SonarCube Analysis (Self-Hosted)
- > Build WAR Artifact
- > Upload WAR to Nexus Repository
- > Deploy WAR to Tomcat
- > Verify Tomcat Deployment
- > Post Cache Maven Dependencies
- > Post Setup Java 17
- > Post Checkout Code
- > Complete job

A search bar at the top right says "Search logs".

✓ 8. Verification Checklist

1. Verify Nexus for the Artifact.

The screenshot shows the Sonatype Nexus Repository Manager interface. The top navigation bar includes "sonatype nexus repository Community Edition", a search bar, and icons for "Alerts" and "Logs". The left sidebar has buttons for "Dashboard", "Search", "Browse" (which is highlighted in green), "Upload", and "Settings". The main content area is titled "Browse" and shows a table of repositories:

Name ↑	Type	Format	Status	URL	Health check
maven-central	proxy	maven2	Online - Ready to Connect	<button>copy</button>	0 0
maven-public	group	maven2	Online	<button>copy</button>	0
maven-releases	hosted	maven2	Online	<button>copy</button>	0
maven-snapshots	hosted	maven2	Online	<button>copy</button>	0
nuget-group	group	nuget	Online	<button>copy</button>	0
nuget-hosted	hosted	nuget	Online	<button>copy</button>	0
nuget.org-proxy	proxy	nuget	Online - Ready to Connect	<button>copy</button>	0 0

2. Verify SonarQube for the test results.

3. Verify tomcat for the Deployed Artifact and test whether it is working or not.

Path	Version	Display Name	Running	Sessions	Commands
/	<i>None specified</i>	Welcome to Tomcat	true	0	Start Stop Reload Undeploy [Expire sessions] with idle ≥ 30
/docs	<i>None specified</i>	Tomcat Documentation	true	0	Start Stop Reload Undeploy [Expire sessions] with idle ≥ 30
/examples	<i>None specified</i>	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy [Expire sessions] with idle ≥ 30
/host-manager	<i>None specified</i>	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy [Expire sessions] with idle ≥ 30
/manager	<i>None specified</i>	Tomcat Manager Application	true	1	Start Stop Reload Undeploy [Expire sessions] with idle ≥ 30
/webapp-add	<i>None specified</i>	Servlet	true	1	Start Stop Reload Undeploy [Expire sessions] with idle ≥ 30

Calculator

Calculator

first number:

Second number :

addition
 subtraction
 production

first number:

Second number :

addition
 subtraction
 production

Addition

30

Subtraction

10

Multiplication

200

Calculator

first number:

Second number :

addition
 subtraction
 production

⚠ 9. Troubleshooting

Issue	Cause	Solution
Old app still loads	Tomcat cache not cleared	Added <code>/undeploy</code> and redeploy step
401 Unauthorized (Tomcat)	Invalid credentials	Update <code>tomcat-users.xml</code>
SonarQube step fails	Server not reachable	Check host & token

Nexus upload 400

Duplicate artifact
version

Version auto-increment added



10. Improvements & Add-Ons

Feature	Description
SonarQube Quality Gate	Fail pipeline if quality < 80%
Rollback Feature	Deploy previous version from Nexus
Multiple Environments	Add staging/production workflows
Parallel Jobs	Split build, test, and deploy stages



11. Summary

Component	Role
GitHub Actions	CI/CD engine
Maven	Build automation
SonarQube	Code quality
Nexus	Artifact repository
Tomcat	Deployment server