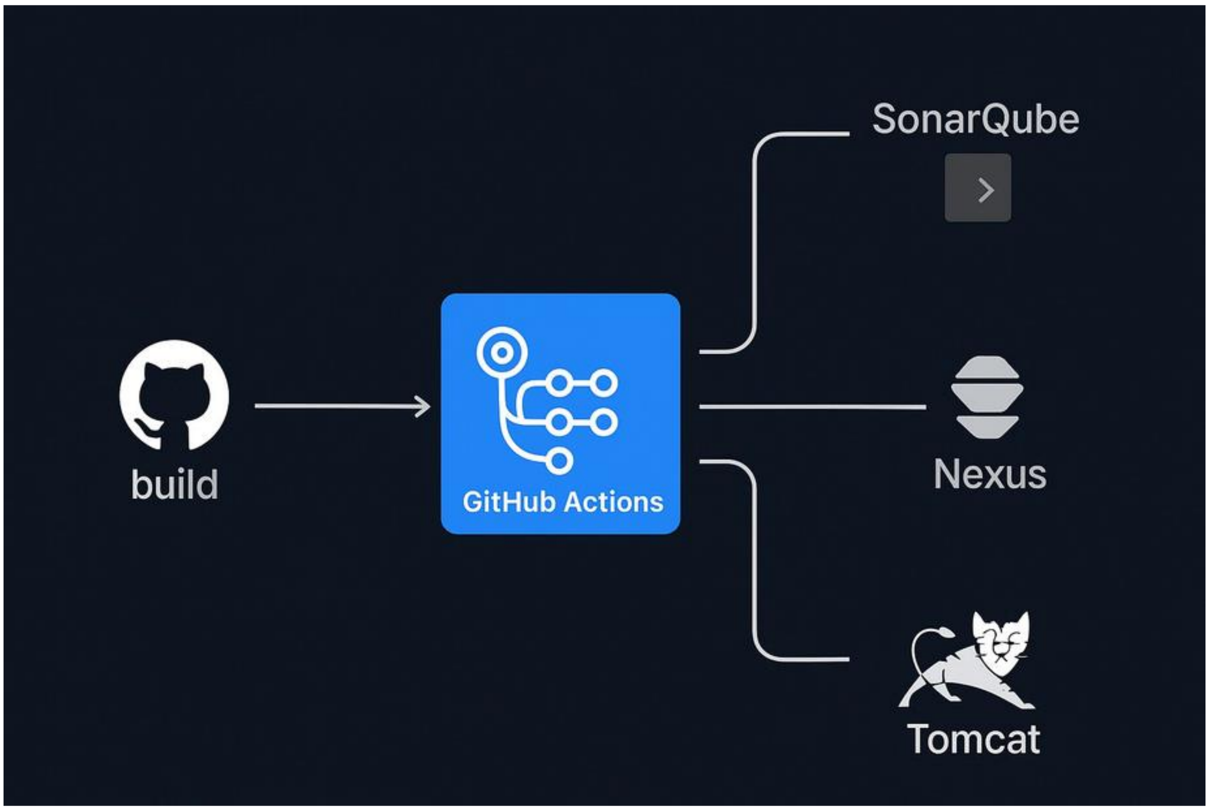
Name: Alekya K

PROJECT – 5

Complete CI/CD workflow for Java web deployment leveraging GitHub Actions,SonarQube analysis, Nexus artifact management, and Tomcat server delivery

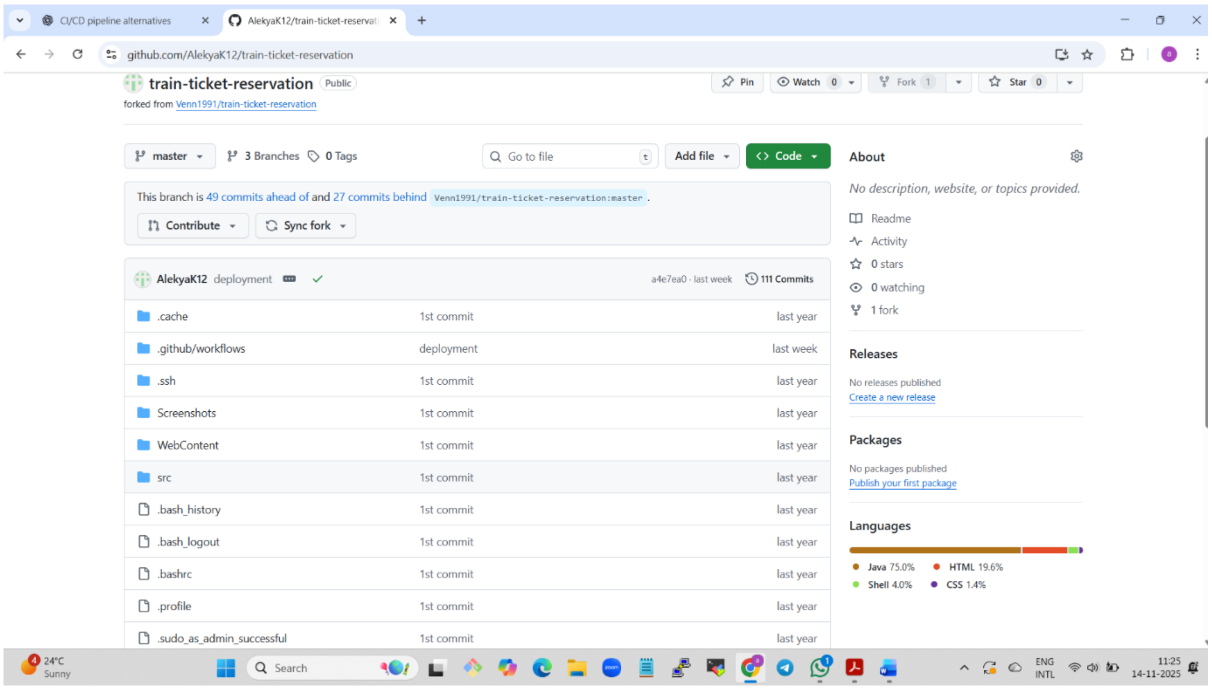


ProjectOverview:
This project showcases a fully automated deployment workflow for a Java web application using GitHub Actions. The pipeline is triggered whenever new code is pushed to the repository, ensuring the application is built, tested, scanned, and deployed without manual intervention.

Integrated Tools:
SonarQube: Performs static code analysis to ensure code quality and detect bugs or vulnerabilities prior to deployment.
Nexus Repository: Serves as a storage and versioning system for build artifacts such as WAR files created during the Maven build.
Tomcat Server: Hosts and deploys the final WAR file, making the application available to end users.

Step 1: Code in GitHub

The Java app is stored on GitHub, and any push to the master branch triggers the CI/CD workflow.



github.com/AlekyaK12/train-ticket-reservation

train-ticket-reservation Public

forked from Venn1991/train-ticket-reservation

master · 3 Branches · 0 Tags

This branch is 49 commits ahead of and 27 commits behind Venn1991/train-ticket-reservation:master .

AlekyaK12 deployment ✓ a4e7ea0 · last week 111 Commits

.cache	1st commit	last year
.github/workflows	deployment	last week
.ssh	1st commit	last year
Screenshots	1st commit	last year
WebContent	1st commit	last year
src	1st commit	last year
.bash_history	1st commit	last year
.bash_logout	1st commit	last year
.bashrc	1st commit	last year
.profile	1st commit	last year
.sudo_as_admin_successful	1st commit	last year

About

No description, website, or topics provided.

Readme · Activity · 0 stars · 0 watching · 1 fork

Releases: No releases published

Packages: No packages published

Languages: ● Java 75.0% ● HTML 19.6% ● Shell 4.0% ● CSS 1.4%

Java Web Application in Github Repo

Step 2: Pipeline Steps in GitHub Actions

Pull the latest code

Set up JDK 17

Build and test with Maven

Scan with SonarQube

Upload the WAR file to Nexus

Deploy the application to Tomcat

Step 3: Setup on AWS EC2

Three EC2 machines (t2 large,15-20GB)were created for SonarQube, Nexus, and Tomcat.Docker was installed on the servers to run the required tools.

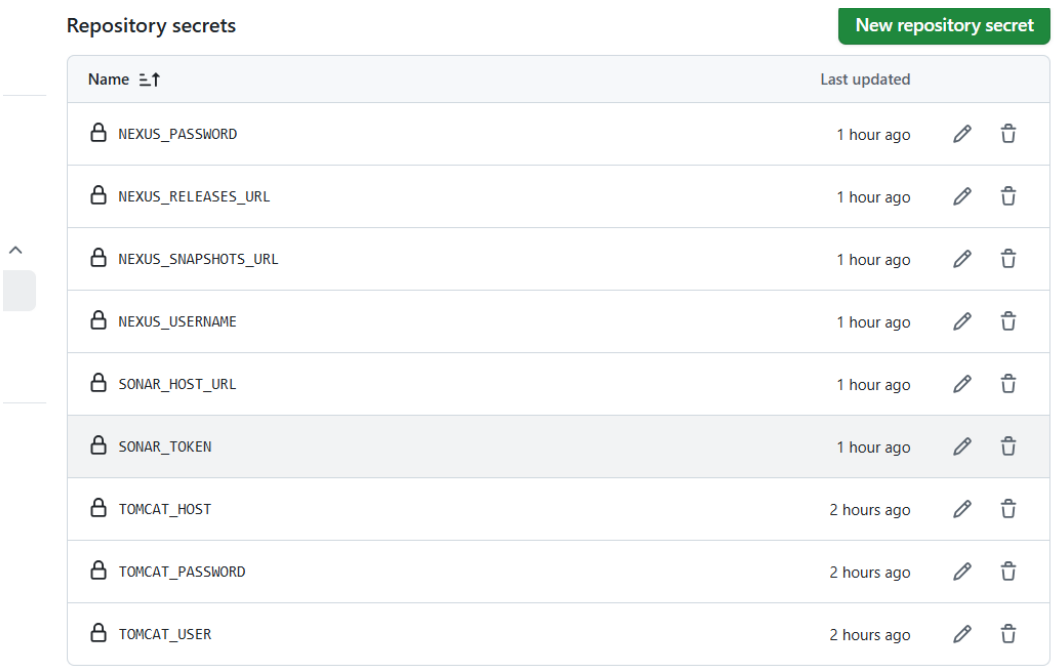
Step 4: Running SonarQube & Nexus with Docker

SonarQube and Nexus were started as Docker containers using:

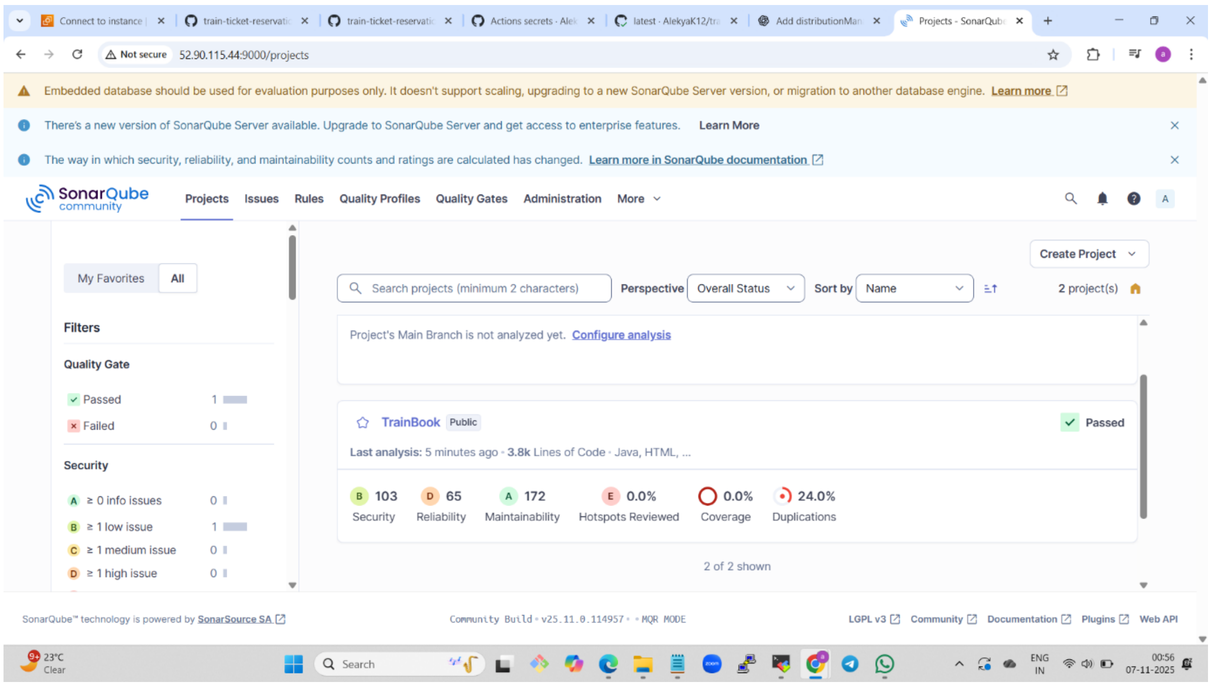
```
docker run -d -p 9000:9000 sonarqube
docker run -d -p 8081:8081 sonatype/nexus3
```

Step 5: Tomcat Setup
Tomcat was installed on another EC2 instance, a deployment user was configured, and the Tomcat dashboard was accessed through the EC2 public IP.
Step 6: SonarQube Integration
A SonarQube token was generated and saved in GitHub Secrets so GitHub Actions can run code analysis.The results can be viewed on the SonarQube dashboard.

Step 7: Nexus Repository Configuration
A Nexus server running in a Docker container on EC2 was configured with separate **snapshot** and **release** repositories.Maven is set up to deploy artifacts automatically based on version patterns—snapshot versions (*-SNAPSHOT) are pushed to the snapshot repository, while stable versions are pushed to the releaserepository.
This structure enforces clean artifact lifecycle management.



Name	Last updated
NEXUS_PASSWORD	1 hour ago
NEXUS_RELEASES_URL	1 hour ago
NEXUS_SNAPSHOTS_URL	1 hour ago
NEXUS_USERNAME	1 hour ago
SONAR_HOST_URL	1 hour ago
SONAR_TOKEN	1 hour ago
TOMCAT_HOST	2 hours ago
TOMCAT_PASSWORD	2 hours ago
TOMCAT_USER	2 hours ago



52.90.115.44:9000/projects

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

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

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

Filters

Quality Gate

Passed	1
Failed	0

Security

A ≥ 0 info issues	0
B ≥ 1 low issue	1
C ≥ 1 medium issue	0
D ≥ 1 high issue	0

Perspective: Overall Status — Sort by: Name — 2 project(s)

Project's Main Branch is not analyzed yet. Configure analysis

TrainBook Public — Passed

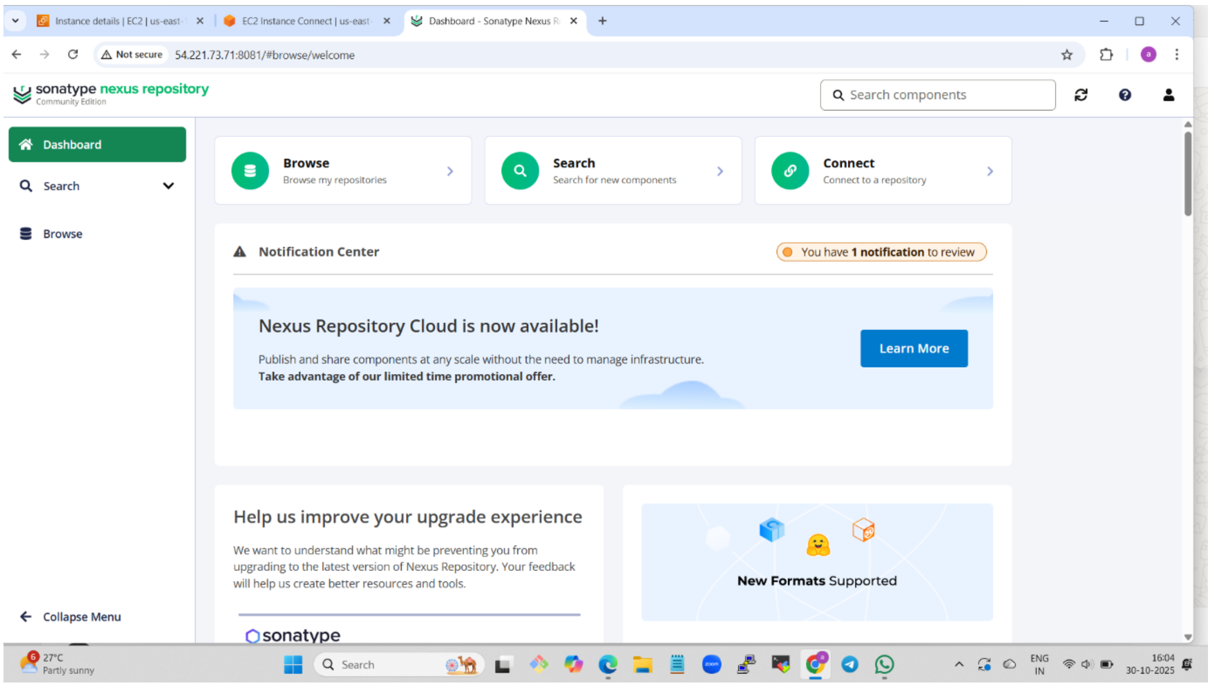
Last analysis: 5 minutes ago · 3.8k Lines of Code · Java, HTML, ...

Security	Reliability	Maintainability	Hotspots Reviewed	Coverage	Duplications
B 103	D 65	A 172	E 0.0%	0.0%	24.0%

2 of 2 shown

SonarQube™ technology is powered by SonarSource SA — Community Build - v25.11.0.114957 - MQR MODE

SonarQube dashboard displaying code quality and vulnerability results



54.221.73.71:8081/#browse/welcome

sonatype nexus repository Community Edition

Browse — Browse my repositories

Search — Search for new components

Connect — Connect to a repository

Notification Center — You have 1 notification to review

Nexus Repository Cloud is now available!

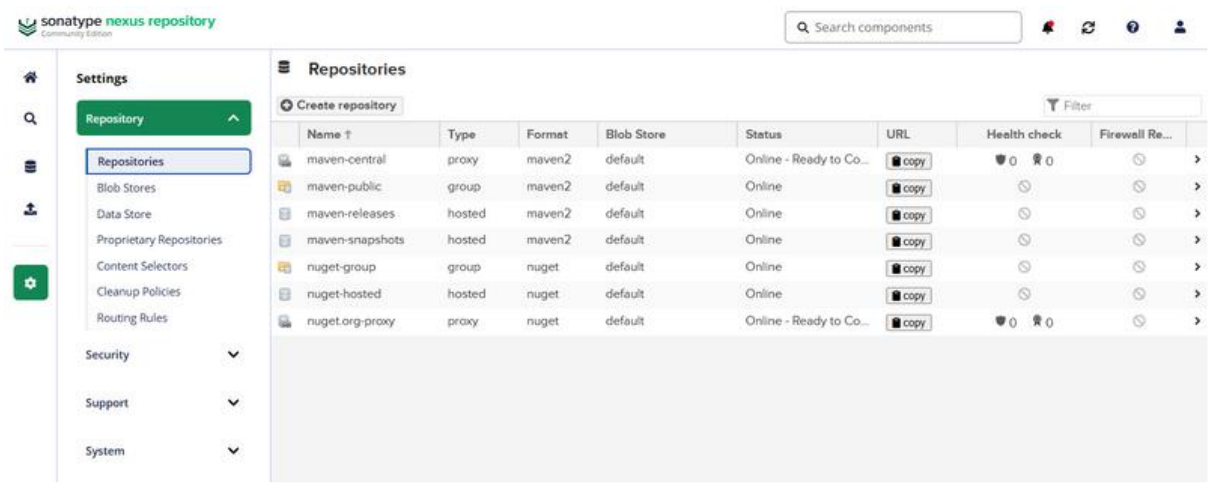
Publish and share components at any scale without the need to manage infrastructure.
Take advantage of our limited time promotional offer.

Learn More

Help us improve your upgrade experience

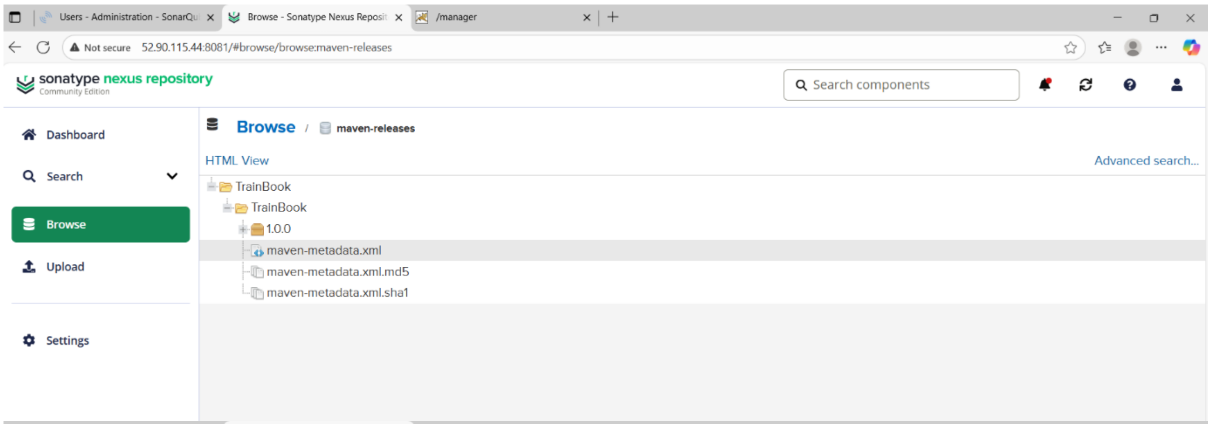
We want to understand what might be preventing you from upgrading to the latest version of Nexus Repository. Your feedback will help us create better resources and tools.

New Formats Supported

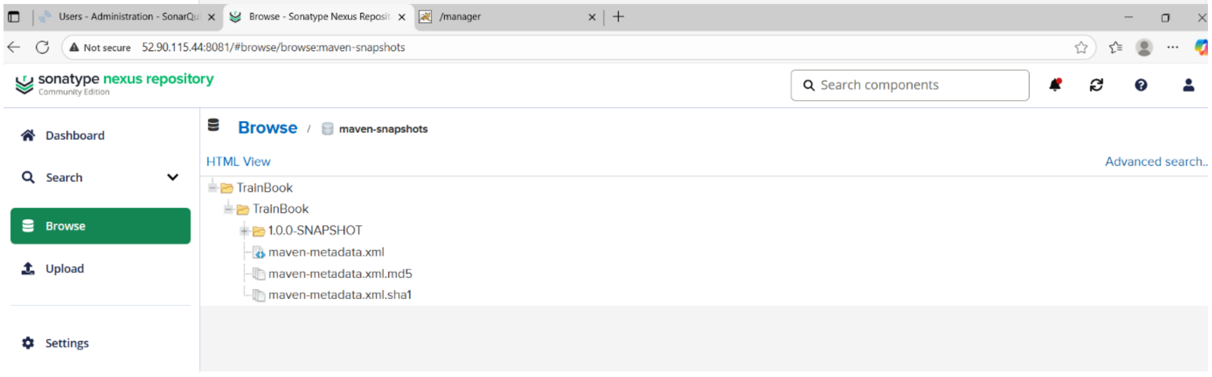


Name ↑	Type	Format	Blob Store	Status	URL	Health check	Firewall Re...
maven-central	proxy	maven2	default	Online - Ready to Co...	copy	0 0	
maven-public	group	maven2	default	Online	copy		
maven-releases	hosted	maven2	default	Online	copy		
maven-snapshots	hosted	maven2	default	Online	copy		
nuget-group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

Snapshot and Release repositories configured in Nexus


Users - Administration - SonarQu
Browse - Sonatype Nexus Reposi
/manager
Not secure 52.90.115.44:8081/#browse/browse:maven-releases
sonatype nexus repository
Community Edition
Search components
Dashboard
Search
Browse
Upload
Settings
Browse / maven-releases
HTML View
Advanced search...

```
TrainBook
  TrainBook
    1.0.0
    maven-metadata.xml
    maven-metadata.xml.md5
    maven-metadata.xml.sha1
```

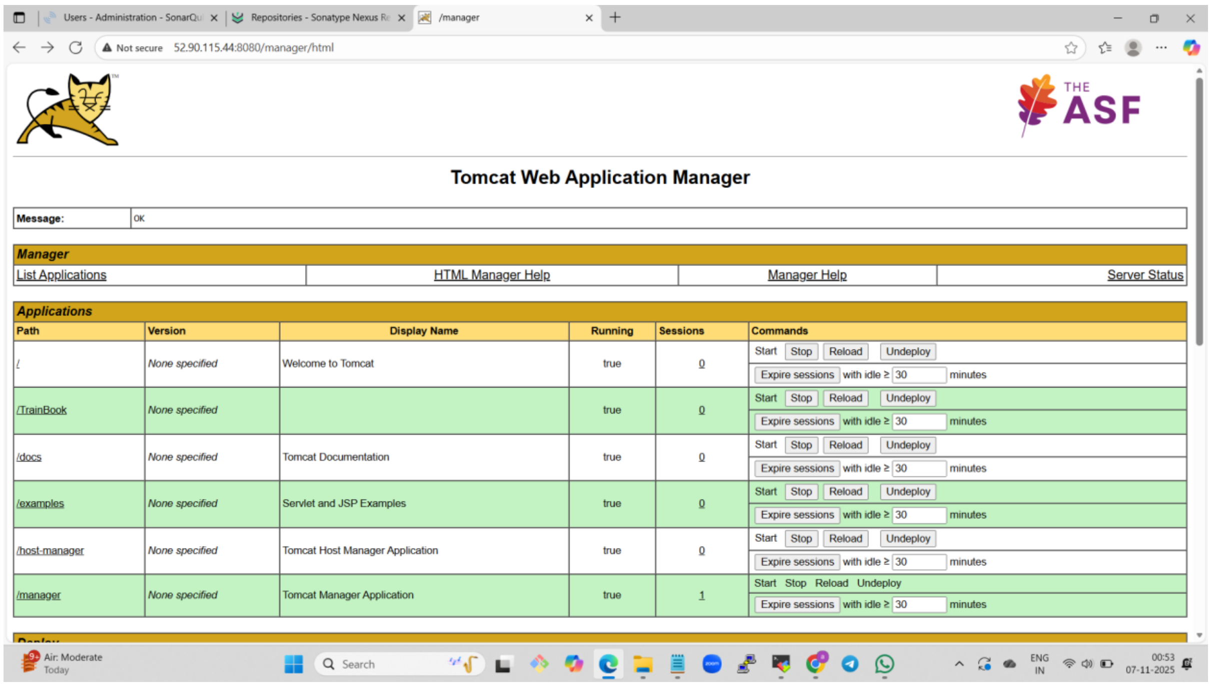

Users - Administration - SonarQu
Browse - Sonatype Nexus Reposi
/manager
Not secure 52.90.115.44:8081/#browse/browse:maven-snapshots
sonatype nexus repository
Community Edition
Search components
Dashboard
Search
Browse
Upload
Settings
Browse / maven-snapshots
HTML View
Advanced search...

```
TrainBook
  TrainBook
    1.0.0-SNAPSHOT
    maven-metadata.xml
    maven-metadata.xml.md5
    maven-metadata.xml.sha1
```

Deployed WAR files stored in Nexus repositories after pipeline execution

Step 8: Deployment to Tomcat Server

The final WAR file is automatically uploaded and deployed to the Tomcat server. The web application is then accessible through the EC2 public IP and port 8080.



Not secure 52.90.115.44:8080/manager/html

Tomcat Web Application Manager

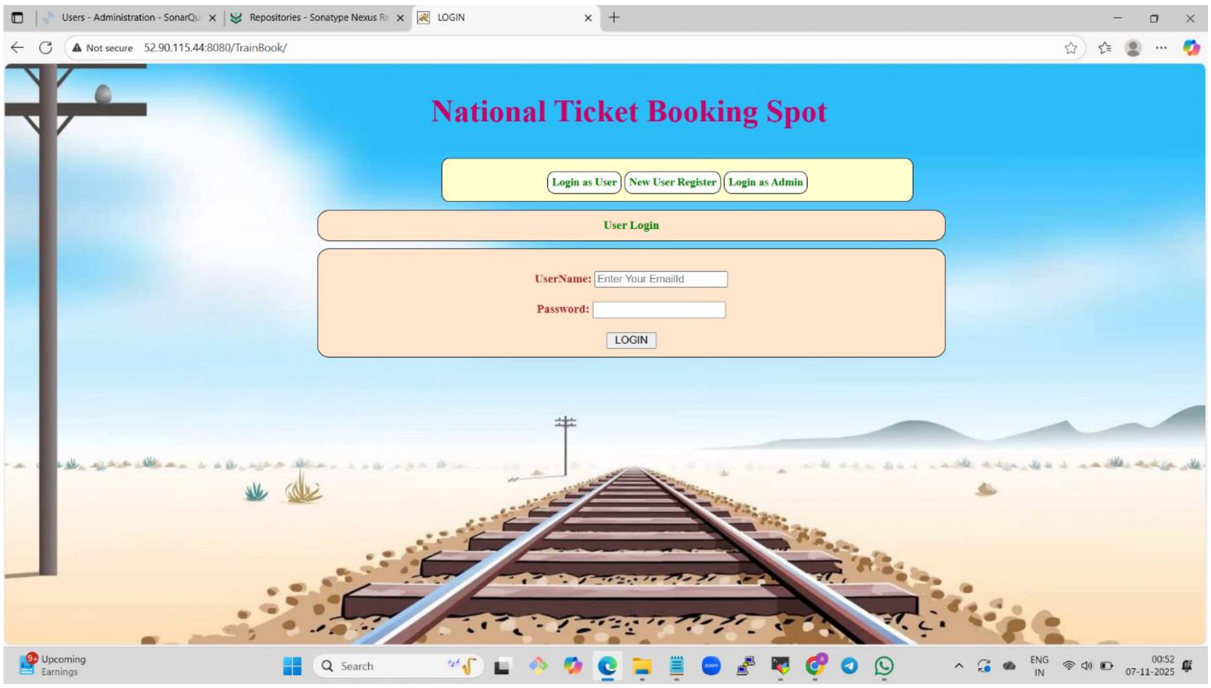
Message:	OK

Manager

List Applications	HTML Manager Help	Manager Help	Server Status

Applications

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

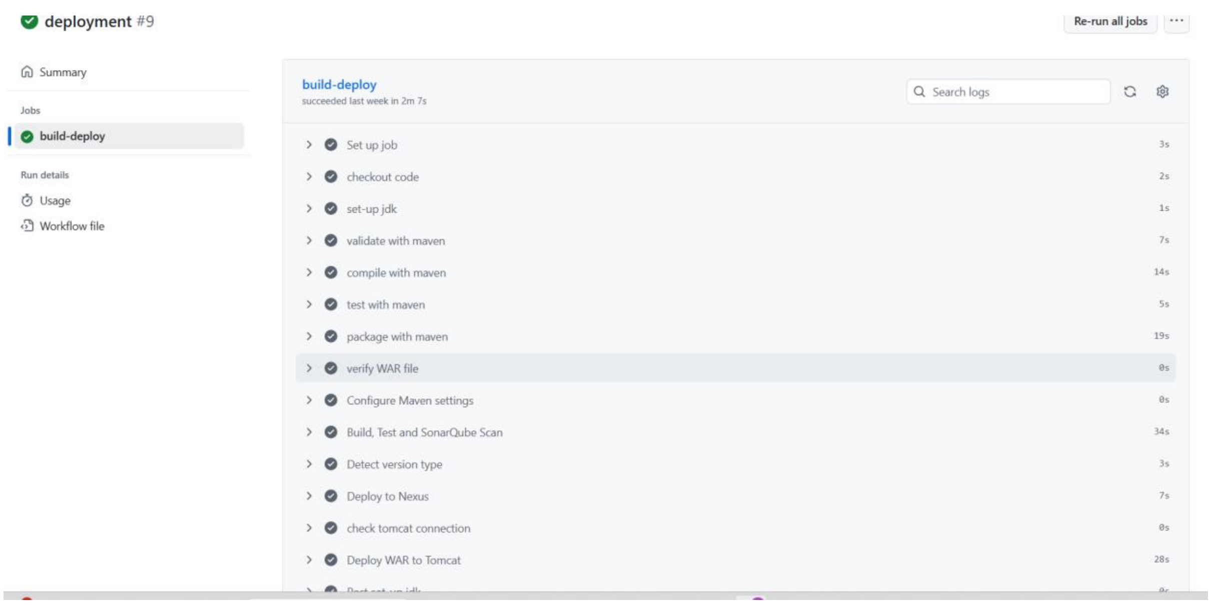


Not secure 52.90.115.44:8080/TrainBook/

Step 9: Final Verification

Verified that the CI/CD pipeline successfully handled all stages — build, code analysis, artifact upload, and deployment.

Confirmed that the web application is accessible and SonarQube/Nexus integrations work properly.


deployment #9

Re-run all jobs

Summary

Jobs

build-deploy

Run details

Usage

Workflow file

build-deploy
succeeded last week in 2m 7s

```
Set up job                        3s
checkout code                     2s
set-up jdk                        1s
validate with maven               7s
compile with maven               14s
test with maven                   5s
package with maven               19s
verify WAR file                   0s
Configure Maven settings          0s
Build, Test and SonarQube Scan   34s
Detect version type               3s
Deploy to Nexus                   7s
check tomcat connection           0s
Deploy WAR to Tomcat             28s
```

GitHub Actions executing the CI/CD pipeline

.github/workflows/deployer.yml

```
name: ci/cd pipeline to deploy to tomcat
on:
push:
branches: master
jobs:
build-deploy:
runs-on: ubuntu-latest
steps:
- name: checkout code
uses: actions/checkout@v3
- name: set-up jdk
uses: actions/setup-java@v3
with:
distribution: 'temurin'
java-version: '17' # Updated Java version
- name: validate with maven
run: mvn validate
- name: compile with maven
run: mvn compile
- name: test with maven
run: mvn test
- name: package with maven
run: mvn clean package
- name: verify WAR file
run: ls -lh target/
- name: Configure Maven settings
run: |
mkdir -p $HOME/.m2
cat > $HOME/.m2/settings.xml <<EOL
<settings>
<servers>
<server>
<id>nexus</id>
<username>${{ secrets.NEXUS_USERNAME }}</username>
<password>${{ secrets.NEXUS_PASSWORD }}</password>
</server>
</servers>
</settings>
EOL
- name: Build, Test and SonarQube Scan
run: |
mvn clean verify sonar:sonar \
-Dsonar.projectKey=train-ticket-reservation \
-Dsonar.host.url="${{ secrets.SONAR_HOST_URL }}" \
-Dsonar.login="${{ secrets.SONAR_TOKEN }}"
- name: Detect version type
id: version_check
run: |
VERSION=$(mvn help:evaluate -Dexpression=project.version -q -DforceStdout)
echo "Project version: $VERSION"
if [[ "$VERSION" == *"-SNAPSHOT" ]]; then
echo "repo_url=${{ secrets.NEXUS_SNAPSHOTS_URL }}" >> $GITHUB_ENV
else
echo "repo_url=${{ secrets.NEXUS_RELEASES_URL }}" >> $GITHUB_ENV
fi
- name: Deploy to Nexus
env:
NEXUS_URL: ${{ env.repo_url }}
run: |
echo "Deploying to $NEXUS_URL"
mvn deploy -DaltDeploymentRepository=nexus::default::${NEXUS_URL}
- name: check tomcat connection
run: |
curl --fail --anyauth -u "${{ secrets.TOMCAT_USER }}:${{ secrets.TOMCAT_PASSWORD }}" \
"http://${{ secrets.TOMCAT_HOST }}:8080/manager/text/list"
# Deploy to Tomcat
- name: Deploy WAR to Tomcat
run: |
WAR_FILE=$(ls target/*.war)
echo "Deploying $WAR_FILE to Tomcat..."
curl -v -u ${{ secrets.TOMCAT_USER }}:${{ secrets.TOMCAT_PASSWORD }} \
-T "$WAR_FILE" \
"http://${{ secrets.TOMCAT_HOST }}:8080/manager/text/deploy?path=/TrainBook&update=true"
```

pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>
<groupId>TrainBook</groupId>
<artifactId>TrainBook</artifactId>
<version>1.0.0</version>
<packaging>war</packaging>
<build>
<sourceDirectory>src</sourceDirectory>
<resources>
<resource>
<directory>src</directory>
<excludes>
<exclude>**/*.java</exclude>
</excludes>
</resource>
</resources>
<plugins>
<plugin>
<artifactId>maven-compiler-plugin</artifactId>
<version>3.8.1</version>
<configuration>
<source>17</source>
<target>17</target>
</configuration>
</plugin>
<plugin>
<artifactId>maven-war-plugin</artifactId>
<version>3.2.3</version>
<configuration>
<warSourceDirectory>WebContent</warSourceDirectory>
</configuration>
</plugin>
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-dependency-plugin</artifactId>
<version>2.3</version>
<executions>
<execution>
<phase>package</phase>
```

```xml
<goals>
<goal>copy</goal>
</goals>
<configuration>
<artifactItems>
<artifactItem>
<groupId>com.github.jsimone</groupId>
<artifactId>webapp-runner</artifactId>
<version>8.0.30.2</version>
<destFileName>webapp-runner.jar</destFileName>
</artifactItem>
</artifactItems>
</configuration>
</execution>
</executions>
</plugin>
<plugin>
<groupId>io.snyk</groupId>
<artifactId>snyk-maven-plugin</artifactId>
<version>2.0.0</version>
<inherited>false</inherited>
<configuration>
<org>Venn1991</org>
</configuration>
</plugin>
</plugins>
</build>
<dependencies>
<dependency>
<groupId>org.postgresql</groupId>
<artifactId>postgresql</artifactId>
<version>42.3.7</version>
</dependency>
<dependency>
<groupId>mysql</groupId>
<artifactId>mysql-connector-java</artifactId>
<version>8.0.28</version>
</dependency>
<dependency>
<groupId>javax.servlet</groupId>
<artifactId>javax.servlet-api</artifactId>
<version>3.1.0</version>
</dependency>
</dependencies>
<properties>
<sonar.host.url>http://52.90.115.44:9000</sonar.host.url>
</properties>
<distributionManagement>
<repository>
<id>nexus</id>
<name>Nexus Release Repository</name>
<url>http://52.90.115.44:8081/repository/maven-releases/</url>
</repository>
<snapshotRepository>
<id>nexus</id>
<name>Nexus Snapshot Repository</name>
<url>http://52.90.115.44:8081/repository/maven-snapshots/</url>
</snapshotRepository>
</distributionManagement>
</project>
```