END TO END CI/CD PIPELINE

(JENKINS + SONARQUBE + NEXUS + TOMCAT)

Project Documentation

Prepared By: Jenifa R

Date: 17 January 2026

❖ PROJECT DETAILS

Item	Details
Project Title	End to End CI/CD Pipeline (Jenkins+ SonarQube +Nexus +Tomcat)
Domains	DevOps / CI-CD Automation
Application Type	Java Maven Application(.War)
Source Code Repository	https://github.com/suffixscope/maven-web-app
Deployment Target	Apache Tomcat (War Deployment)
Cloud Platform	AWS EC2
Pipeline Tool	Jenkins
Code Quality	SonarQube
Artifact Repository	Nexus Repository Manager

Tool & version

Tool	Version

Jenkins	2.528.3
Java	17.0.17
SonarQube	9.9.8.100196
Maven	3.8.7
Tomcat	10.1.50
Nexus	3.88.0-08

❖ OBJECTIVE

Implement a complete CI/CD workflow for a Maven-based Java web application to:

- Automatically pull source code from GitHub
- Build and package the project using Maven
- Perform static code analysis with SonarQube
- Store the generated .war artifact in Nexus Repository Manager
- Deploy the application seamlessly to Apache Tomcat
- Verify successful deployment by accessing the application in a browser

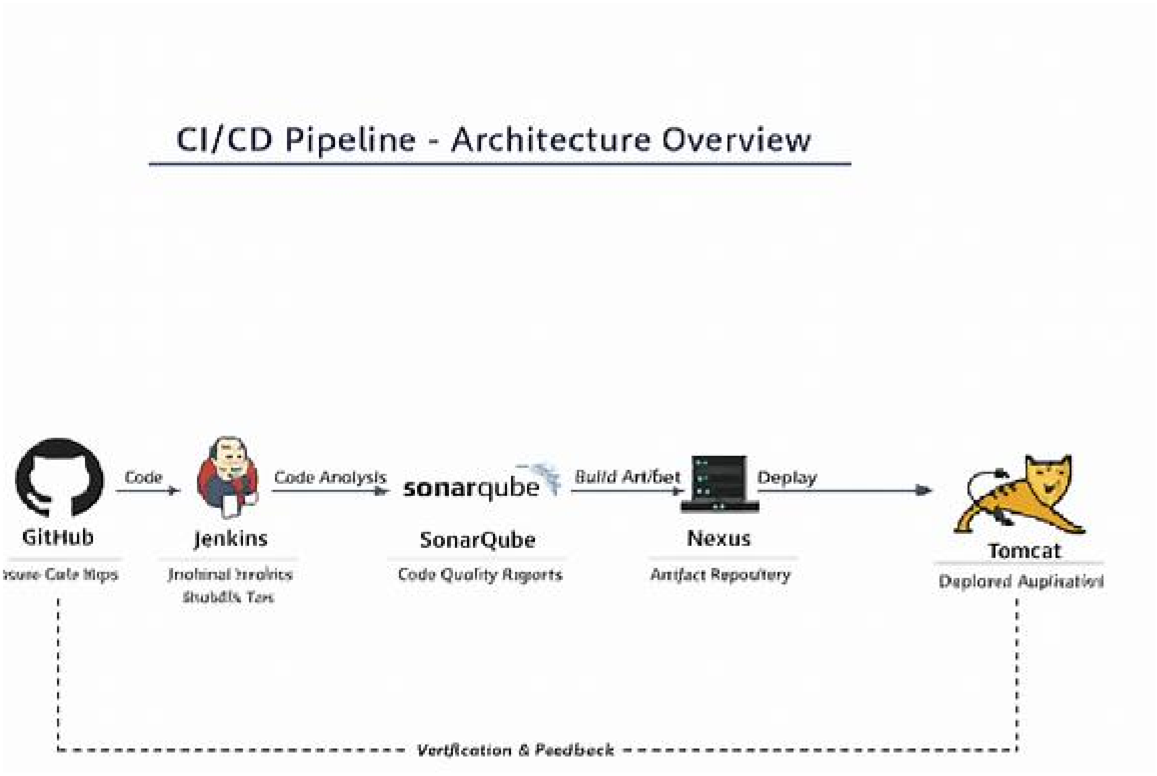
❖ HIGH-LEVEL-ARCHITECTURE

Flow: GitHub → Jenkins Pipeline → (Build/Test) → SonarQube Analysis → Nexus Upload → Tomcat Deployment → Verification

Components:

- GitHub: Stores and manages the source code repository
- Jenkins: Orchestrates the CI/CD pipeline and automates build/test stages
- SonarQube: Performs static code analysis and generates quality reports
- Nexus Repository Manager: Stores and versions the build artifacts (.war)
- Apache Tomcat: Hosts and runs the deployed web application
- Browser: Used to validate successful deployment by accessing the application

Architecture Diagram



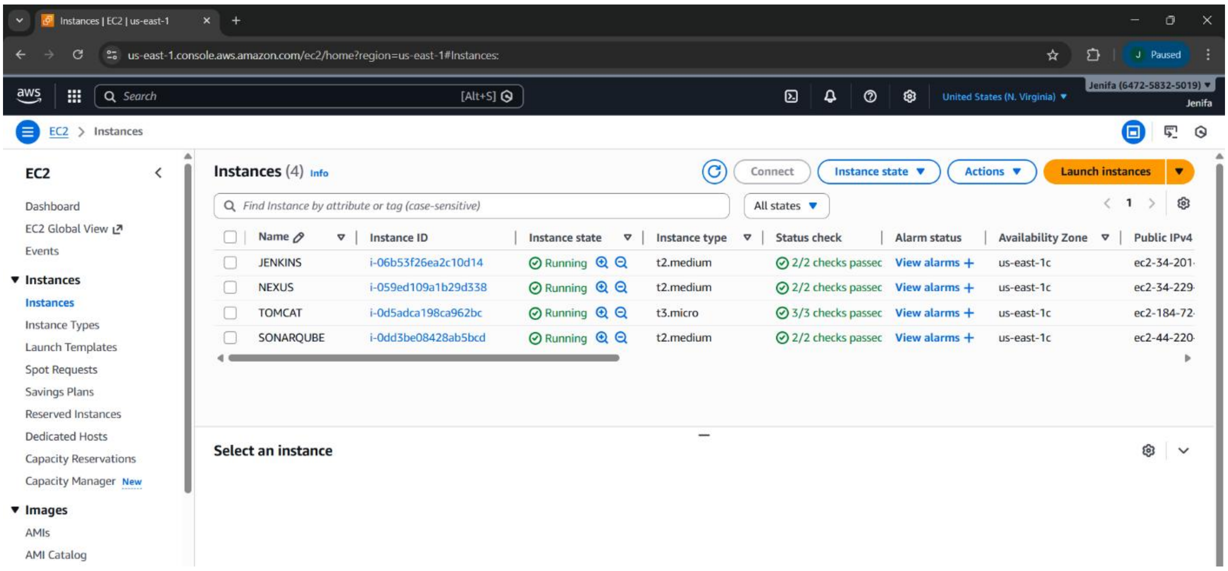
❖ INFRASTRUCTURE SETUP-AWS EC2 ARCHITECTURE

Operating System: Amazon Linux AMI

Service	EC2 Instance	Purpose	Default Port
Jenkins	Jenkins	CI/CD pipeline execution	8080
SonarQube	sonar	Code quality analysis	9000
Nexus	nexus	Artifact repository	8081
Tomcat	tomcat	Application deployment	8080

Security Group – Inbound Ports:

- 22 (SSH): Restrict to your IP only
- 8080: Jenkins / Tomcat
- 9000: SonarQube
- 8081: Nexus



Instances (4) Info

Find Instance by attribute or tag (case-sensitive)

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
□	JENKINS	i-06b53f26ea2c10d14	✓ Running	t2.medium	✓ 2/2 checks passec	View alarms +	us-east-1c	ec2-34-201-
□	NEXUS	i-059ed109a1b29d338	✓ Running	t2.medium	✓ 2/2 checks passec	View alarms +	us-east-1c	ec2-34-229-
□	TOMCAT	i-0d5adca198ca962bc	✓ Running	t3.micro	✓ 3/3 checks passec	View alarms +	us-east-1c	ec2-184-72-
□	SONARQUBE	i-0dd3be08428ab5bcd	✓ Running	t2.medium	✓ 2/2 checks passec	View alarms +	us-east-1c	ec2-44-220-

Select an instance

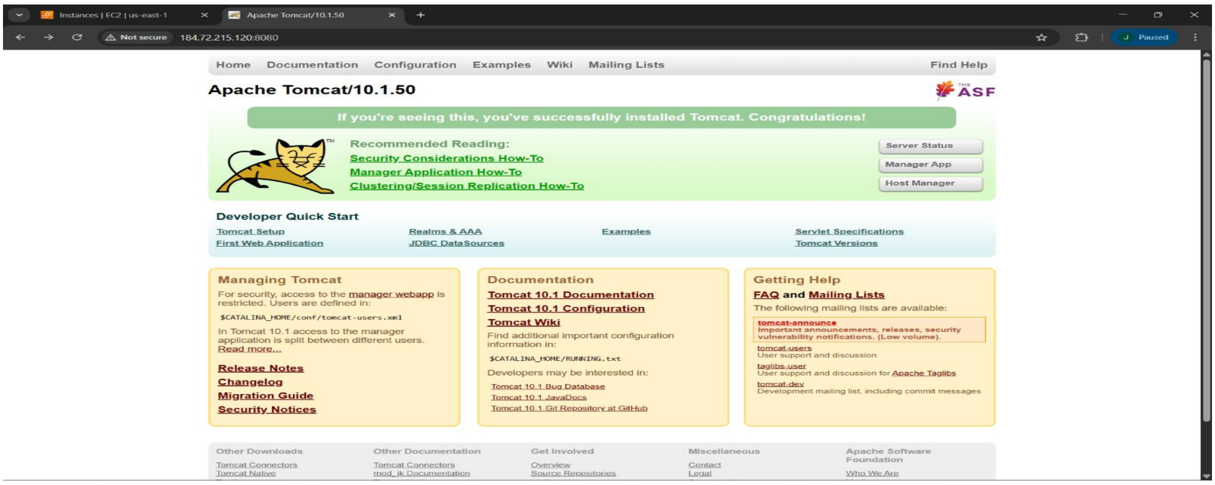
❖ TOMCAT SETUP

Tomcat Installation:

- Tomcat was installed on the dedicated EC2 instance (tomcat).
- The server was started manually using the extracted binaries and the startup.sh script.

Verification:

Tomcat UI was successfully accessed at http://184.72.215.120:8080/



Apache Tomcat/10.1.50

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:
Security Considerations How-To
Manager Application How-To
Clustering/Session Replication How-To

Developer Quick Start
Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Examples
Servlet Specifications
Tomcat Versions

Managing Tomcat
For security, access to the manager webapp is restricted. Users are defined in:

```
$CATALINA_HOME/conf/tomcat-users.xml
```

In Tomcat 10.1 access to the manager application is split between different users. Read more...

Release Notes
Changelog
Migration Guide
Security Notices

Documentation
Tomcat 10.1 Documentation
Tomcat 10.1 Configuration
Tomcat Wiki
Find additional important configuration information in:

```
$CATALINA_HOME/RUNNING.txt
```

Developers may be interested in:
Tomcat 10.1 Bug Database
Tomcat 10.1 JavaDocs
Tomcat 10.1 Git Repository at GitHub

Getting Help
FAQ and Mailing Lists
The following mailing lists are available:
tomcat-announce
Important announcements, releases, security vulnerability notifications. (Low volume).
tomcat-users
User support and discussion
taglibs-user
User support and discussion for Apache Taglibs
tomcat-dev
Development mailing list, including commit messages

Other Downloads
Tomcat Connectors
Tomcat Native

Other Documentation
Tomcat Connectors
mod_jk Documentation

Get Involved
Overview
Source Repositories

Miscellaneous
Contact
Legal

Apache Software Foundation
Who We Are

❖ SONARQUBE SETUP

SonarQube Installation:

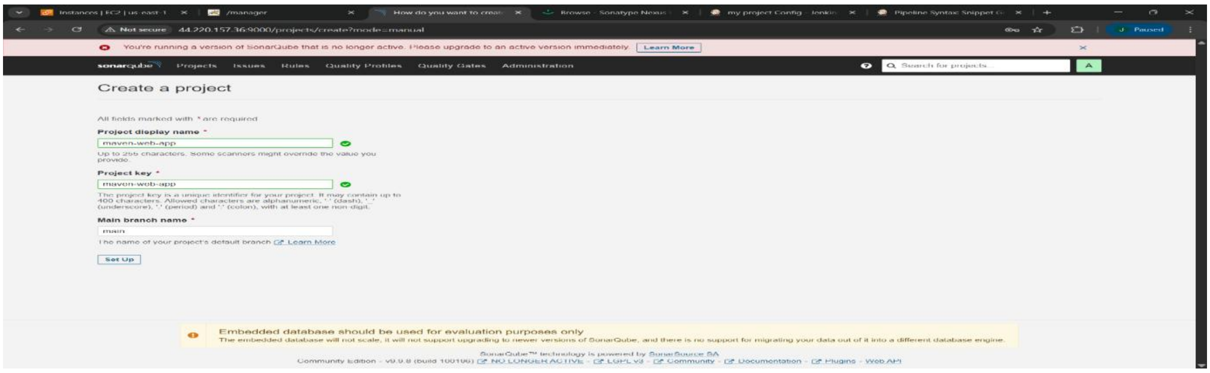
SonarQube was installed and started on the dedicated EC2 instance (sonar).

Access URL:

http://44.220.157.36:9000/

Configuration Steps:

- Logged into the SonarQube UI
- Created project: maven-web-app



You're running a version of SonarQube that is no longer active. Please upgrade to an active version immediately. Learn More

Create a project

All fields marked with * are required

Project display name *: maven-web-app

Up to 255 characters. Some scanners might override the value you provide.

Project key *: maven-web-app

The project key is a unique identifier for your project. It may contain up to 400 characters. Allowed characters are alphanumeric, '-' (dash), '_' (underscore), '.' (period) and ':' (colon), with at least one non-digit.

Main branch name *: main

The name of your project's default branch Learn More

Set Up

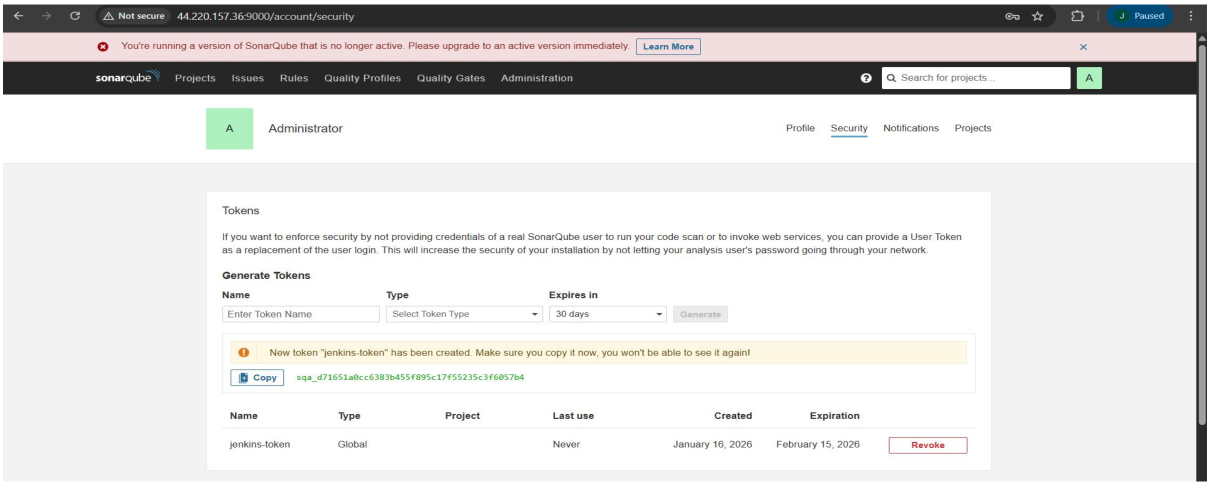
Embedded database should be used for evaluation purposes only

The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

SonarQube™ technology is powered by SonarSource SA

Community Edition - v9.9.8 (build 100196) NO LONGER ACTIVE - LGPL v3 - Community - Documentation - Plugins - Web API

- Generated a token/secret for Jenkins integration(Stored the token securely in Jenkins credentials)



You're running a version of SonarQube that is no longer active. Please upgrade to an active version immediately. Learn More

Tokens

If you want to enforce security by not providing credentials of a real SonarQube user to run your code scan or to invoke web services, you can provide a User Token as a replacement of the user login. This will increase the security of your installation by not letting your analysis user's password going through your network.

Generate Tokens

Name	Type	Expires in	
Enter Token Name	Select Token Type	30 days	Generate

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

Copy `sqa_d71651a0cc6383b455f895c17f55235c3f6057b4`

Name	Type	Project	Last use	Created	Expiration	
jenkins-token	Global		Never	January 16, 2026	February 15, 2026	Revoke

❖ NEXUS SETUP

Nexus Installation:

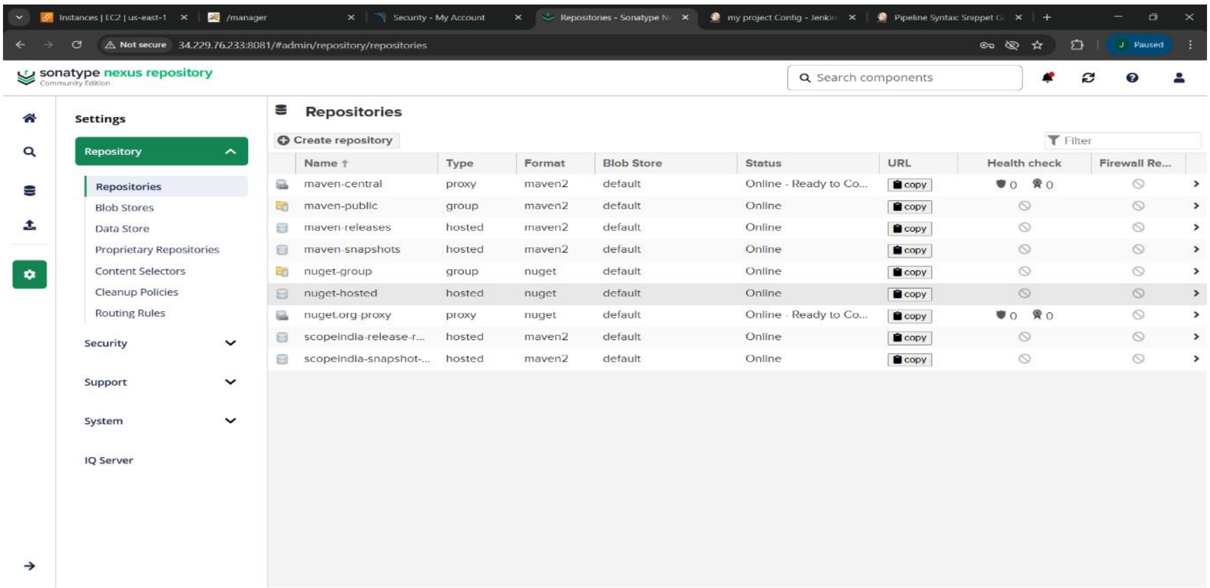
Nexus Repository Manager was installed and started on the dedicated EC2 instance (nexus).

Access URL:

http://34.229.76.233:8081/

Configuration Steps:

- Logged into the Nexus UI
- Created Maven (hosted) repositories:
  - scopeindia-snapshot-repository → for snapshot builds
  - scopeindia-release-repository → for release builds
- Confirmed repositories are available for publishing artifacts from Jenkins



Repositories

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	⦸
scopeindia-release-r...	hosted	maven2	default	Online	copy	⦸	⦸
scopeindia-snapshot-...	hosted	maven2	default	Online	copy	⦸	⦸

❖ JENKINS SETUP

Jenkins Installation:

Jenkins was installed and started on the dedicated EC2 instance (Jenkins).

Access URL:

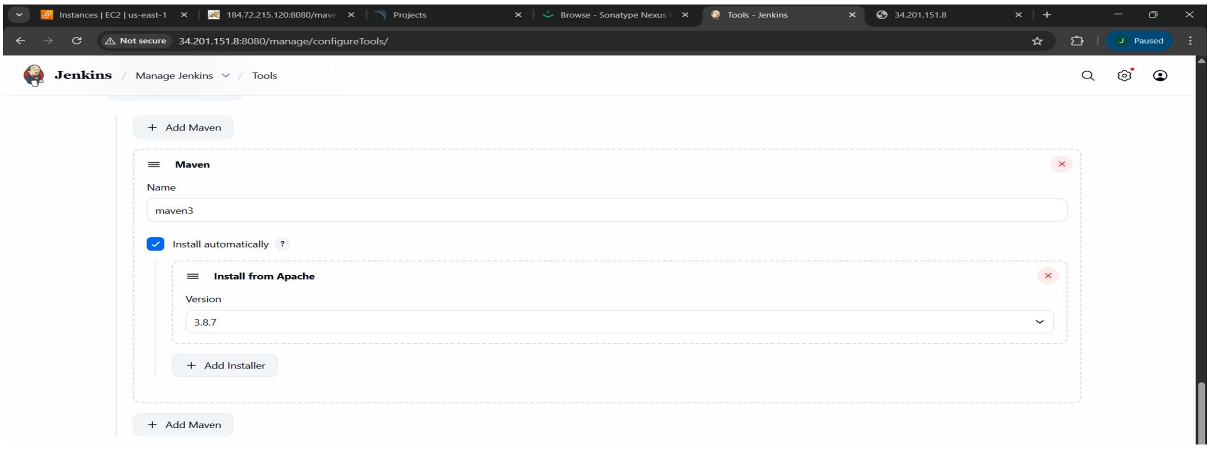
http://34.201.151.8:8080/

Plugins Used in This Project:

- SonarQube Scanner for Jenkins → integrates SonarQube analysis into the pipeline
- Nexus Artifact Uploader → publishes build artifacts to Nexus repositories
- SSH Agent → enables secure deployment to remote servers (e.g., Tomcat)

⇒ Global Tool Configuration:

- Navigate to: Manage Jenkins → Global Tool Configuration
- Configured Maven installation with the name: Maven3



34.201.151.8:8080/manage/configureTools/

Maven

Name: maven3

☑ Install automatically

Install from Apache

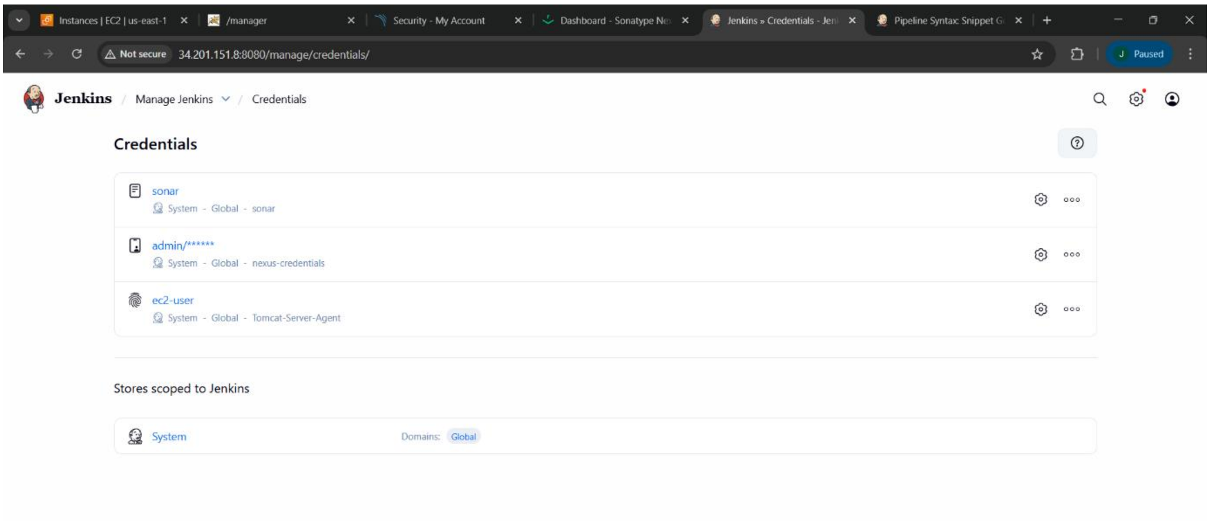
Version: 3.8.7

⇒ Credentials Stored in Jenkins:

- Nexus credentials → credentials Id: nexus-credentials
- Tomcat SSH key → credentials Id: Tomcat-Server-Agent
- SonarQube token → credentials Id: sonar

Purpose:

- Nexus credentials: Used by Jenkins to upload artifacts (.war) into Nexus repositories.
- Tomcat SSH key: Enables Jenkins to securely connect and deploy applications to the Tomcat server.
- SonarQube token: Allows Jenkins to authenticate with SonarQube for static code analysis.


Instances | EC2 | us-east-1
/manager
Security - My Account
Dashboard - Sonatype Ne
Jenkins » Credentials - Jen
Pipeline Syntax: Snippet G
Not secure
34.201.151.8:8080/manage/credentials/
Paused
Jenkins
Manage Jenkins
Credentials

Credentials

sonar
System - Global - sonar

admin/******
System - Global - nexus-credentials

ec2-user
System - Global - Tomcat-Server-Agent

Stores scoped to Jenkins

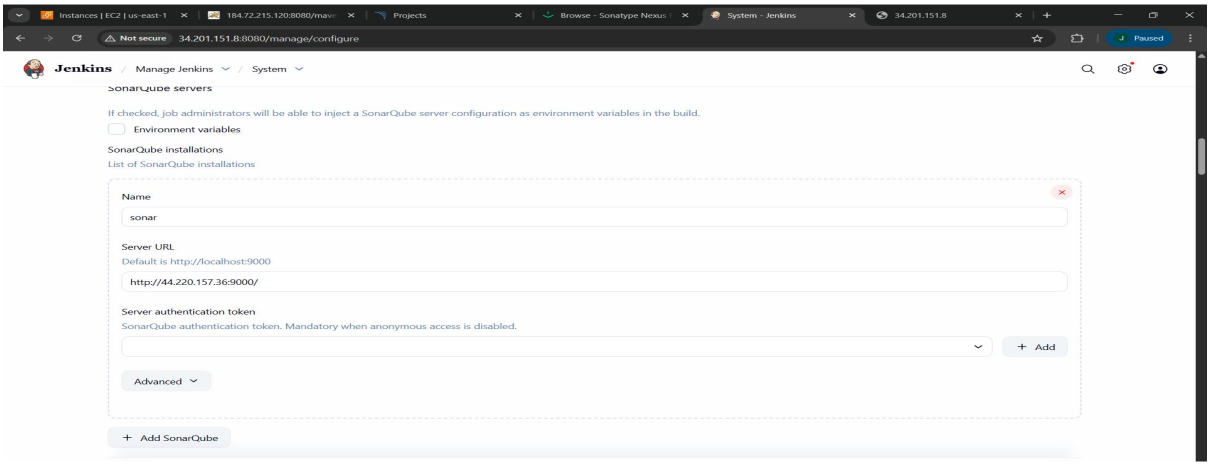
System Domains: Global

⇒ SonarQube Integration in Jenkins

Configuration Steps:

- Navigate to: Manage Jenkins → System → SonarQube Servers
- Configured server name: sonar

- **Added SonarQube server URL and token credential (previously created in SonarQube and stored in Jenkins credentials)**



Jenkins / Manage Jenkins / System

34.201.151.8:8080/manage/configure

SonarQube servers

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

□ Environment variables

SonarQube installations

List of SonarQube installations

Name: sonar

Server URL

Default is http://localhost:9000

http://44.220.157.36:9000/

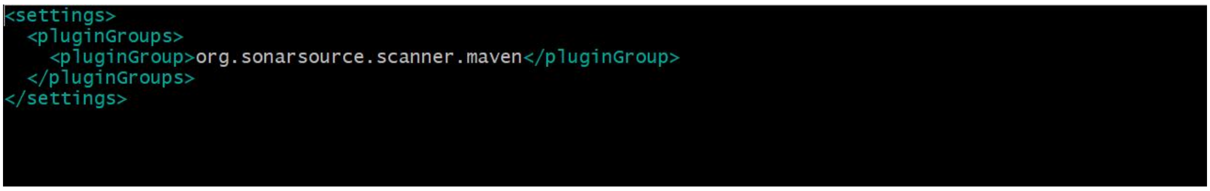
Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

⇒ Maven Settings Requirement:

- **For mvn sonar:sonar to work in the pipeline, you need a proper Maven settings file.**
- **Path:** /var/lib/jenkins/.m2/settings.xml

```
<settings>
   <pluginGroups>
      <pluginGroup>org.sonarsource.scanner.maven</pluginGroup>
  </pluginGroups>
</settings>
```



```
<settings>
  <pluginGroups>
    <pluginGroup>org.sonarsource.scanner.maven</pluginGroup>
  </pluginGroups>
</settings>
```

Alternative SonarQube Command (No settings.xml needed)

Run sh '${mvn} -B org.sonarsource.scanner.maven:sonar-maven-plugin: 5.5.0.6356:sonar'

- ${mvn} → refers to your Maven tool configured in Jenkins (e.g., Maven3).
- -B → batch mode (non-interactive, CI-friendly).

The plugin is called directly, so the settings.xml entry isn’t required.

⇒ Nexus Integration

- Configured in Jenkins pipeline using the Nexus Artifact Uploader plugin.
- Purpose: Upload build artifacts (.war, .jar) to Nexus repository for versioned storage.

⇒ Tomcat Integration

- Deployment handled via Jenkins pipeline using SSH Agent + SCP.
- Purpose: Securely copy .war files from Jenkins workspace to the Tomcat server for deployment.

❖ CI/CD Pipeline Implementation (Jenkins)

Pipeline Stages

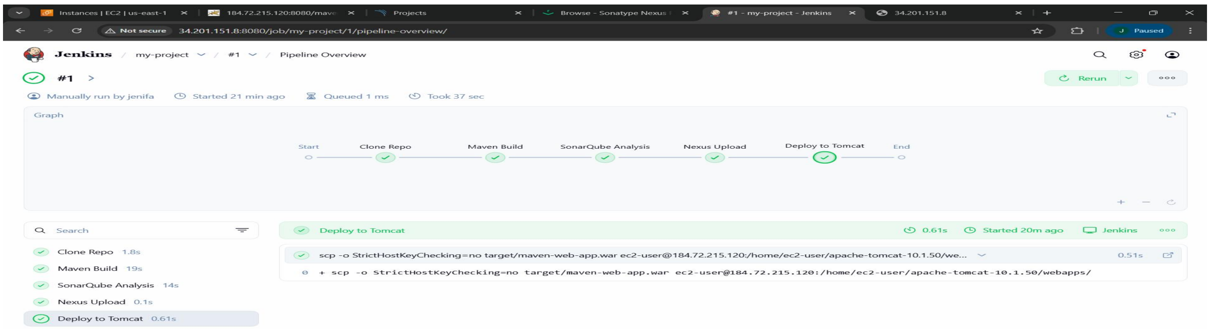
Stage	What it does	Output / Result
Clone Repo	Clones the master branch from GitHub into Jenkins workspace	Source code available in workspace

Stage	What it does	Output / Result
Maven Build	Runs mvn -B clean package using Maven3	WAR generated: target/maven-web-app.war
SonarQube Analysis	Runs mvn sonar:sonar (or direct plugin call without settings.xml) and publishes report to SonarQube	Latest analysis visible in SonarQube
Nexus Upload	Uploads generated WAR/JAR to Nexus repository using Nexus Artifact Uploader plugin	Artifact stored and versioned in Nexus
Tomcat Deployment	Deploys WAR file to Tomcat server using SSH Agent + SCP	Application deployed and accessible via Tomcat server

❖ FULL JENKINS PIPELINE SCRIPT (SCRIPTED PIPELINE)

```
node {
    def MAVEN_HOME = tool name: 'maven3', type: 'maven'
    def MAVEN_CMD = "${MAVEN_HOME}/bin/mvn"

    try {
        stage('Clone Repo') {
            git branch: 'master',
                url: 'https://github.com/suffixscope/maven-web-app.git'
        }

        stage('Maven Build') {
            sh "${MAVEN_CMD} -Dmaven.repo.local='${WORKSPACE}/.m2/repo' -B clean package"
        }

        stage('SonarQube Analysis') {
            withSonarQubeEnv('sonar') {
                withCredentials([string(credentialsId: 'sonar-token', variable: 'SONAR_TOKEN')]) {
                    sh "${MAVEN_CMD} -Dmaven.repo.local='${WORKSPACE}/.m2/repo' sonar:sonar -Dsonar.login=$SONAR_TOKEN"
                }
            }
        }

        stage('Nexus Upload') {
            nexusArtifactUploader(
                artifacts: [[
                    artifactId: 'maven-web-app',
                    classifier: '',
                    file: 'target/maven-web-app.war',
                    type: 'war'
                ]],
                credentialsId: 'nexus-credentials',
                groupId: 'org.scopeindia',
                nexusUrl: '34.229.76.233:8081',
                nexusVersion: 'nexus3',
                protocol: 'http',
                repository: 'scopeindia-release-repository',
                version: '1.2-RELEASE'
            )
        }

        stage('Deploy to Tomcat') {
            sshagent(['Tomcat-Server-Agent']) {
                sh '''
                scp -o StrictHostKeyChecking=no \
                target/maven-web-app.war \
                ec2-user@184.72.215.120:/home/ec2-user/apache-tomcat-10.1.50/webapps/
                '''
            }
        }

        echo "Pipeline executed successfully!"
    } catch (err) {
        echo "Pipeline failed. Check logs!"
        throw err
    }
}
```

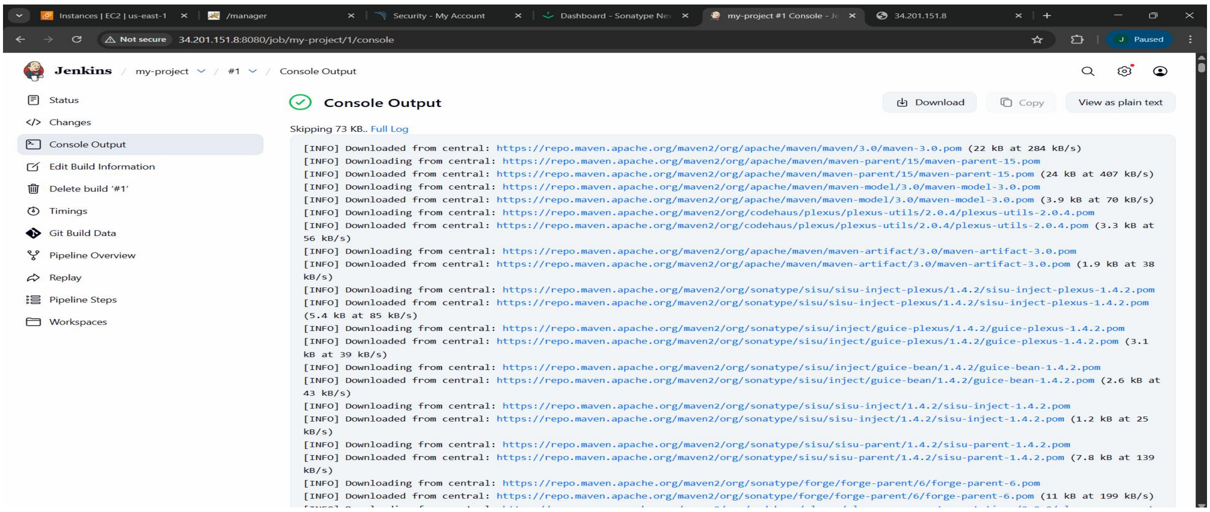
```
<settings>
  <pluginGroups>
    <pluginGroup>org.sonarsource.scanner.maven</pluginGroup>
  </pluginGroups>
</settings>
```



Clone Repo 1.8s
Maven Build 19s
SonarQube Analysis 14s
Nexus Upload 0.1s
Deploy to Tomcat 0.61s

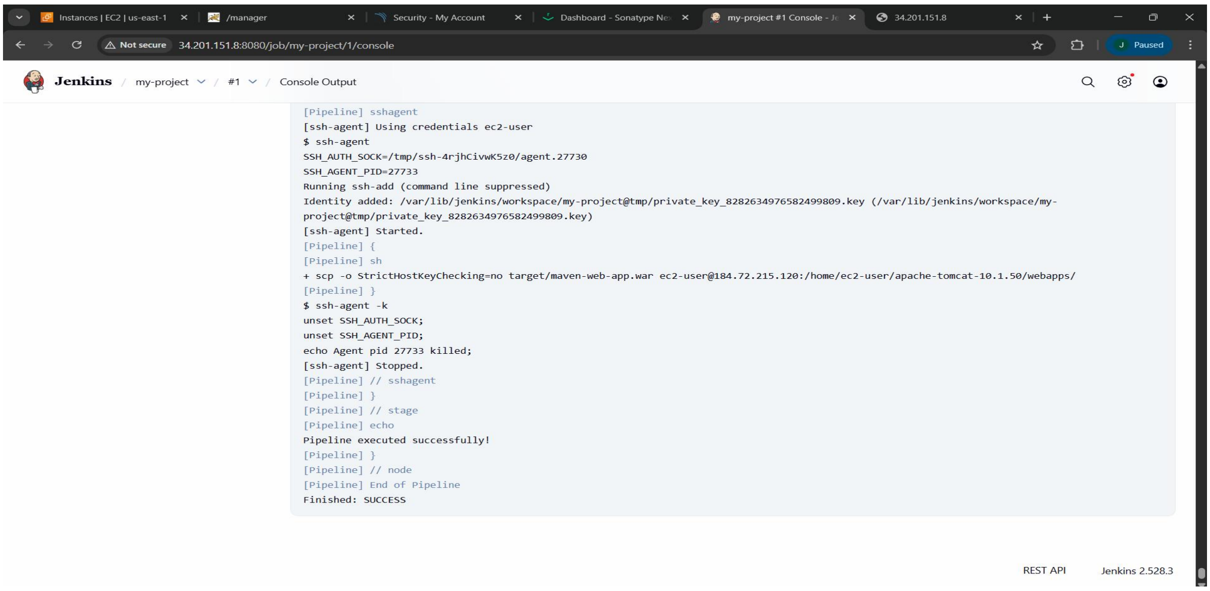
Deploy to Tomcat 0.61s Started 20m ago Jenkins

```
scp -o StrictHostKeyChecking=no target/maven-web-app.war ec2-user@184.72.215.120:/home/ec2-user/apache-tomcat-10.1.50/we...   0.51s
0  + scp -o StrictHostKeyChecking=no target/maven-web-app.war ec2-user@184.72.215.120:/home/ec2-user/apache-tomcat-10.1.50/webapps/
```



Skipping 73 KB.. Full Log

```
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven/3.0/maven-3.0.pom (22 kB at 284 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/15/maven-parent-15.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/15/maven-parent-15.pom (24 kB at 407 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-model/3.0/maven-model-3.0.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-model/3.0/maven-model-3.0.pom (3.9 kB at 70 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/2.0.4/plexus-utils-2.0.4.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/2.0.4/plexus-utils-2.0.4.pom (3.3 kB at
56 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-artifact/3.0/maven-artifact-3.0.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-artifact/3.0/maven-artifact-3.0.pom (1.9 kB at 38
kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/sonatype/sisu/sisu-inject-plexus/1.4.2/sisu-inject-plexus-1.4.2.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/sonatype/sisu/sisu-inject-plexus/1.4.2/sisu-inject-plexus-1.4.2.pom
(5.4 kB at 85 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/sonatype/sisu/inject/guice-plexus/1.4.2/guice-plexus-1.4.2.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/sonatype/sisu/inject/guice-plexus/1.4.2/guice-plexus-1.4.2.pom (3.1
kB at 39 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/sonatype/sisu/inject/guice-bean/1.4.2/guice-bean-1.4.2.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/sonatype/sisu/inject/guice-bean/1.4.2/guice-bean-1.4.2.pom (2.6 kB at
43 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/sonatype/sisu/sisu-inject/1.4.2/sisu-inject-1.4.2.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/sonatype/sisu/sisu-inject/1.4.2/sisu-inject-1.4.2.pom (1.2 kB at 25
kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/sonatype/sisu/sisu-parent/1.4.2/sisu-parent-1.4.2.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/sonatype/sisu/sisu-parent/1.4.2/sisu-parent-1.4.2.pom (7.8 kB at 139
kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/sonatype/forge/forge-parent/6/forge-parent-6.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/sonatype/forge/forge-parent/6/forge-parent-6.pom (11 kB at 199 kB/s)
```



Instances | EC2 | us-east-1
/manager
Security - My Account
Dashboard - Sonatype Nex
my-project #1 Console - Je
34.201.151.8
Not secure
34.201.151.8:8080/job/my-project/1/console
Paused
Jenkins / my-project / #1 / Console Output
REST API
Jenkins 2.528.3

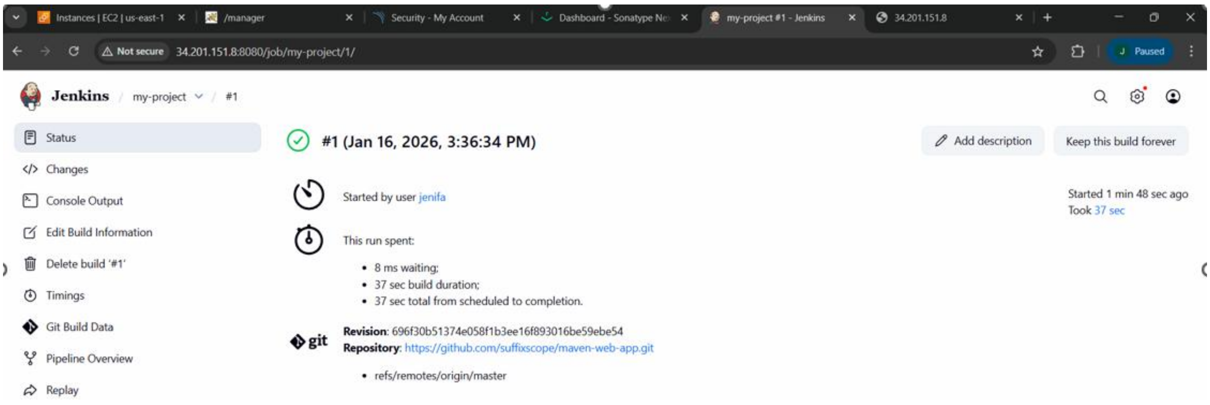
```
[Pipeline] sshagent
[ssh-agent] Using credentials ec2-user
$ ssh-agent
SSH_AUTH_SOCK=/tmp/ssh-4rjhCivwK5z0/agent.27730
SSH_AGENT_PID=27733
Running ssh-add (command line suppressed)
Identity added: /var/lib/jenkins/workspace/my-project@tmp/private_key_8282634976582499809.key (/var/lib/jenkins/workspace/my-
project@tmp/private_key_8282634976582499809.key)
[ssh-agent] Started.
[Pipeline] {
[Pipeline] sh
+ scp -o StrictHostKeyChecking=no target/maven-web-app.war ec2-user@184.72.215.120:/home/ec2-user/apache-tomcat-10.1.50/webapps/
[Pipeline] }
$ ssh-agent -k
unset SSH_AUTH_SOCK;
unset SSH_AGENT_PID;
echo Agent pid 27733 killed;
[ssh-agent] Stopped.
[Pipeline] // sshagent
[Pipeline] }
[Pipeline] // stage
[Pipeline] echo
Pipeline executed successfully!
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

❖ TESTING & VERIFICATION

1. Jenkins Working

Jenkins UI accessible at: http://13.233.207.42:8080

Pipeline Status: Runs successfully without failure



34.201.151.8:8080/job/my-project/1/

Jenkins / my-project / #1

#1 (Jan 16, 2026, 3:36:34 PM)

Started by user jenifa

Started 1 min 48 sec ago
Took 37 sec

This run spent:

- 8 ms waiting;
- 37 sec build duration;
- 37 sec total from scheduled to completion.

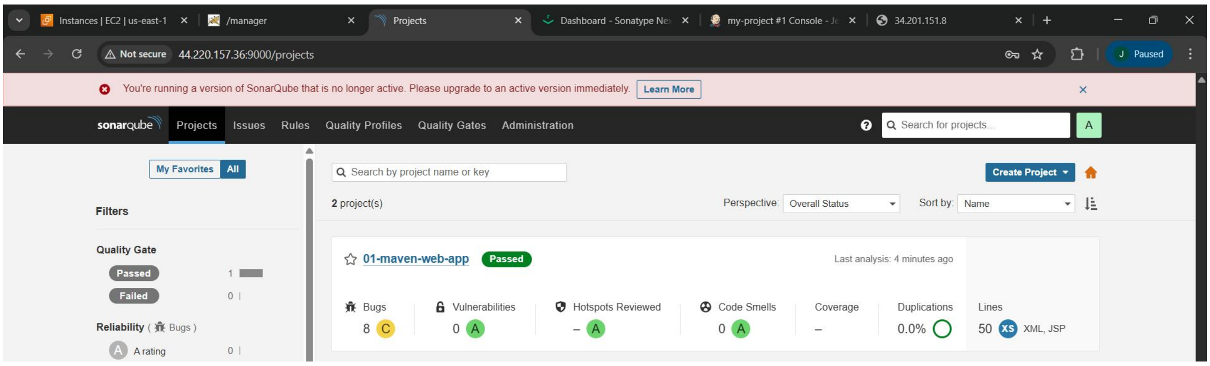
Revision: 696f30b51374e058f1b3ee16f893016be59ebe54
Repository: https://github.com/suffixscope/maven-web-app.git

- refs/remotes/origin/master

2. SonarQube Report Generated

SonarQube project shows analysis and quality gate status

SonarQube URL: http://44.220.157.36:9000/



44.220.157.36:9000/projects

You're running a version of SonarQube that is no longer active. Please upgrade to an active version immediately.

2 project(s)

01-maven-web-app Passed

Last analysis: 4 minutes ago

Bugs	Vulnerabilities	Hotspots Reviewed	Code Smells	Coverage	Duplications	Lines
8 C	0 A	– A	0 A	–	0.0%	50 XS XML, JSP

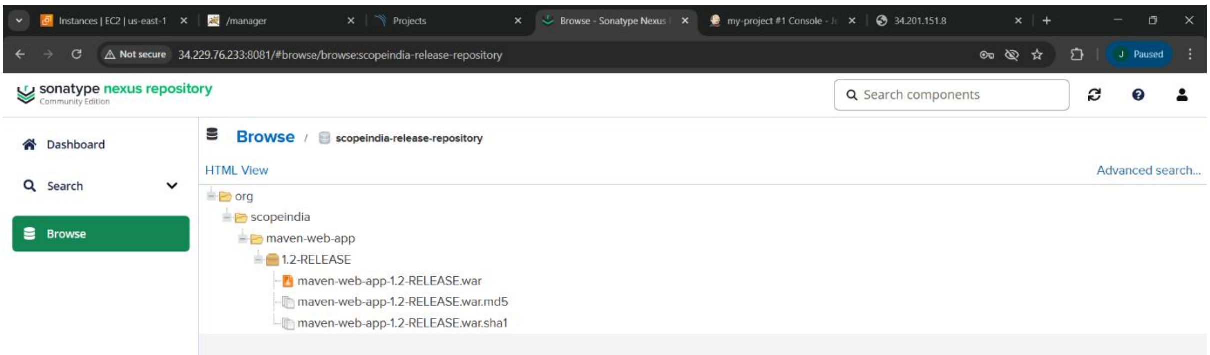
Quality Gate: Passed 1, Failed 0

Reliability (Bugs): A rating 0

3. Artifact Stored in Nexus

WAR artifact Visibile under scopeindia-release-repository with version 1.2 release.

Nexus URL: http://34.229.76.233:8081/

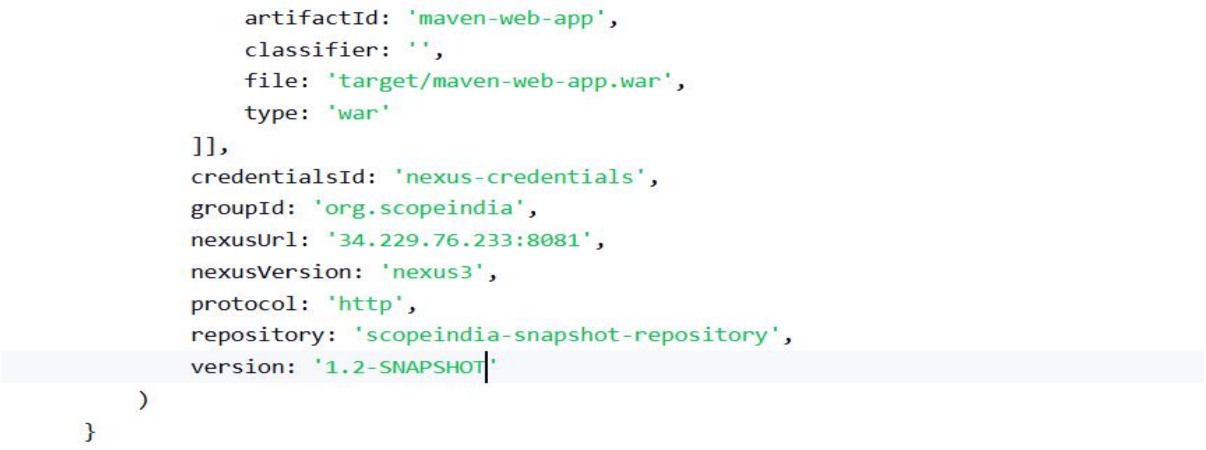


```
org
  scopeindia
    maven-web-app
      1.2-RELEASE
        maven-web-app-1.2-RELEASE.war
        maven-web-app-1.2-RELEASE.war.md5
        maven-web-app-1.2-RELEASE.war.sha1
```

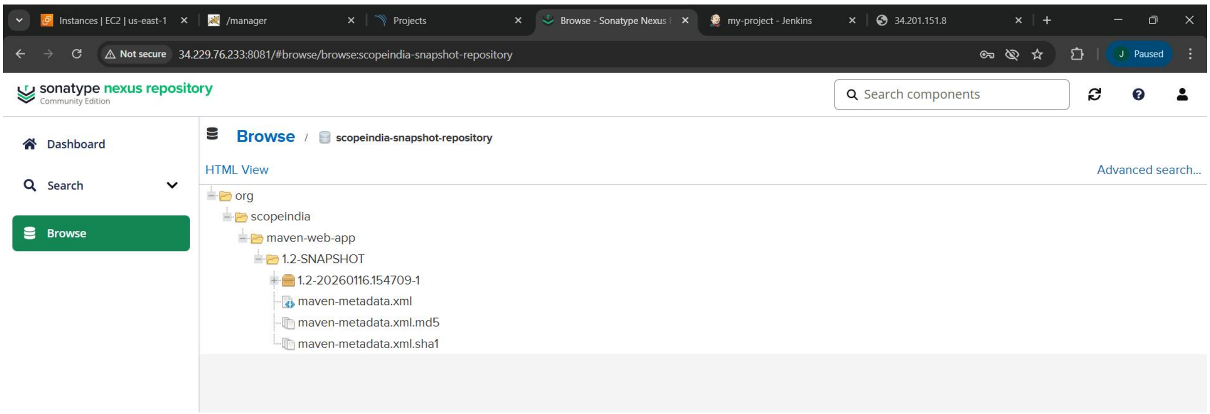
Nexus Upload: Snapshot vs Release (pipeline change):

Currently the pipeline uploads the artifact to the release repository. To upload to the Snapshot update the Nexus upload stage values: repository and version.

Upload to the snapshot repository (change these files)



```
            artifactId: 'maven-web-app',
            classifier: '',
            file: 'target/maven-web-app.war',
            type: 'war'
        ]],
        credentialsId: 'nexus-credentials',
        groupId: 'org.scopeindia',
        nexusUrl: '34.229.76.233:8081',
        nexusVersion: 'nexus3',
        protocol: 'http',
        repository: 'scopeindia-snapshot-repository',
        version: '1.2-SNAPSHOT'
    )
}
```

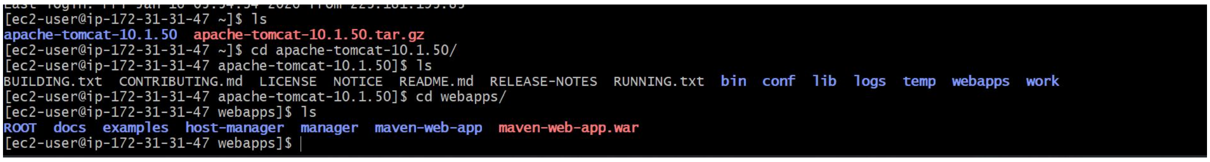


```
org
  scopeindia
    maven-web-app
      1.2-SNAPSHOT
        1.2-20260116.154709-1
        maven-metadata.xml
        maven-metadata.xml.md5
        maven-metadata.xml.sha1
```

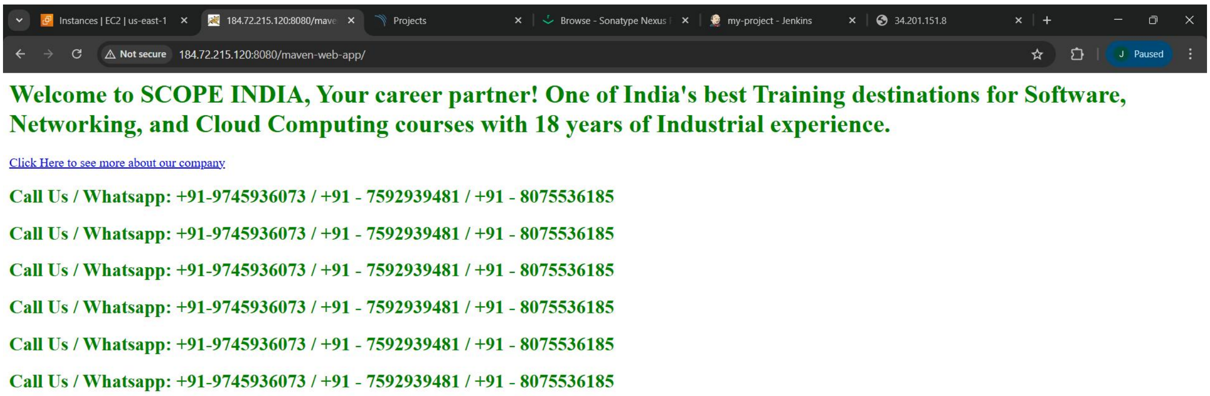
4. Application Deployed on Tomcat

WAR deployed to Tomcat webapps/ and application via browser.

Tomcat URL: http://184.72.215.120:8080/



```
[ec2-user@ip-172-31-31-47 ~]$ ls
apache-tomcat-10.1.50  apache-tomcat-10.1.50.tar.gz
[ec2-user@ip-172-31-31-47 ~]$ cd apache-tomcat-10.1.50/
[ec2-user@ip-172-31-31-47 apache-tomcat-10.1.50]$ ls
BUILDING.txt  CONTRIBUTING.md  LICENSE  NOTICE  README.md  RELEASE-NOTES  RUNNING.txt  bin  conf  lib  logs  temp  webapps  work
[ec2-user@ip-172-31-31-47 apache-tomcat-10.1.50]$ cd webapps/
[ec2-user@ip-172-31-31-47 webapps]$ ls
ROOT  docs  examples  host-manager  manager  maven-web-app  maven-web-app.war
[ec2-user@ip-172-31-31-47 webapps]$
```



Welcome to SCOPE INDIA, Your career partner! One of India's best Training destinations for Software, Networking, and Cloud Computing courses with 18 years of Industrial experience.

Click Here to see more about our company

Call Us / Whatsapp: +91-9745936073 / +91 - 7592939481 / +91 - 8075536185

Call Us / Whatsapp: +91-9745936073 / +91 - 7592939481 / +91 - 8075536185

Call Us / Whatsapp: +91-9745936073 / +91 - 7592939481 / +91 - 8075536185

Call Us / Whatsapp: +91-9745936073 / +91 - 7592939481 / +91 - 8075536185

Call Us / Whatsapp: +91-9745936073 / +91 - 7592939481 / +91 - 8075536185

Call Us / Whatsapp: +91-9745936073 / +91 - 7592939481 / +91 - 8075536185

❖ Conclusion

This project successfully implemented an end-to-end CI/CD pipeline on AWS using:

- Jenkins as the automation server
- SonarQube for code quality checks
- Nexus for artifact storage and versioning
- Tomcat for application deployment

The pipeline integrates source code management, automated builds, static code analysis, artifact repository management, and deployment into a seamless workflow. Each stage was tested and verified, ensuring reliability and efficiency in the delivery process.

Final Outcomes

- **Automated build and packaging with Maven**
- **Static code analysis performed with SonarQube, quality gate verified**
- **Artifacts stored securely in Nexus with proper versioning**
- **Automated deployment to Apache Tomcat server on AWS EC2**
- **Application successfully accessible via browser after deployment**

Authored by: Jenifa R | 17 January 2026