DEVELOPING

TESTING

CI/CD

Jenkins

Agenda

DEPLOY AN APPLICATION IN TOMCAT APPLICATION SERVER USING CI/CD

GIT: To maintain the source code.
MAVEN: To build the source code.
SONAR: For code quality test.
NEXUS: To store the artifact.
TOMCAT: Webserver to deploy an application.
JENKINS: To Integrate all the tools.

TOOLS

1. GIT
2. MAVEN
3. JENKINS
4. SONAR
5. NEXUS
6. TOMCAT

CODE

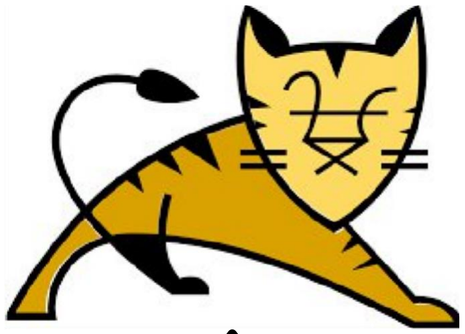
PULL

BUIL
D

WAR

WAR

WAR



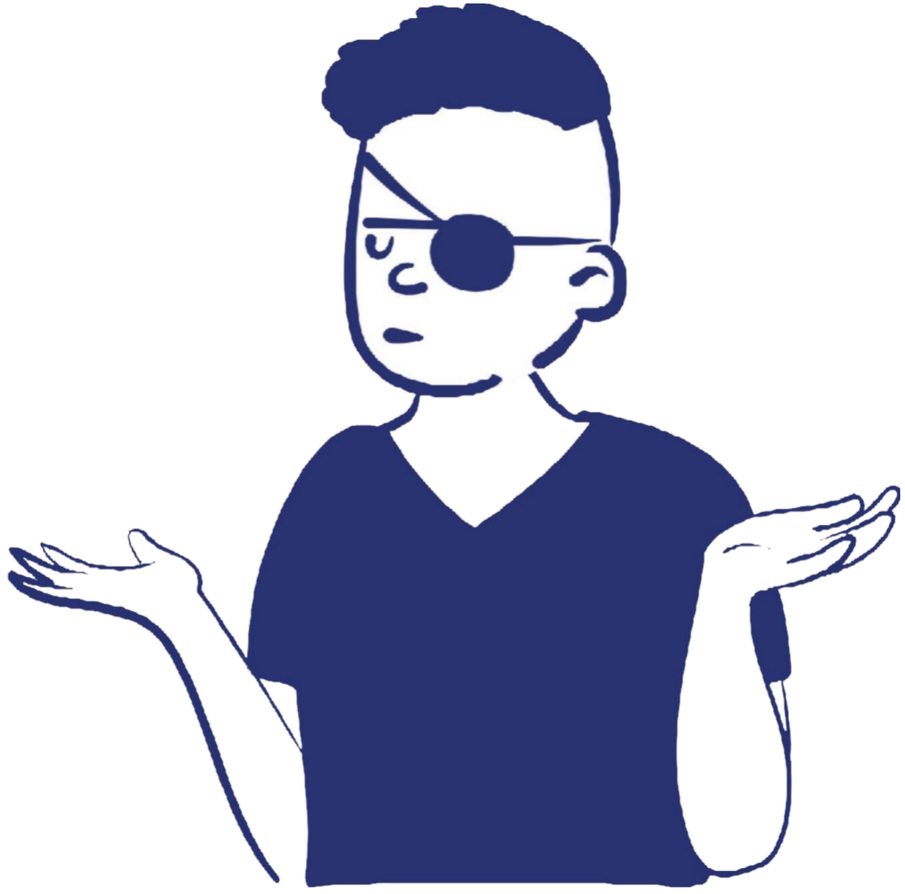
DEPLOYMENT:

The term Deployments refers the installation of a Web-Applicationon WebApplication server.

WHY DEPLOYMENT:

The main reasons of the deployment are: Adding

- New Features to the Application. Removing
- the Bugs.
- Enhancing the New Features.
- Improving the Performance.
- Breaking Large Applications to MicroServices.



NEED FOR DEPLOYMENT:

- We need to have Infra Setup. New
- Release of code.
- Involvement of Dev, QA and DevOps Team.
- Creating New Db’s If needed.
- Approval from RM (in case of New release).

PRECAUTIONS:

- Need to take Backups of the Current Builds and DataBases. Need
- To Provide Isolated Environments for Dev, Test and Prod.
- Can be able to do RollBack if the Deployment fails in some cases.
- Make sure we are deploying the correct env and Client Application Servers.

STEP-1:

LAUNCH 4 INSTANCES WITH SAME PEM FILE

1. TOMCAT: T2.MICRO
2. SONAR & Jenkins: T2.MEDIUM (20 GB OF EBS VOLUME)
3. NEXUS: T2.MEDIUM (20 GB OF EBS VOLUME)

SETUP SERVICES IN THEIR RESPECTIVE SERVERS.

STEP-2:

LOGIN INTO JENKINS DASHBOARD AND INSTALL THESE FOLLOWING PLUGINS

1. SONAR SCANNER: to scan the code
2. NEXUS ARTIFACTORY UPLOADED: to store the files in nexus3.Deploy to Container: To send the war files to tomcat server

STEP-3:

Create a jenkins pipeline job and write a Jenkins file for deploy a web application, usually we have 2 types of pipelines,

- scripted
- declarative

Here i am using scripted pipeline for the Jenkins file

STAGE-1 : GET THE CODE FROM GITHUB TO CI-SERVER

```
node {
  stage ("code") {
    git "https://github.com/bheemamallasani/one.git"
  }
}
```

STAGE-2 : BUILD THE SOURCE CODE:

GO TO MANAGE JENKINS >> TOOL >> MAVEN
ADD INSTALLER WITH THE NAME OF maven WITH VERSION (3.8.6)

```
node {
  stage ("build"){
    sh ‘mvn clean package’
  }
}
```

STAGE-3 : SCAN THE SOURCE CODE:

LOGIN INTO SONAR
GO TO MY ACCOUNT >> SECURITY >> ENTER A TOKEN NAME AND GENERATE A TOKENNOW INTEGRATE THE SONAR TO JENKINS
MANAGE JENKINS >> CONFIGURE SYSTEM >> SONAR SERVER :
NAME: mysonar
Url: PublicIP:9000/
credentials: —— (secretkey)

```
node
  { stage("Test") {
    withSonarQubeEnv('mysonar')
    {
      def mavenHome = tool name: "maven", type: "maven"
      def mavenCMD = "${mavenHome}/bin/mvn"
      sh "${mavenCMD} sonar:sonar"
    }
  }
}
```

NOTE: If the pipeline fails, check

- **the java version must be java11check the**
- **sonar credentials & sonar integration**
- **check the sonar url on manage jenkins add**
- **maven and sonar tools on manage jenkins >>>> tools**

STAGE-4 : UPLOAD WAR FILE INTO ARTIFACTORY:

CREATE A REPO IN NEXUS :

Name: pipeline-repo

formar: maven2 hosted

Version policy: releases

Deployment policy: allow redeploy

INSTALL NEXUS ARTIFACTORY UPLOAD PLUGIN

To GENERATE THE PIPELINE SYNTAX, WE NEED TO USE NEXUS ARTIFACTORY UPLOADER IN PIPELINE SYNTAX AND GIVE ALL INPUTS AND GENERATE PIPELINE SYNTAX

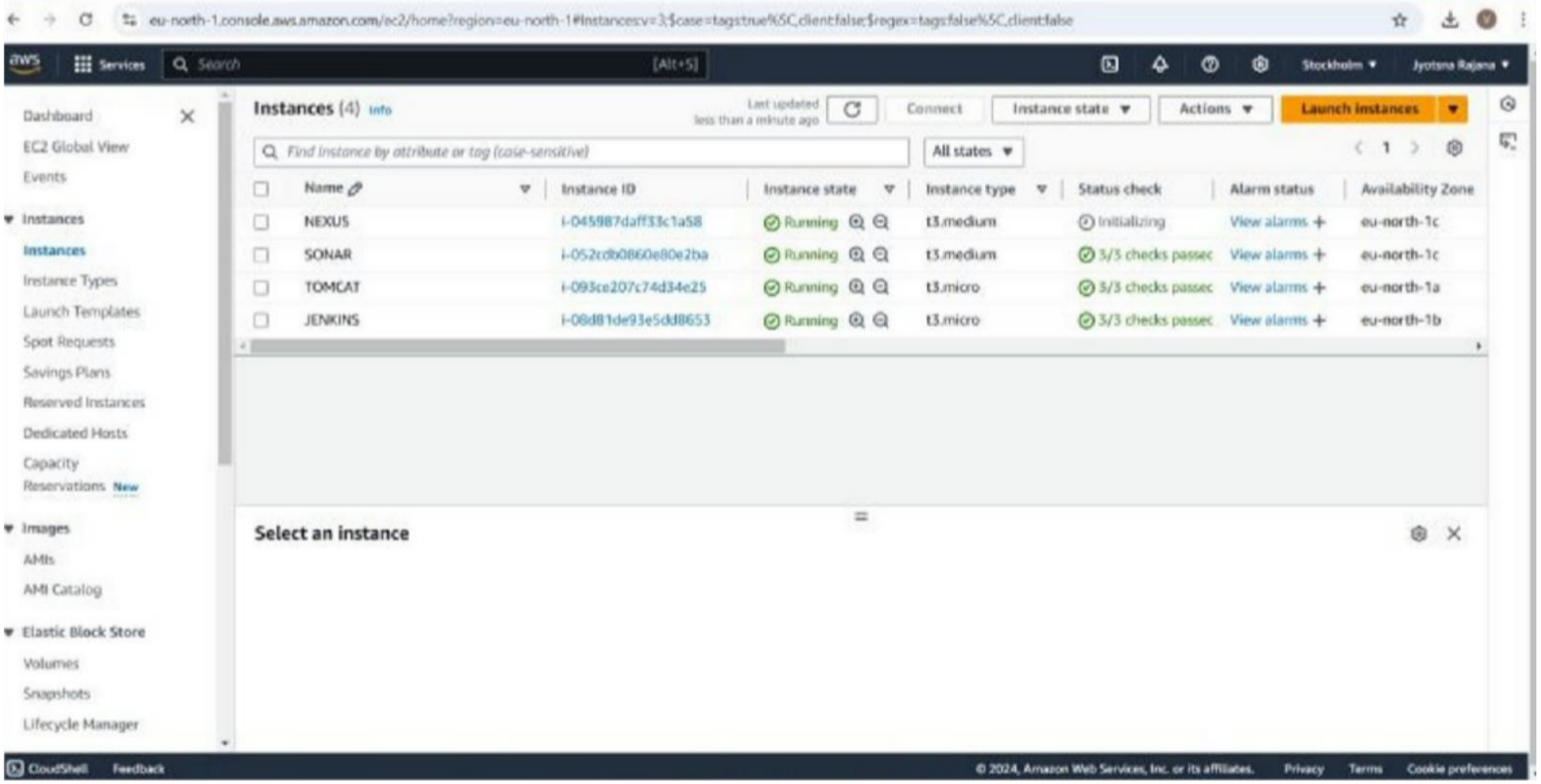
————————————————————————————————————

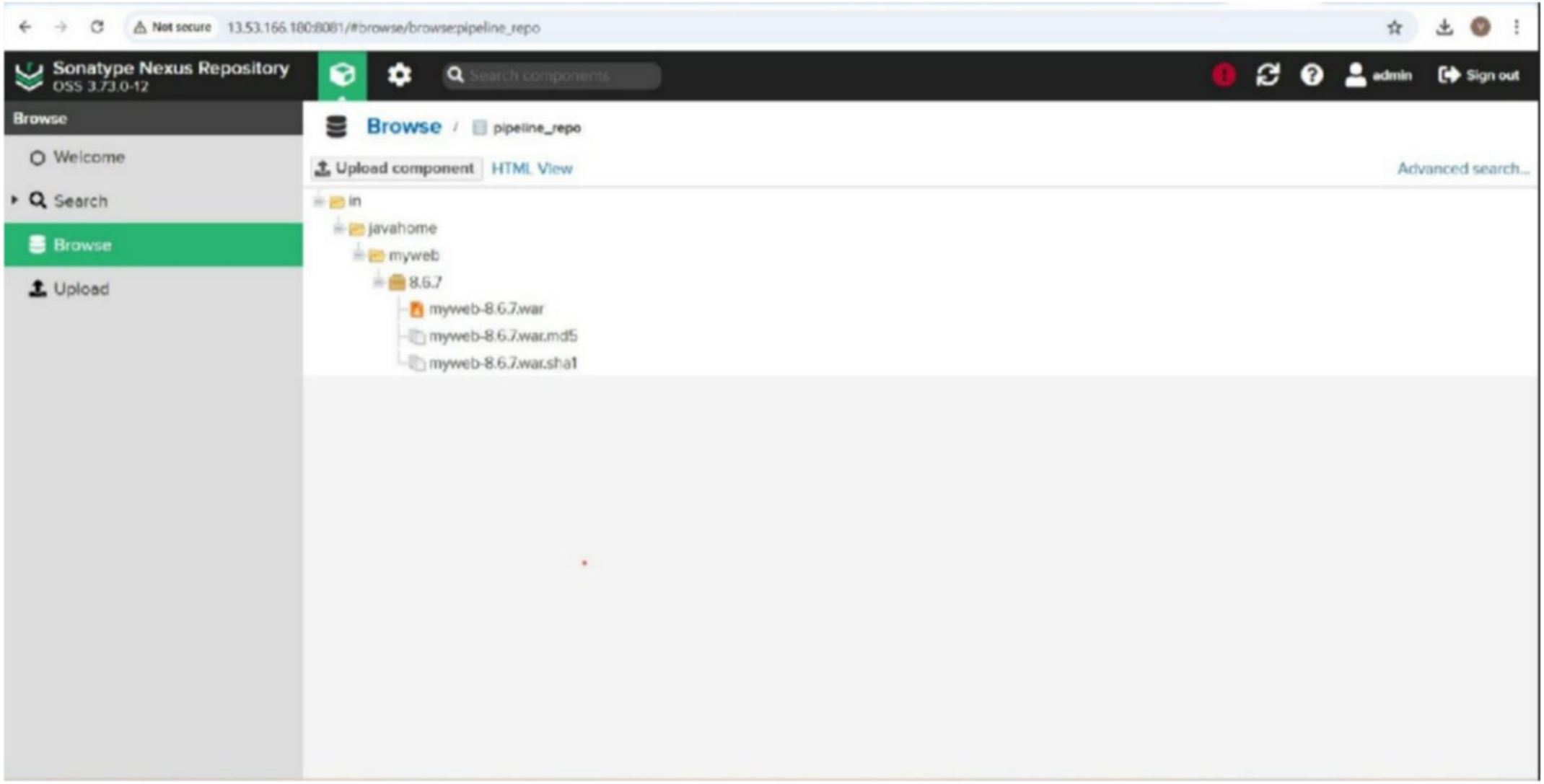
```
node {
  stage ("upload") {
    nexusArtifactUploader artifacts: [[artifactId: 'myweb', classifier: '', file: 'target/myweb-8.3.5.war', type: '.war']], credentialsId: '4949746a-34ae-4d80-acaa-a73815fda645', groupId: 'in.javahome', nexusUrl: '18.117.135.27:8081', nexusVersion: 'nexus3', protocol: 'http', repository: 'pipeline-repo', version: '8.3.5'
  }
}
```

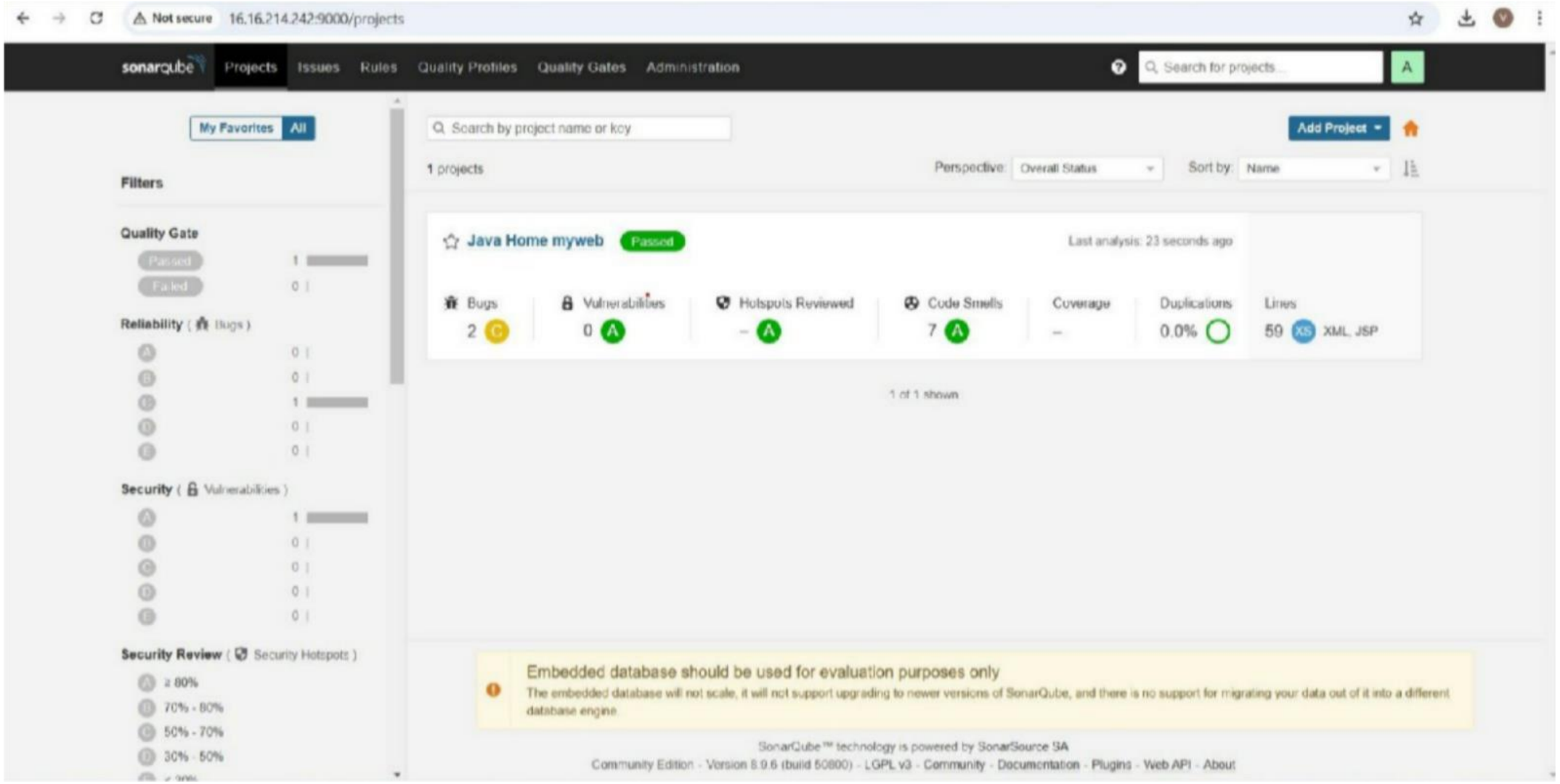
STAGE-5 : DEPLOY THE APPLICATION INTO TOMCAT WEB SERVER:

AND USE PIPELINE SYNTAX

```
node {
  stage ("deploy") {
     deploy adapters: [tomcat9(credentialsld: 'tomcat', path: '', url: 'http://3.91.64.252:8080')], contextPath:
     'pipeline_mini_project', war: 'target/*.war
    }
  }
}
```


eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#Instances:v=3;$case=tags:true%5C,client:false;$regex=tags:false%5C,client:false
Services
Search
[Alt+S]
Stockholm
Jyotsna Rajana
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Instances (4) info
Last updated less than a minute ago
Connect
Instance state
Actions
Launch instances
Find instance by attribute or tag (case-sensitive)
All states
1
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
NEXUS
i-045987daff33c1a58
Running
t3.medium
Initializing
View alarms
eu-north-1c
SONAR
i-052cdb0860e80e2ba
Running
t3.medium
3/3 checks passed
View alarms
eu-north-1c
TOMCAT
i-093ce207c74d34e25
Running
t3.micro
3/3 checks passed
View alarms
eu-north-1a
JENKINS
i-08d81de93e5dd8653
Running
t3.micro
3/3 checks passed
View alarms
eu-north-1b
Select an instance
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Not secure
13.53.166.180:8081/#browse/browse:pipeline_repo
Sonatype Nexus Repository
OSS 3.73.0-12
Search components
admin
Sign out
Browse
Welcome
Search
Browse
Upload
Browse / pipeline_repo
Upload component
HTML View
Advanced search...
in
javahome
myweb
8.6.7
myweb-8.6.7.war
myweb-8.6.7.war.md5
myweb-8.6.7.war.sha1


Not secure
16.16.214.242:9000/projects
sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
Search for projects...
A
My Favorites
All
Filters
Quality Gate
Passed 1
Failed 0
Reliability (Bugs)
A 0
B 0
C 1
D 0
E 0
Security (Vulnerabilities)
A 1
B 0
C 0
D 0
E 0
Security Review (Security Hotspots)
A ≥ 80%
B 70% - 80%
C 50% - 70%
D 30% - 50%
Search by project name or key
Add Project
1 projects
Perspective: Overall Status
Sort by: Name
Java Home myweb
Passed
Last analysis: 23 seconds ago
Bugs
2 C
Vulnerabilities
0 A
Hotspots Reviewed
– A
Code Smells
7 A
Coverage
–
Duplications
0.0%
Lines
59 XS XML, JSP
1 of 1 shown
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 - Version 8.9.6 (build 50800) - LGPL v3 - Community - Documentation - Plugins - Web API - About

Not secure 13.49.69.29:8080/manager/html

/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	2	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/myweb-0.6.7	None specified	Archetype Created Web Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/one_application	None specified	Archetype Created Web Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/pipeline_job	None specified	Archetype Created Web Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/pipeline_mini_project	None specified	Archetype Created Web Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/sonar	None specified	Archetype Created Web Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Deploy

Deploy directory or WAR file located on server

Context Path:

Version (for parallel deployment):

XML Configuration file path:

WAR or Directory path:

Deploy