

# DEVSECOPS PROJECTS – 2 (COMPLETE CI-CD PETSHOP Project) Deploy to Tomcat

**NAME:** S. Venkateswarareddy

**EMAIL:** Singareddyv91@gmail.com

**PHONE NO:** 9491029062

**BATCH NO:** 3PM, 113

**COURSE:** AWS AND DEVOPS

**TRAINER NAME:** V. MADHUKAR REDDY

## PIPELINE SYNTAX:

DEVSECOPS-Project-2-Venkatesh

```
pipeline{
    agent any

    tools {
        jdk 'jdk17'
        maven 'maven3'
    }

    environment {
        SCANNER_HOME=tool 'sonar-scanner'
    }

    stages{
        stage ('clean Workspace'){
            steps{
                cleanWs()
            }
        }

        stage ('checkout scm') {
            steps {
                git 'https://github.com/Aj7Ay/jpetstore-6.git'
            }
        }

        stage ('maven compile') {
            steps {
```

```

        sh 'mvn clean compile'
    }
}
stage ('maven Test') {
    steps {
        sh 'mvn test'
    }
}
stage("Sonarqube Analysis "){
    steps{
        withSonarQubeEnv('sonar-server') {
            sh "' $SCANNER_HOME/bin/sonar-scanner -Dsonar.projectName=Petshop \
            -Dsonar.java.binaries=. \
            -Dsonar.projectKey=Petshop '"
        }
    }
}
stage("quality gate"){
    steps {
        script {
            waitForQualityGate abortPipeline: false, credentialsId: 'Sonar-token'
        }
    }
}
stage("maven package"){
    steps {
        sh 'mvn package'
    }
}
stage("OWASP Dependency Check"){
    steps{
        dependencyCheck additionalArguments: '--scan ./ --format XML ', odciInstallation: 'Dependency-
Check'
    }
}

```

```

        dependencyCheckPublisher pattern: '**/dependency-check-report.xml'
    }
}

stage("nexus artifactory"){
    steps{
        nexusArtifactUploader artifacts: [[artifactId: 'jpetstore', classifier: '', file:
'/var/lib/jenkins/workspace/Devsecops-project-2/target/jpetstore.war', type: 'war']], credentialsId: '',
groupId: 'org.mybatis', nexusUrl: '', nexusVersion: 'nexus2', protocol: 'http', repository: 'maven-snapshots',
version: '6.1.1-SNAPSHOT'

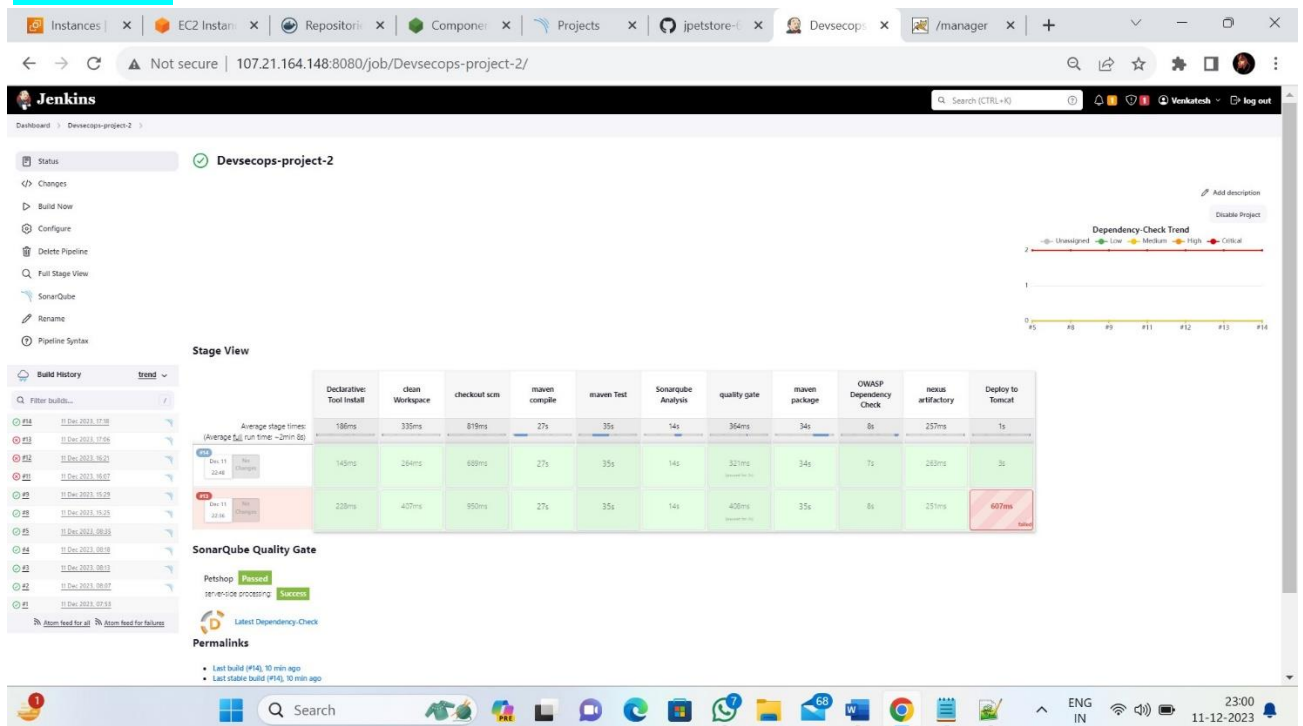
    }
}

stage('Deploy to Tomcat'){
    steps{
        deploy adapters: [tomcat9(credentialsId: 'tomcat', path: '', url: 'http://107.21.164.148:8083')],
contextPath: null, war: '**/*.war'

    }
}
}

```

## OUTPUT:



## SonarQube Analysis status:

The screenshot shows the SonarQube web interface. The top navigation bar includes 'sonarqube', 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', and 'Administration'. A search bar is present. The left sidebar has 'My Favorites' and 'All' tabs. The main content area shows a search for 'Petshop' with 1 project(s) found. The project 'Petshop' is marked as 'Passed'. The last analysis was 3 minutes ago. The metrics displayed are: Bugs (66, C), Vulnerabilities (0, A), Hotspots Reviewed (0.0%, E), Code Smells (79, A), Coverage (0.0%, R), Duplications (5.2%, Y), and Lines (6.7k, S). A warning message states: '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.'

Metric	Value	Rating
Bugs	66	C
Vulnerabilities	0	A
Hotspots Reviewed	0.0%	E
Code Smells	79	A
Coverage	0.0%	R
Duplications	5.2%	Y
Lines	6.7k	S

## Dependency check status:

The screenshot shows the Jenkins web interface. The top navigation bar includes 'Instances | EC2 | us-east-1 | EC2 Instance Connect | Projects | Dependency-Check Report | Pipeline Syntax: Snippet |'. The main content area shows the 'Dependency-Check Results' for the 'Devsecops-project-2' job. The severity distribution is shown as a bar chart with 2 Critical, 2 High, and 2 Medium vulnerabilities. The table below lists the vulnerabilities:

File Name	Vulnerability	Severity	Weakness
+ jpetstore.war: spring-core-5.3.29.jar	<a href="#">CVE-2016-1000027</a>	Critical	CWE-502
+ jpetstore.war: spring-web-5.3.29.jar	<a href="#">CVE-2016-1000027</a>	High	CWE-502
+ spring-core-5.3.29.jar	<a href="#">CVE-2016-1000027</a>	Critical	CWE-502
+ spring-web-5.3.29.jar	<a href="#">CVE-2016-1000027</a>	High	CWE-502

Instances | EC2 | us-east-1 | EC2 Instance Connect | Projects | Dependency-Check Rep | Pipeline Syntax: Snippet | +

18.207.151.235:8080/job/Devsecops-project-2/lastCompletedBuild/dependency-check-findings/

# Jenkins

Search (CTRL+K) Venkatesh log out

Dashboard > Devsecops-project-2 > #5 > Dependency-Check

- Status
- Changes
- Console Output
- View as plain text
- Edit Build Information
- Delete build '#5'
- Git Build Data
- Dependency-Check**
- Restart from Stage
- Replay

## Dependency-Check Results

SEVERITY DISTRIBUTION

File Name	Vulnerability	Severity	Weakness
+ jpetstore.war: spring-core-5.3.29.jar	NVD CVE-2016-1000027	Critical	CWE-502
+ jpetstore.war: spring-web-5.3.29.jar	OSSINDEX CVE-2016-1000027	High	CWE-502
+ spring-core-5.3.29.jar	NVD CVE-2016-1000027	Critical	CWE-502
+ spring-web-5.3.29.jar	OSSINDEX CVE-2016-1000027	High	CWE-502

29°C Haze 14:11 11-12-2023

## Nexus Artifact Uploader:

SecurityGr | EC2 Instan | DevSecop | Pipeline Sy | Projects | Componen | jpetstore | Docker Hu | +

54.167.247.114:8081/#browse/browse/components:maven-snapshots:e5c50d09b73fd3c5edd9446dd1d23a80

# Nexus Repository Manager

Search components admin Sign out

Components / maven-snapshots / jpetstore

Repository	maven-snapshots	Group	org.mybatis	Most popular version
Format	maven2	Name	jpetstore	Age
		Version	6.11-20231219141123-1	Popularity

Delete component

Name ↑
org.mybatis/jpetstore/6.11-SNAPSHOT/jpetstore-6.11-20231219141123-1.war
org.mybatis/jpetstore/6.11-SNAPSHOT/jpetstore-6.11-20231219141123-1.war.md5
org.mybatis/jpetstore/6.11-SNAPSHOT/jpetstore-6.11-20231219141123-1.war.sha1

20°C 19-12-2023

## Deployed to Tomcat:

The screenshot shows the Tomcat Web Application Manager interface. At the top, there's a message bar with "Message: OK". Below it, a navigation bar includes "List Applications", "HTML Manager Help", "Manager Help", and "Server Status". The main section is titled "Applications" and contains a table with columns: Path, Version, Display Name, Running, Sessions, and Commands. The table lists several applications, including "Welcome to Tomcat", "Tomcat Documentation", "Servlet and JSP Examples", "Tomcat Host Manager Application", "JPetStore", and "Tomcat Manager Application". Each application has a "Start", "Stop", "Reload", and "Undeploy" button, along with an "Expire sessions" option. Below the table, there's a "Deploy" section with a text input field for the "Deploy directory or WAR file located on server". The browser's address bar shows "107.21.164.148:8083/manager/html".

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

## FINALL OUTPUT:

The screenshot shows the JPetStore Demo application. The header includes the "JPetStore Demo" logo, a "Sign In ?" link, and a search bar. The main content area is divided into two sections. On the left, there's a list of categories: "Fish" (Saltwater, Freshwater), "Dogs" (Various Breeds), "Cats" (Various Breeds, Exotic Varieties), "Reptiles" (Lizards, Turtles, Snakes), and "Birds" (Exotic Varieties). On the right, there's a large image of a green parrot, surrounded by smaller circular images of various animals (fish, dog, cat, bird, reptile). The footer shows the URL "www.mybatis.org". The browser's address bar shows "107.21.164.148:8083/jpetstore/actions/Catalog.action".

## **Plugins Installed In these Project are:**

**Maven Integration (No Need to restart the jenkins)**

**Config file provider**

**Pipeline maven integration**

**Eclipse Temurin installer**

**Sonarqube Scanner**

**OWASP dependency-Check**

**Deploy to container**