

DEVOPS PROFESSIONAL CERTIFICATION PROGRAM

Lab – 6: Deployment — Jfrog Artifactory / Apache Tomcat

- Create Java Application
 - Simple Java Application
 - Web Application
 - Spring-Boot Application
- Configure Jenkins to job to perform deployment of Artifact into binary repository (Jfrog) / container (Tomcat)

Explore:

- Deploy artifact into artifactory using “distributionManagement”
- Deploy artifact into artifactory using Jenkins plugin “Deploy artifact to Artifactory”
- Deploy web application artifact into webserver by configuring pom.xml
- Deploy web application artifact into webserver by using Jenkins plugin “Deploy to war/ear container”
- Execute simple jar file to observe solution response on console
- Run Spring-boot application and check responses using curl
- Deploy war into container and walkthrough web solution using browser

A. Deploy to Jfrog

- **Option A: Using maven goal “deploy”**

(As explained in earlier lab, configure pom, maven settings)

- Configure pom.xml using “distributionManagement”
- Deploy settings.xml on local repository path
- Configure Artifactory server details in Maven settings.xml
- Run maven goal “deploy”

- **Option B: Using Jenkins plugin “deploy artifact to Artifactory” – Steps are illustrated below**

1. Create Jenkins Job – Maven Project

The screenshot shows the Jenkins 'New Item' form. At the top, there is a text input field for 'Enter an item name' with the value '7 Lab'. Below this, there are three project type options: 'Freestyle project', 'Maven project', and 'Pipeline'. The 'Maven project' option is selected and highlighted with a blue border. The descriptions for each option are as follows:

- Freestyle project:** This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
- Maven project:** Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
- Pipeline:** Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

2. Configure SCM & Build with maven goal package / “install”

The screenshot shows the Jenkins configuration page for a new job. It is divided into two main sections: 'Source Code Management' and 'Build'.

Source Code Management: The 'Git' radio button is selected. Under 'Repositories', the 'Repository URL' is set to '/home/osgdev/up/GettingStarted/mavenProcess/Demo1'. The 'Credentials' dropdown is set to 'none'. There are buttons for 'Add Repository' and 'Advanced...'. There is also an 'Add' button next to the credentials dropdown.

Build: The 'Root POM' is set to 'pom.xml'. The 'Goals and options' are set to 'clean install'.

3. Configure Artifactory under “Add post build Action”

The screenshot shows the Jenkins configuration page for a new job, specifically the 'Post Steps' tab. A dropdown menu is open, showing a list of post-build actions. The option 'Deploy artifacts to Artifactory' is highlighted with a blue dashed border. Other options in the list include 'Aggregate downstream test results', 'Archive the artifacts', 'Build other projects', 'Deploy artifacts to Maven repository', 'Record fingerprints of files to track usage', 'Git Publisher', 'GitLab commit status publisher', 'SonarQube analysis with Maven', 'Accept GitLab merge request on success', 'Add note with build status on GitLab merge requests', 'Add vote for build status on GitLab merge requests', 'Deploy war/ear to a container', 'Editable Email Notification', and 'Publish build status to GitLab commit (GitLab 8.1+ required)'. Below the dropdown, there are two radio buttons: 'Run only if build succeeds or is unstable' (selected) and 'Run regardless of build result'.

Post-build Actions

Deploy artifacts to Artifactory

Artifactory server

Target releases repository

Different Value

Target snapshot repository

Different Value

Custom staging configuration

Refreshing...

Refresh

☐ Override default credentials

Save

Apply

build is unstable

Deploy artifacts to Artifactory

Artifactory server

Target releases repository

Different Value

Target snapshot repository

Different Value

Custom staging configuration

Items refreshed successfully

Refresh

☒ Override default credentials

Username

Password

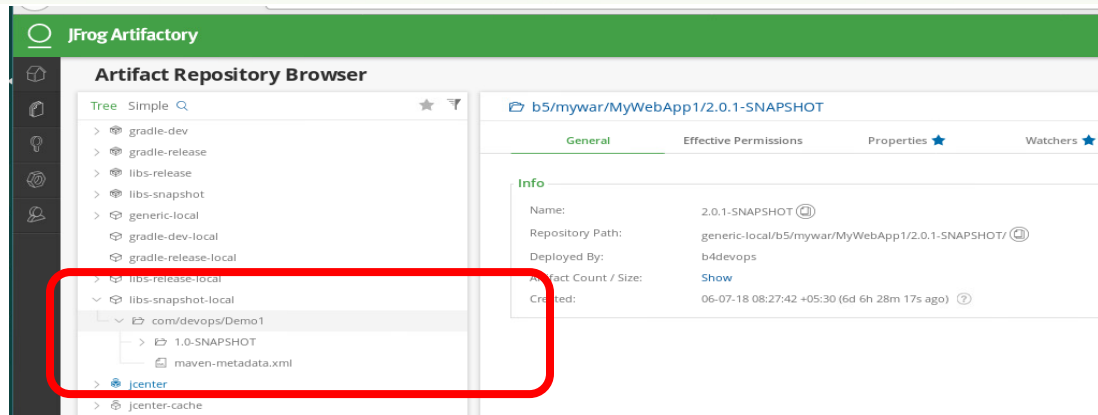
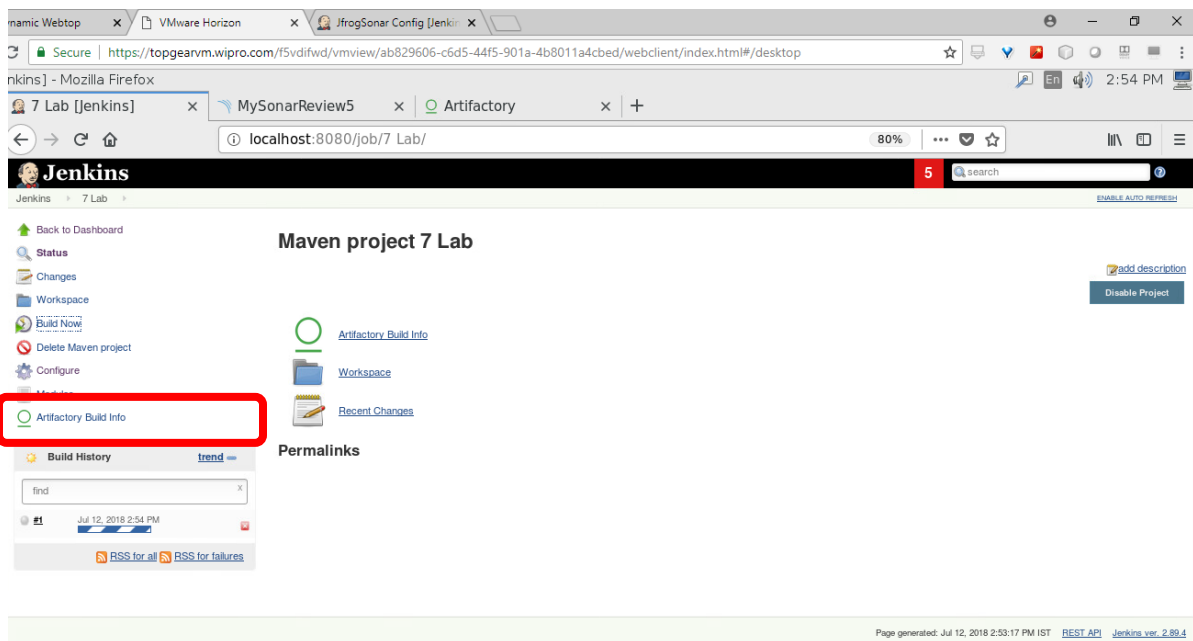
Save

Apply

build is unstable

- Select “Build Now” to start build process. Observe Jenkins Console output

```
[JENKINS] Recording test results
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ Demo1 ---
[INFO] Building jar: /var/lib/jenkins/workspace/7 Lab/target/Demo1-1.0-SNAPSHOT.jar
[INFO] --- maven-install-plugin:2.4:install (default-install) @ Demo1 ---
[INFO] Installing /var/lib/jenkins/workspace/7 Lab/target/Demo1-1.0-SNAPSHOT.jar to /var/lib/jenkins/.m2/repository/com/devops/Demo1/1.0-SNAPSHOT/Demo1-1.0-SNAPSHOT.jar
[INFO] Installing /var/lib/jenkins/workspace/7 Lab/pom.xml to /var/lib/jenkins/.m2/repository/com/devops/Demo1/1.0-SNAPSHOT/Demo1-1.0-SNAPSHOT.pom
[INFO] Artifactory Build Info Recorder: Saving build info to /var/lib/jenkins/workspace/7 Lab/target/build-info.json
[INFO] Deploying artifact: http://localhost:5040/artifactory/libs-snapshot-local/com/devops/Demo1/1.0-SNAPSHOT/Demo1-1.0-SNAPSHOT.jar
[INFO] Deploying artifact: http://localhost:5040/artifactory/libs-snapshot-local/com/devops/Demo1/1.0-SNAPSHOT/Demo1-1.0-SNAPSHOT.pom
[INFO] Artifactory Build Info Recorder: Deploying build info ...
[INFO] Deploying build descriptor to: http://localhost:5040/artifactory/api/build
[INFO] Build successfully deployed. Browse it in Artifactory under http://localhost:5040/artifactory/webapp/builds/Demo1-$BUILD_NUMBER/1
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 21.766 s
[INFO] Finished at: 2018-07-12T14:55:12+05:30
[INFO] Final Memory: 29M/183M
[INFO] -----
[JENKINS] Archiving /var/lib/jenkins/workspace/7 Lab/pom.xml to com.devops/Demo1/1.0-SNAPSHOT/Demo1-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/7 Lab/target/Demo1-1.0-SNAPSHOT.jar to com.devops/Demo1/1.0-SNAPSHOT/Demo1-1.0-SNAPSHOT.jar
channel stopped
Finished: SUCCESS
```



Note: Explore deploying release copy, prefix with build number for tracking, downloading artifacts etc.,

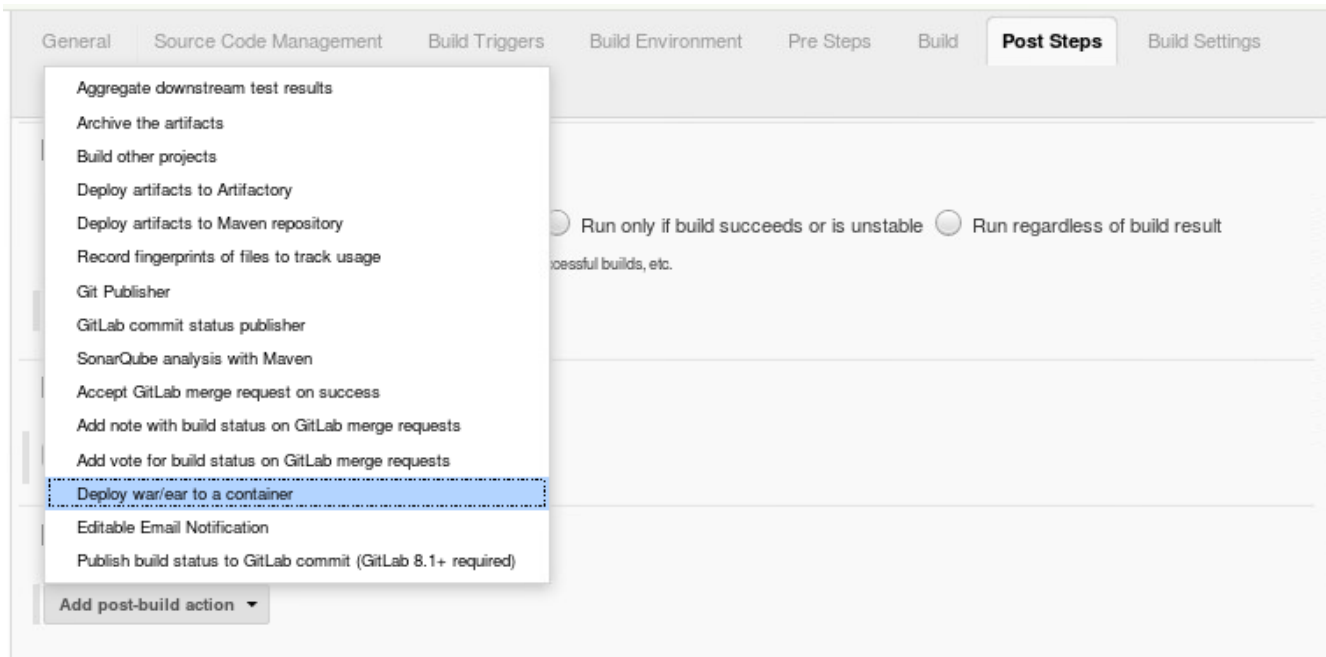
B. Deploy to Tomcat

- **Option A: Using maven goal “deploy with maven-deploy plugin”**

- Configure pom.xml using tomcat plugin to setup server & maven-deploy-plugin for deployment
- Run maven goal “tomcat7:deploy”

- **Option B: Using Jenkins plugin “deploy war/ear to container” – Steps are illustrated below**

- Create / Use Web Application type - maven project (created in earlier Lab).
- Create Jenkins Job – Maven Project
- Configure SCM & Build with maven goal “package” / “install”
- Configure Apache Tomcat under “Add post build Action” > Select Deploy war/ear to a container
- Specify Tomcat url and credentials



Post-build Actions

Deploy war/ear to a container
X

WAR/EAR files

Context path

Containers

Tomcat 8.x
X

Credentials Add

Tomcat URL

Add Container

Deploy on failure ☐

- Select “Build Now” to start build process. Notice Jenkins Console output for deployment

```
[INFO] Installing /var/lib/jenkins/workspace/8 Lab/pom.xml to /var/lib/jenkins/.m2/repository/com/devops/Demo2/1.0-SNAPSHOT/Demo2-1.0-SNAPSHOT.pom
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 11.087 s
[INFO] Finished at: 2018-07-17T12:29:40+05:30
[INFO] Final Memory: 15M/156M
[INFO] -----
Waiting for Jenkins to finish collecting data
[JENKINS] Archiving /var/lib/jenkins/workspace/8 Lab/pom.xml to com.devops/Demo2/1.0-SNAPSHOT/Demo2-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/8 Lab/target/Demo2.war to com.devops/Demo2/1.0-SNAPSHOT/Demo2-1.0-SNAPSHOT.war
channel stopped
Deploying /var/lib/jenkins/workspace/8 Lab/target/Demo2.war to container Tomcat 8.x Remote with context Lab8
[/var/lib/jenkins/workspace/8 Lab/target/Demo2.war] is not deployed. Doing a fresh deployment.
Deploying [/var/lib/jenkins/workspace/8 Lab/target/Demo2.war]
Finished: SUCCESS
```

Page generated: Jul 17, 2018 12:29:15 PM IST [REST API](#) [Jenkins ver. 2.89.4](#)

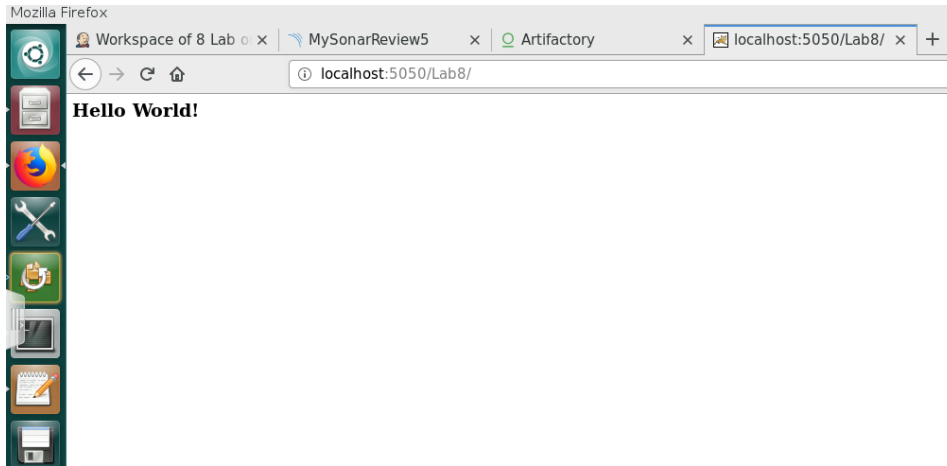
- Upon build process success, refresh tomcat service to verify deployment

Message:	FAIL - No context exists named [#47;MyArtifactory]
-----------------	--

Manager		
List Applications	HTML Manager Help	

Applications					
Path	Version	Display Name	Running	Sessions	Com
/	None specified	Welcome to Tomcat	true	0	Start
/Lab8	None specified	Archetype Created Web Application	true	0	Start
/...	None specified	Tomcat Demo Application	true	0	Start

- **Verify application execution on browser**

**Explore execution of jar after packaging**

option 1: `java -cp <yourPath>/yourPackage.jar somepackage.yourMainClass`

option 2: update pom with manifest

`java -jar <yourPath>/yourPackage.jar`

```
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-jar-plugin</artifactId>
    <version>2.4</version>
    <configuration>
        <archive>
            <manifest>
                <mainClass>com.base.quickdemo.App</mainClass>
            </manifest>
        </archive>
    </configuration>
</plugin>
```

Option3: `nohup java -jar example.jar & (nohup stands for "no hangup.")`