

Absolutely, Sarvesh! Here's a complete **step-by-step summary** of how you successfully built a Java web application using **Maven, Jenkins, GitHub, and deployed it to Apache Tomcat** as part of your DevOps lab program:

✅ Lab Program Summary: CI/CD Pipeline using Maven, Jenkins & Tomcat

🔧 1. Create a Maven Web Project

- Use Maven to generate a project:
 - `mvn archetype:generate -DgroupId=myapp -DartifactId=myapp -DarchetypeArtifactId=maven-archetype-webapp -DinteractiveMode=false`
 - Or create the directory manually and inside it place a pom.xml with:
 - `<packaging>war</packaging>`
-

📄 2. Configure pom.xml

Add the following if not already:

```
<build>
<plugins>
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-war-plugin</artifactId>
<version>3.3.2</version>
<configuration>
<webXml>src/main/webapp/WEB-INF/web.xml</webXml>
</configuration>
</plugin>
</plugins>
</build>
```

✅ Ensure your folder structure is:

```
src/
main/
java/
webapp/
```

index.html

WEB-INF/

web.xml

3. Build the WAR file

From the project root (where pom.xml is):

mvn clean install

✅ WAR gets created in:
target/yourappname.war

4. Push Code to GitHub

git init

git add .

git commit -m "Initial commit"

git remote add origin https://github.com/yourusername/repo.git

git branch -M main # Ensure the branch name is main

git push -u origin main

5. Install & Configure Jenkins

- Install Jenkins (Windows Installer or .war)
- Start Jenkins on <http://localhost:8080>

Install Plugins:

- Git Plugin
 - Maven Integration Plugin
 - Deploy to Container Plugin
-

6. Configure Jenkins Tools

Go to:

Manage Jenkins → Global Tool Configuration

Set up:

- JDK (point to your JDK installation)

- Maven (point to Maven installation or install via Jenkins)
-

7. Setup Apache Tomcat

- Download [Tomcat 9](#)
 - Extract it, e.g., to C:\apache-tomcat-9.0.105
 - Set environment variable:
 - set JAVA_HOME=C:\Program Files\Java\jdk-17
 - Start Tomcat:
 - cd C:\apache-tomcat-9.0.105\bin
 - startup.bat
 - Confirm at:
http://localhost:8081/
-

8. Enable Tomcat Manager

Edit tomcat-users.xml in:

C:\apache-tomcat-9.0.105\conf\

Add:

```
<role rolename="manager-script"/>
```

```
<user username="admin" password="admin" roles="manager-script"/>
```

 Save and restart Tomcat.

9. Jenkins Job Setup

- **New Item → Freestyle Project**
- **Source Code Management → Git**
Add repo URL: <https://github.com/yourusername/repo.git>
- **Build → Add build step → Invoke top-level Maven targets**
Target: clean install
- **Post-build Actions → Deploy WAR/EAR to a container**
 - WAR file: **/target/*.war
 - Container: Tomcat 9.x remote
 - Manager URL: <http://localhost:8081/manager/text>
 - Context path: /yourapp

- Add Credentials: admin / admin
-

10. Build & Deploy

- Click **Build Now**
 - You'll see logs like:
 - BUILD SUCCESS
 - Deploying to Tomcat...
-

11. Verify Deployment

Visit:

<http://localhost:8081/yourapp/>

-  You should see your index.html or JSP page loaded.
-

Done!

You've now:

- Built a Maven WAR project
- Used Git + GitHub
- Configured Jenkins to auto-build from GitHub
- Deployed automatically to Tomcat

Let me know if you want a **PDF version** of this summary or a **diagram**!