

# 1. Prerequisites (System & Config)

Before creating the job in Jenkins, ensure these settings are active on your Laptop.

## A. Tools should present in your Laptop

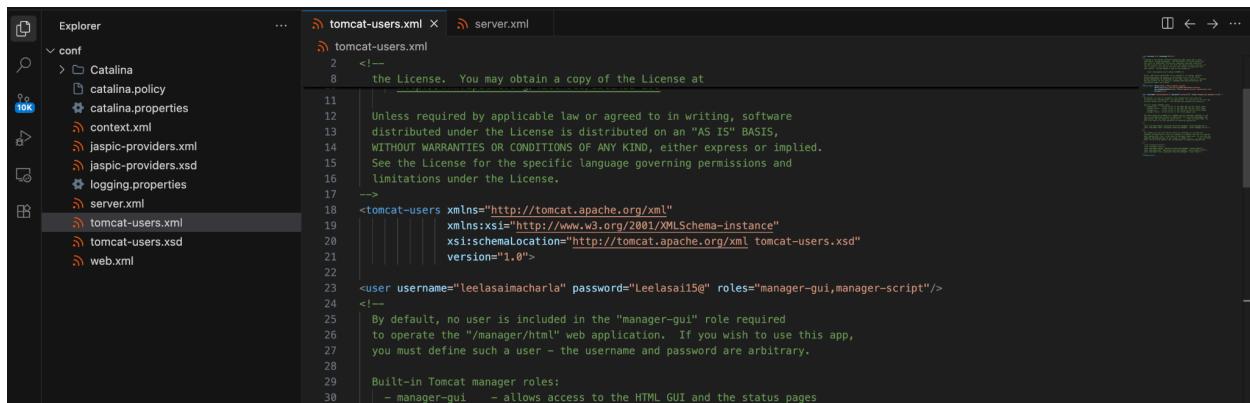
1. Install jenkins
2. Install Tomcat@9
3. Install Maven
4. Install JDK (with respect to the tomcat)
5. Install Git

## B. Tomcat User Permissions

Jenkins needs access to the Tomcat Manager API. Ensure your /opt/homebrew/etc/tomcat@9/tomcat-users.xml has these specific roles:

XML

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



```
tomcat-users.xml | server.xml
tomcat-users.xml
<!--
  The License. You may obtain a copy of the License at
  http://www.apache.org/licenses/LICENSE-2.0
  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
-->
<tomcat-users xmlns="http://tomcat.apache.org/xml"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
               version="1.0">
  <user username="leelasaimacharla" password="Leelasai150" roles="manager-gui,manager-script"/>
<!--
  By default, no user is included in the "manager-gui" role required
  to operate the "/manager/html" web application. If you wish to use this app,
  you must define such a user - the username and password are arbitrary.
-->
  Built-in Tomcat manager roles:
    - manager-gui      - allows access to the HTML GUI and the status pages

```

**IMP :** Make sure the port numbers and server ports of Jenkins and Tomcat are configured (shouldn't use same port number or server port)

## C. Required Jenkins Plugins

Go to **Manage Jenkins > Manage Plugins > Available** and install:

1. Deploy to container
2. Maven integration
3. Config file provider
4. Pipeline maven integration
5. Pipeline plugin api
6. Git Plugin

The screenshot shows the Jenkins dashboard. At the top, there are links for 'New Item', 'Build History', 'Project Relationship', and 'Check File Fingerprint'. Below these are two dropdown menus: 'Build Queue' (with 3 items) and 'Build Executor Status' (0/2). The main area displays a table of builds with columns: S (Status), W (Waiting), Name, Last Success, Last Failure, and Last Duration. The builds listed are 'Maven-pipeline', 'Maven-pipeline2', and 'Maven-pipeline3'. At the bottom, there are icons for 'Icon: S M L' and links for 'REST API' and 'Jenkins 2.541.1'.

The screenshot shows the 'Manage Jenkins' page under 'System Configuration'. It includes sections for 'System', 'Nodes', 'Managed files', 'Security', 'Tools', 'Clouds', 'Plugins' (which is highlighted with a red box), 'Appearance', 'Credentials', and 'Credential Providers'. A search bar at the top right says 'Search settings'.

The screenshot shows the Jenkins 'Manage Jenkins > Plugins' page. A search bar at the top contains the text 'Git'. Below it, a list of available plugins is shown. The first item is 'Git server' version 137.ve0060b\_432302, which is checked and has a green status badge with the number '93'. The second item is 'Pipeline: Deprecated Groovy Libraries' version 615.vb\_b\_0664a\_b\_19f3, which is unchecked and has an orange status badge with the number '71'. A tooltip on this item states: 'This plugin is deprecated. In general, this means that it is either obsolete, no longer being developed, or may no longer work. Learn more.' Other items listed include 'Jersey 2 API' and 'Git Pipeline for Blue Ocean'.

## D. Give the tomcat credentials

Go to **Manage Jenkins > Credentials** and give all the required fields. Then save.

The screenshot shows the Jenkins 'Manage Jenkins > Credentials > Global credentials (unrestricted)' page. A modal window titled 'Update credentials' is open. The 'Scope' dropdown is set to 'Global (Jenkins, nodes, items, all child items, etc)'. The 'Username' field contains 'leelasaimacharla'. The 'Password' field is set to 'Concealed'. The 'ID' field is 'tomcat\_credentials'. The 'Description' field is 'tomcat\_credentials'. At the bottom of the modal is a blue 'Save' button.

## 2. Jenkins Global Tool Configuration

You must tell Jenkins exactly where your Java and Maven are located on your Laptop.

1. Go to **Manage Jenkins > Global Tool Configuration**.

The screenshot shows the Jenkins 'Manage Jenkins' interface. In the top left, there's a Jenkins logo and the text 'Jenkins / Manage Jenkins'. On the right, there's a search bar and some icons. Below the header, there are two main sections: 'System Configuration' and 'Security'. Under 'System Configuration', there are several items: 'System' (Configure global settings and paths), 'Nodes' (Add, remove, control and monitor the various nodes Jenkins runs jobs on), 'Managed files' (e.g. settings.xml for maven, central managed scripts, custom files, ...), 'Tools' (Configure tools, their locations and automatic installers, highlighted with a red box), 'Clouds' (Add, remove, and configure cloud instances to provision agents on-demand), 'Plugins' (Add, remove, disable or enable plugins that can extend the functionality of Jenkins), and 'Appearance' (Configure the look and feel of Jenkins). Under 'Security', there are 'Security' (Secure Jenkins; define who is allowed to access/use the system), 'Credentials' (Configure credentials), and 'Credential Providers' (Configure the credential providers and types).

2. **JDK:** Add JDK, name it "JDK17", and set the path to:

- `/Library/Java/JavaVirtualMachines/jdk-17.jdk/Contents/Home`

The screenshot shows the Jenkins 'Tools' configuration page under 'Manage Jenkins'. The title is 'JDK installations'. There's a button '+ Add JDK'. A form is open for adding a new JDK:

- Name: jdk23
- JAVA\_HOME: /Library/Java/JavaVirtualMachines/jdk-17.jdk/Contents/Home
- Install automatically

At the bottom, there are 'Save' and 'Apply' buttons.

3. **Git:** Add Git, name it "Git", and check "Install automatically" or provide the path if installed.

The screenshot shows the Jenkins 'Tools' configuration page under 'Manage Jenkins'. A 'Git' entry is selected. The configuration form includes fields for 'Name' (set to 'Default'), 'Path to Git executable' (set to 'git'), and an unchecked 'Install automatically' checkbox. Below the form are 'Save' and 'Apply' buttons.

### 3. Creating the Jenkins Pipeline (Step-by-Step)

#### Step 1: General & Source Code Management (SCM)

1. **New Item** > Enter EcommerceApp-Pipeline > Select **Maven project**.

The screenshot shows the Jenkins dashboard. On the left, a red box highlights the '+ New Item' button. The main area displays a table of existing projects: 'Maven-pipeline' (status green, last success 1 hr 54 min ago, last failure 1 hr 57 min ago, duration 4.8 sec), 'Maven-pipeline2' (status green, last success 1 hr 32 min ago, last failure 1 hr 35 min ago, duration 2 min 8 sec), and 'Maven-pipeline3' (status green, last success 1 hr 23 min ago, last failure 1 hr 25 min ago, duration 9.1 sec). At the bottom, the footer shows 'Icon: S M L' and 'REST API Jenkins 2.541.1'.

2. When you click the new item you will be directed to the next page as shown below.  
Now give the name, select Freestyle project and click ok.

New Item

Enter an item name  
I

Select an item type

Pipeline Build, test, and deploy using pipelines. Supports stages, parallel work, and running on multiple agents.

**Freestyle project** Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

Maven project Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

Multi-configuration project Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter.

OK

3. In general add a description.

Configure General Enabled

Description

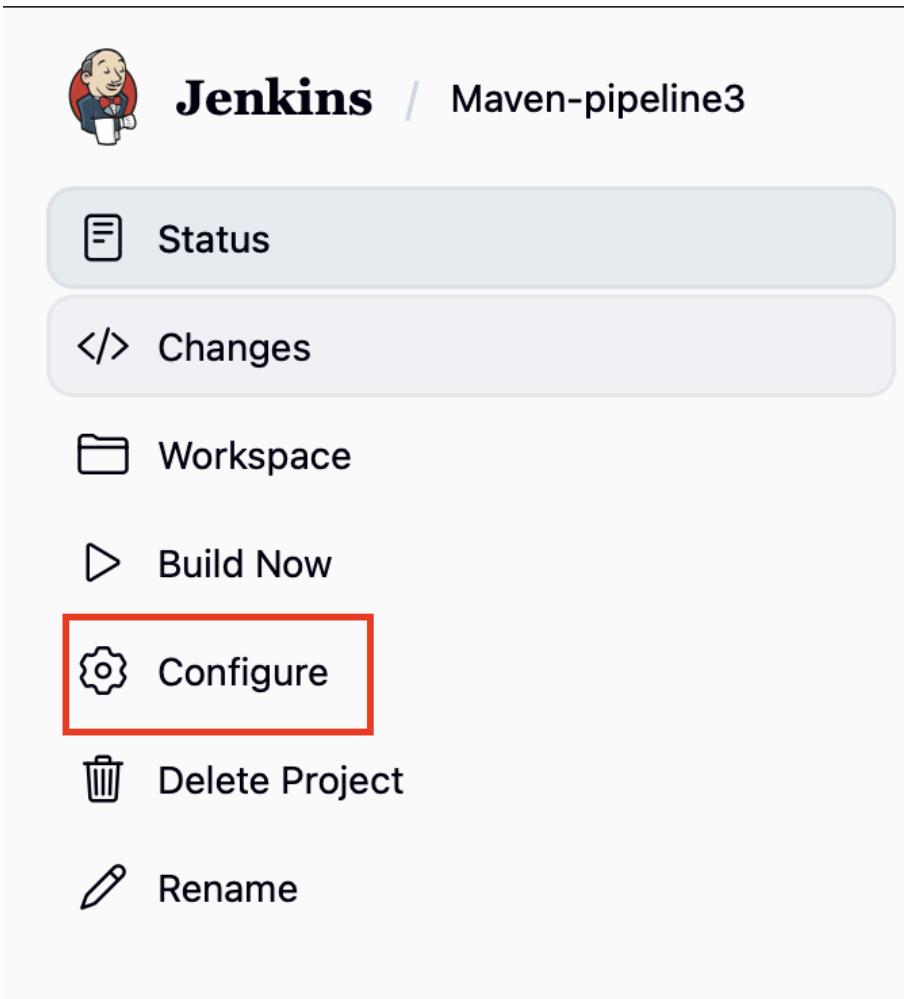
This is a maven application

Plain text Preview

Discard old builds ?  
 GitHub project  
 This project is parameterized ?  
 Throttle builds ?

Save Apply

4. Click on configure. Then we will go to configure page



5. Now if we scroll little bit down we will find **Source Code Management** section

A screenshot of the "Source Code Management" configuration section. It shows a "None" option selected for now, with "Git" selected. The "Repositories" section contains a single repository entry with a URL of "https://github.com/Msocial123/EcommerceApp.git", no credentials specified, and an "Advanced" dropdown. A "Add Repository" button is at the bottom.

By default it will be **None** we need to change it to **Git** and provide the repo url in which the project is present.

6. Now if we scroll a little down we will find **Branches to build** section. Here we need to provide in which branch the project is present(main/master).

The screenshot shows the Jenkins configuration interface for a job named "Maven-pipeline3". The left sidebar has tabs for General, Source Code Management (selected), Triggers, Environment, Build Steps, and Post-build Actions. The main panel shows the "Source Code Management" section with a "Branches to build" configuration. It includes a "Branch Specifier" field containing "\*/\*master" and a "Repository browser" dropdown set to "(Auto)". There are "Save" and "Apply" buttons at the bottom.

7. Leave Triggers and Environment sections

## 8. Build Settings

1. **Root POM:** EcommerceApp/pom.xml (Based on your logs, the POM is inside a subfolder).
2. **Goals and options:** clean package

The screenshot shows the Jenkins configuration interface for the same job. The left sidebar has tabs for General, Source Code Management, Triggers, Environment, Build Steps (selected), and Post-build Actions. The main panel shows the "Build Steps" configuration, which includes an "Invoke top-level Maven targets" step. Under "Goals", the value "clean package" is listed. There are "Advanced" and "Edited" buttons, along with a "+ Add build step" button at the bottom.

## 9. Now go to Post\_build Actions

Click the question mark we will get text bar below copy **\*\*/\*war** without (" ")

The screenshot shows the Jenkins 'Configure' screen for the 'Maven-pipeline3' job. The left sidebar has 'Post-build Actions' selected. In the main area, under 'Post-build Actions', there is a section titled 'Deploy war/ear to a container'. It includes a 'WAR/EAR files' input field containing '\*\*/\*.war' and a note below it stating 'War/ear files to deploy. Relative to the workspace root. You can also specify Ant-style GLOBs, like "\*\*/\*.war" (from Deploy to container Plugin)'. A red box highlights the question mark icon in the 'WAR/EAR files' field and the '\*\*/\*.war' text in the note.

## 10. Give Context Path

11. Now for Containers click add container and choose **Tomcat9x Remote**

The screenshot shows the Jenkins 'Configure' screen for the 'Maven-pipeline3' job. The left sidebar has 'Post-build Actions' selected. In the main area, under 'Post-build Actions', there is a 'Context path' input field containing 'EcommerceApp' and a 'Containers' section with a '+ Add Container' button highlighted by a red box. Below these are 'Deploy on failure' and '+ Add post-build action' buttons. At the bottom are 'Save' and 'Apply' buttons. A red box highlights the '+ Add Container' button.

Jenkins / Maven-pipeline3 / Configure

Configure

General

Source Code Management

Triggers

Environment

Build Steps

Post-build Actions

Context path ? EcommerceApp

Containers

- + Add Container
- GlassFish 2.x
- GlassFish 3.x
- GlassFish 4.x
- JBoss AS 3.x
- JBoss AS 4.x
- JBoss AS 5.x
- JBoss AS 6.x
- JBoss AS 7.x
- Tomcat 4.x Remote
- Tomcat 5.x Remote
- Tomcat 6.x Remote
- Tomcat 7.x Remote
- Tomcat 8.x Remote
- Tomcat 9.x Remote

REST API Jenkins 2.541.1

Jenkins / Maven-pipeline3 / Configure

Configure

General

Source Code Management

Triggers

Environment

Build Steps

Post-build Actions

Context path ? EcommerceApp

Containers

- Tomcat 9.x Remote

Credentials

- none -

Tomcat URL ?

Advanced

+ Add Container

Deploy on failure

Save Apply

12. Now Click none and select the previous created credentials and give the Tomcat url

Jenkins / Maven-pipeline3 / Configure

Configure

General

Source Code Management

Triggers

Environment

Build Steps

Post-build Actions

Context path ? EcommerceApp

Containers

- Tomcat 9.x Remote

Credentials

- none -

leefasaimcharia/\*\*\*\*\* (tomcat\_credentials)

Tomcat URL ?

Advanced

+ Add Container

Deploy on failure

+ Add post-build action

Save Apply

The screenshot shows the Jenkins 'Configure' screen for the pipeline 'Maven-pipeline3'. The left sidebar has 'Post-build Actions' selected. The main area is titled 'Post-build Actions' with the sub-instruction 'Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs.' A 'Deploy war/ear to a container' action is configured with 'WAR/EAR files' set to '\*\*/\*.war', 'Context path' set to 'EcommerceApp', and 'Containers' set to 'Tomcat 9.x Remote'. Under 'Tomcat URL', the value 'http://localhost:8082' is entered. A credential 'leelasaimacharla/\*\*\*\*\*\*\*\* (tomcat\_credentials)' is selected.

13. Now click **apply and save**. After successfully saving click on pipeline name ("Maven-pipeline3")

The screenshot shows the Jenkins 'Configure' screen for the pipeline 'Maven-pipeline3'. The left sidebar has 'General' selected. The right panel shows the 'General' configuration with a description 'Maven-pipeline: 3'. Under 'Advanced' settings, several options are listed: 'Discard old builds', 'GitHub project', 'This project is parameterized', 'Throttle builds', and 'Execute concurrent builds if necessary'. The 'Enabled' toggle switch is turned on. At the bottom, there are 'Save' and 'Apply' buttons, with 'Save' highlighted by a red box.

14. Now to run the pipeline click on **Build Now**

The screenshot shows the Jenkins Pipeline screen for the pipeline 'Maven-pipeline3'. The left sidebar has 'Build Now' selected. The main area displays the pipeline status with a green checkmark icon and the text 'Maven-pipeline3'. Below it is a 'Permalinks' section listing recent builds. At the bottom, a 'Builds' table shows two entries: '#2 4:16pm' (highlighted with a red box) and '#1 4:14pm'. The 'Build Now' button in the sidebar is also highlighted with a red box.

15. If we click the tick mark or wrong mark we will be redirected to the console output. If we find **success** at the last line then our pipeline is running successfully else we can see the errors and we should rectify them.

The screenshot shows the Jenkins pipeline console output for job 'Maven-pipeline3' run #2. The output details the Maven build process, including compilation, test execution, war packaging, and deployment to Tomcat. The final message indicates a successful deployment:

```
[INFO] NO SOURCES TO COMPILE
[INFO]
[INFO] --- surefire:3.2.5:test (default-test) @ EcommerceApp ---
[INFO] No tests to run.
[INFO]
[INFO] --- war:3.4.0:war (default-war) @ EcommerceApp ---
[INFO] Packaging webapp
[INFO] Assembling webapp [EcommerceApp] in [/Users/leelasaimacharla/.jenkins/workspace/Maven-pipeline3/EcommerceApp/target/EcommerceApp]
[INFO] Processing war project
[INFO] Copying webapp resources [/Users/leelasaimacharla/.jenkins/workspace/Maven-pipeline3/EcommerceApp/src/main/webapp]
[INFO] Building war: /Users/leelasaimacharla/.jenkins/workspace/Maven-pipeline3/EcommerceApp/target/EcommerceApp.war
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 6.326 s
[INFO] Finished at: 2026-01-28T16:16:24+05:30
[INFO] -----
[DeployPublisher][INFO] Attempting to deploy 1 war file(s)
[DeployPublisher][INFO] Deploying /Users/leelasaimacharla/.jenkins/workspace/Maven-pipeline3/EcommerceApp/target/EcommerceApp.war to container Tomcat 9.x Remote with context EcommerceApp
[INFO] [/Users/leelasaimacharla/.jenkins/workspace/Maven-pipeline3/EcommerceApp/target/EcommerceApp.war] is not deployed.
Doing a fresh deployment.
Deploying [/Users/leelasaimacharla/.jenkins/workspace/Maven-pipeline3/EcommerceApp/target/EcommerceApp.war]
Finished: SUCCESS
```

16. Go to Tomcat Server and Click on Manager App and Login with Tomcat Credentials

The screenshots show the Tomcat Web Application Manager interface and the Spring Petclinic application home page.

**Tomcat Web Application Manager:**

- Message:** OK
- Manager:** List Applications, HTML Manager Help, Manager Help, Server Status
- Applications:**

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	<a href="#">Start</a> <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Undeploy</a> <a href="#">Expire sessions</a> with idle ≥ 30 minutes
EcommerceApp	None specified		true	0	<a href="#">Start</a> <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Undeploy</a> <a href="#">Expire sessions</a> with idle ≥ 30 minutes
/petclinic/boom.xml	None specified		true	0	<a href="#">Start</a> <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Undeploy</a> <a href="#">Expire sessions</a> with idle ≥ 30 minutes
/ShoppingApp	None specified	shopping-cart	true	0	<a href="#">Start</a> <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Undeploy</a> <a href="#">Expire sessions</a> with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	<a href="#">Start</a> <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Undeploy</a> <a href="#">Expire sessions</a> with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	<a href="#">Start</a> <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Undeploy</a> <a href="#">Expire sessions</a> with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	<a href="#">Start</a> <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Undeploy</a> <a href="#">Expire sessions</a> with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	<a href="#">Start</a> <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Undeploy</a> <a href="#">Expire sessions</a> with idle ≥ 30 minutes
/maven-web-application/boom.xml	None specified	maven-web-application	true	0	<a href="#">Start</a> <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Undeploy</a> <a href="#">Expire sessions</a> with idle ≥ 30 minutes
- Deploy:** Deploy directory or WAR file located on server

**Spring Petclinic Home Page:**

Welcome

spring.vetstreet.com