

## TASK 5 MAVEN CREATION

### Step 1: Folder Creation

Create a folder named maven and install git in that folder.

### Step 2: Installing Java

Clone the repository inside the folder and install java openjdk-17-jdk -y

### Step 3: Maven test

Test the maven and check any errors present

### Step 4: Maven Clean

Clean the maven and remove the unwanted items

```
shalini@DESKTOP-32J80T2: ~ - x + v
shalini@DESKTOP-32J80T2:~/devops/spring-framework-petclinic$ ls
Jenkinsfile LICENSE.txt dockerfile mvnw mvnw.cmd pom.xml readme.md src
shalini@DESKTOP-32J80T2:~/devops/spring-framework-petclinic$ mvn clean
mvn test
[INFO] Scanning for projects...
[INFO] -----< org.springframework.samples:spring-framework-petclinic >-----
[INFO] Building Spring Framework Petclinic 6.1.4
[INFO] [ war ]-----
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/2.5/maven-clean-plugin-2.5.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/2.5/maven-clean-plugin-2.5.pom (3.9 kB at 9.2 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/22/maven-plugins-22.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/22/maven-plugins-22.pom (13 kB at 225 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/21/maven-parent-21.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/21/maven-parent-21.pom (26 kB at 361 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/apache/10/apache-10.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/apache/10/apache-10.pom (15 kB at 336 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/2.5/maven-clean-plugin-2.5.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/2.5/maven-clean-plugin-2.5.jar (25 kB at 513 kB/s)
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ spring-framework-petclinic ---
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-plugin-api/2.0.6/maven-plugin-api-2.0.6.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-plugin-api/2.0.6/maven-plugin-api-2.0.6.pom (1.5 kB at 38 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven/2.0.6/maven-2.0.6.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven/2.0.6/maven-2.0.6.pom (9.0 kB at 171 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/5/maven-parent-5.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/5/maven-parent-5.pom (15 kB at 339 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/apache/3/apache-3.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/apache/3/apache-3.pom (3.4 kB at 78 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0/plexus-utils-3.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0/plexus-utils-3.0.pom (4.1 kB at 110 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/sonatype/spice/spice-parent/16/spice-parent-16.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/sonatype/spice/spice-parent/16/spice-parent-16.pom (8.4 kB at 186 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-plugin-api/2.0.6/maven-plugin-api-2.0.6.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-plugin-api/2.0.6/maven-plugin-api-2.0.6.jar (13 kB at 292 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0/plexus-utils-3.0.jar (226 kB at 1.5 MB/s)
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 16.324 s
[INFO] Finished at: 2025-03-22T04:39:21Z
[INFO] -----
shalini@DESKTOP-32J80T2:~/devops/spring-framework-petclinic$ cd target
shalini@DESKTOP-32J80T2:~/devops/spring-framework-petclinic/target$ ls
classes generated-test-sources maven-archiver petclinic site test-classes
generated-sources jacoco.exec maven-status petclinic.war surefire-reports
shalini@DESKTOP-32J80T2:~/devops/spring-framework-petclinic/target$ cd ..
shalini@DESKTOP-32J80T2:~/devops/spring-framework-petclinic$ docker build -t petclinic .
[+] Building 3.7s (8/8) FINISHED
ker:default
=> [internal] load build definition from dockerfile
0.1s
=> => transferring dockerfile: 148B
0.0s
=> [internal] load metadata for docker.io/library/tomcat:latest
2.6s
=> [auth] library/tomcat:pull token for registry-1.docker.io
0.0s
=> [internal] load .dockerignore
0.0s
=> => transferring context: 2B
0.0s
=> [internal] load build context
0.4s
=> => transferring context: 44.38MB
0.3s
=> CACHED [1/2] FROM docker.io/library/tomcat:latest@sha256:1374a565d5122fdb42807f3a5f2d4fcc245a5e15426ff5bb5123afedc8ef769d
0.0s
=> [2/2] COPY target/*.war /usr/local/tomcat/webapps/ROOT.war
0.2s
=> exporting to image
0.3s
=> => exporting layers
0.3s
=> => writing image sha256:7b258f401222d54b336651dfc168300a8212908lead4f7f385169c1216238809
```

## Step 5: Login in Docker

Login into docker using the username and password of the docker hub.

## Step 6: Push in Docker

Push the image inside the docker hub

```
shalini@DESKTOP-32J80T2: ~  
-tree  
  
What's Next?  
View a summary of image vulnerabilities and recommendations → docker scout quickview  
shalini@DESKTOP-32J80T2:~/devops/spring-framework-petclinic$ docker login -u ysshallini  
Password:  
Login Succeeded  
shalini@DESKTOP-32J80T2:~/devops/spring-framework-petclinic$ docker tag petclinic ysshallini/petclinic:latest  
shalini@DESKTOP-32J80T2:~/devops/spring-framework-petclinic$ docker push ysshallini/petclinic:latest  
The push refers to repository [docker.io/ysshallini/petclinic]  
95e3f64b9a76: Pushed  
5f70bf18a086: Mounted from library/tomcat  
6fbd02a6a33: Mounted from library/tomcat  
49cb1bc2daeb: Mounted from library/tomcat  
4e5b554b7345: Mounted from library/tomcat  
39cf8ac89a5a: Pushed  
f844dcf94898: Pushed  
3359bc3d7a6a: Pushed  
4b7c01ed0534: Pushed  
latest: digest: sha256:1fc5556e6b506ed4e1222a51c285cb7260761b4df506d6bf8d4626f0e8fb49c2 size: 2413  
shalini@DESKTOP-32J80T2:~/devops/spring-framework-petclinic$ minikube start  
🐳 minikube v1.35.0 on Ubuntu 24.04 (amd64)  
🔗 Using the docker driver based on existing profile  
👉 Starting "minikube" primary control-plane node in "minikube" cluster  
📶 Pulling base image v0.0.46 ...  
🔄 Updating the running docker "minikube" container ...  
🔧 Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...  
🔍 Verifying Kubernetes components...  
  * Using image gcr.io/k8s-minikube/storage-provisioner:v5  
🌟 Enabled addons: storage-provisioner, default-storageclass  
👉 Done! kubectrl is now configured to use "minikube" cluster and "default" namespace by default  
shalini@DESKTOP-32J80T2:~/devops/spring-framework-petclinic$ kubectl create deployment petclinic --image=ysshallini/petclinic --port=8080  
error: unknown flag: --images  
See 'kubectl create deployment --help' for usage.  
shalini@DESKTOP-32J80T2:~/devops/spring-framework-petclinic$ kubectl create deployment petclinic --image=ysshallini/petclinic --port=8080  
deployment.apps/petclinic created  
shalini@DESKTOP-32J80T2:~/devops/spring-framework-petclinic$ kubectl expose deployment.apps/petclinic --port=8080 --type=NodePort  
service/petclinic exposed
```

## Step 7: Minikube

Start the minikube using minikube start

## Step 8: Deployment

Create the deployment and expose it .Once it is exposed use minikube service to find the url of the webpage. Copy the url.

```
4b7c01ed0534: Pushed  
latest: digest: sha256:1fc5556e6b506ed4e1222a51c285cb7260761b4df506d6bf8d4626f0e8fb49c2 size: 2413  
shalini@DESKTOP-32J80T2:~/devops/spring-framework-petclinic$ minikube start  
🐳 minikube v1.35.0 on Ubuntu 24.04 (amd64)  
🔗 Using the docker driver based on existing profile  
👉 Starting "minikube" primary control-plane node in "minikube" cluster  
📶 Pulling base image v0.0.46 ...  
🔄 Updating the running docker "minikube" container ...  
🔧 Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...  
🔍 Verifying Kubernetes components...  
  * Using image gcr.io/k8s-minikube/storage-provisioner:v5  
🌟 Enabled addons: storage-provisioner, default-storageclass  
👉 Done! kubectrl is now configured to use "minikube" cluster and "default" namespace by default  
shalini@DESKTOP-32J80T2:~/devops/spring-framework-petclinic$ kubectl create deployment petclinic --image=ysshallini/petclinic --port=8080  
error: unknown flag: --images  
See 'kubectl create deployment --help' for usage.  
shalini@DESKTOP-32J80T2:~/devops/spring-framework-petclinic$ kubectl create deployment petclinic --image=ysshallini/petclinic --port=8080  
deployment.apps/petclinic created  
shalini@DESKTOP-32J80T2:~/devops/spring-framework-petclinic$ kubectl expose deployment.apps/petclinic --port=8080 --type=NodePort  
service/petclinic exposed  
shalini@DESKTOP-32J80T2:~/devops/spring-framework-petclinic$ minikube service petclinic  
minikube: command not found  
shalini@DESKTOP-32J80T2:~/devops/spring-framework-petclinic$ minikube service petclinic  


| NAMESPACE | NAME      | TARGET PORT | URL                       |
|-----------|-----------|-------------|---------------------------|
| default   | petclinic | 8080        | http://192.168.49.2:30997 |

  
👉 Starting tunnel for service petclinic.  


| NAMESPACE | NAME      | TARGET PORT | URL                    |
|-----------|-----------|-------------|------------------------|
| default   | petclinic |             | http://127.0.0.1:39617 |

  
🌐 Opening service default/petclinic in default browser...  
👉 http://127.0.0.1:39617  
! Because you are using a Docker driver on linux, the terminal needs to be open to run it.
```

## Step 6: Output

Paste the url link in the browser and the output will be displayed.

