

Step-by-Step Guide to Deploying a Spring Boot Application with Docker and Kubernetes

NAME : SANDEEP P S

ROLLNUM : 22CSL261

1. Initialize and Clone the Repository

Initializes a new Git repository.

Clones the Spring Framework PetClinic project from GitHub.

Code:

```
git init
git clone "https://github.com/AranganathanPrakash/spring-framework-petclinic"
```

Screenshot:

```
root@Ubuntu:/home/vboxuser/task5.1# git init
hint: Using 'master' as the name for the initial branch. This default branch nam
e
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/vboxuser/task5.1/.git/
```

2. Navigate to the Project Directory

Moves into the cloned repository folder.

Code:

```
cd spring-framework-petclinic
```

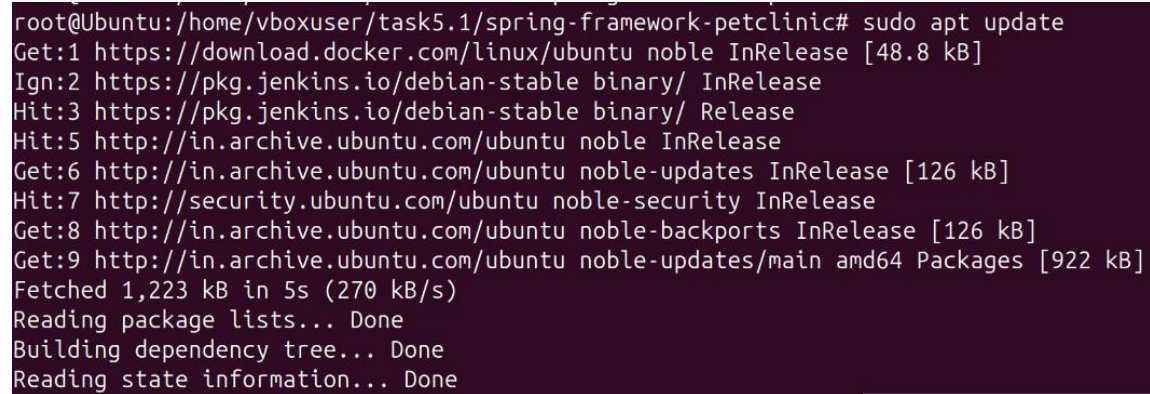
3. Update System Packages

Updates the package list to ensure the latest versions are available.

Code:

```
sudo apt update
```

Screenshot:



```
root@Ubuntu:/home/vboxuser/task5.1/spring-framework-petclinic# sudo apt update
Get:1 https://download.docker.com/linux/ubuntu noble InRelease [48.8 kB]
Ign:2 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:3 https://pkg.jenkins.io/debian-stable binary/ Release
Hit:5 http://in.archive.ubuntu.com/ubuntu noble InRelease
Get:6 http://in.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:7 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:8 http://in.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [922 kB]
Fetched 1,223 kB in 5s (270 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

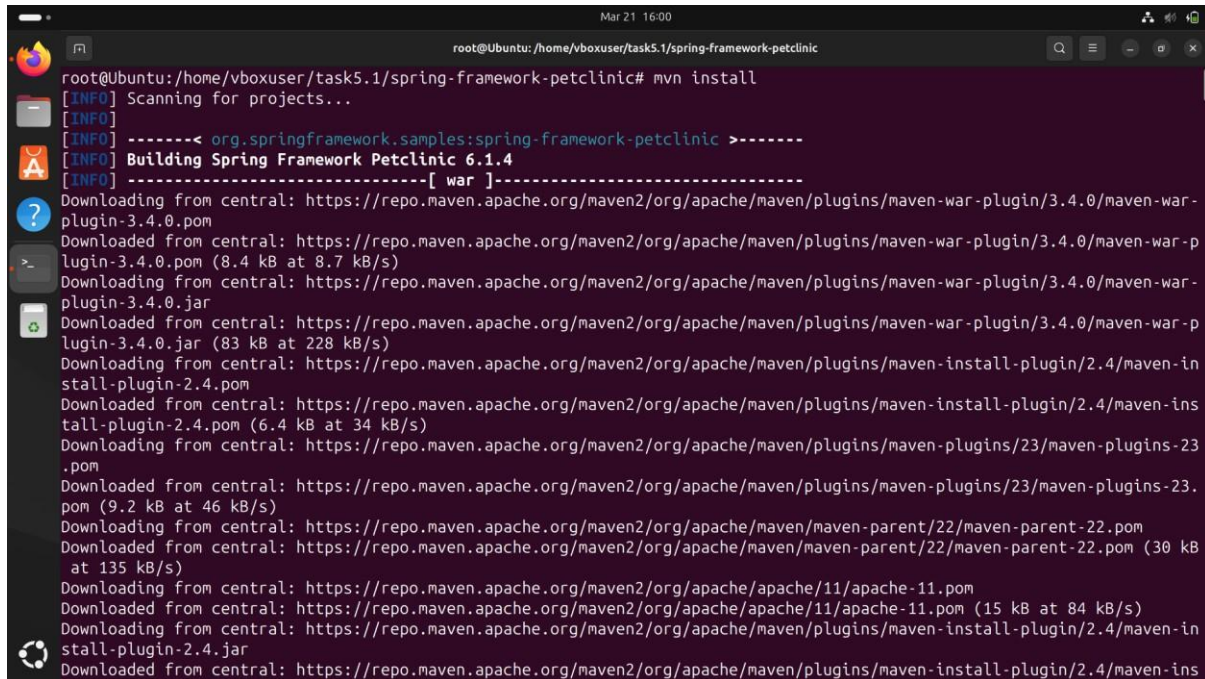
4. Install Maven

Installs Apache Maven, required for building the Spring Boot application.

Code:

```
sudo apt install maven
```

Screenshot:

A terminal window titled 'root@Ubuntu: /home/vboxuser/task5.1/spring-framework-petclinic' showing the output of the 'mvn install' command. The output includes status messages like '[INFO] Scanning for projects...' and '[INFO] Building Spring Framework Petclinic 6.1.4'. It also shows the download progress of various Maven plugins from the central repository, such as 'maven-war-plugin/3.4.0', 'maven-install-plugin/2.4', and 'maven-parent/22'.

```
root@Ubuntu: /home/vboxuser/task5.1/spring-framework-petclinic# mvn install
[INFO] Scanning for projects...
[INFO]
[INFO] -----< org.springframework.samples:spring-framework-petclinic >-----
[INFO] Building Spring Framework Petclinic 6.1.4
[INFO] -----[ war ]-----
[INFO]
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-war-plugin/3.4.0/maven-war-plugin-3.4.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-war-plugin/3.4.0/maven-war-plugin-3.4.0.pom (8.4 kB at 8.7 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-war-plugin/3.4.0/maven-war-plugin-3.4.0.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-war-plugin/3.4.0/maven-war-plugin-3.4.0.jar (83 kB at 228 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/2.4/maven-install-plugin-2.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/2.4/maven-install-plugin-2.4.pom (6.4 kB at 34 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/23/maven-plugins-23.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/23/maven-plugins-23.pom (9.2 kB at 46 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/22/maven-parent-22.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/22/maven-parent-22.pom (30 kB at 135 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/apache/11/apache-11.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/apache/11/apache-11.pom (15 kB at 84 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/2.4/maven-install-plugin-2.4.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/2.4/maven-install-plugin-2.4.jar
```

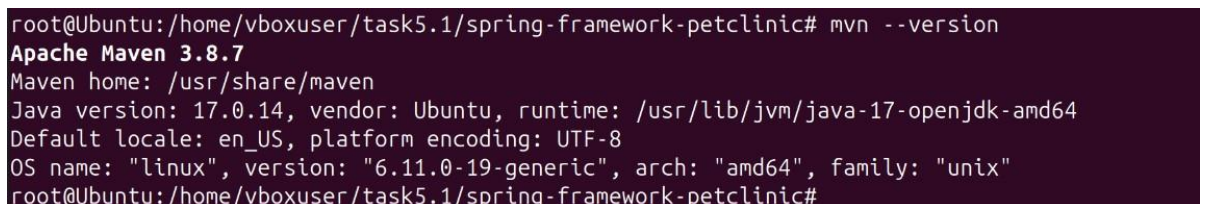
5. Verify Maven Installation

Checks if Maven is installed correctly and displays the version.

Code:

```
mvn --version
```

ScreenShot:

A terminal window showing the output of the 'mvn --version' command. It displays the Apache Maven version as 3.8.7, the Maven home directory as /usr/share/maven, the Java version as 17.0.14, and the OS as Linux.

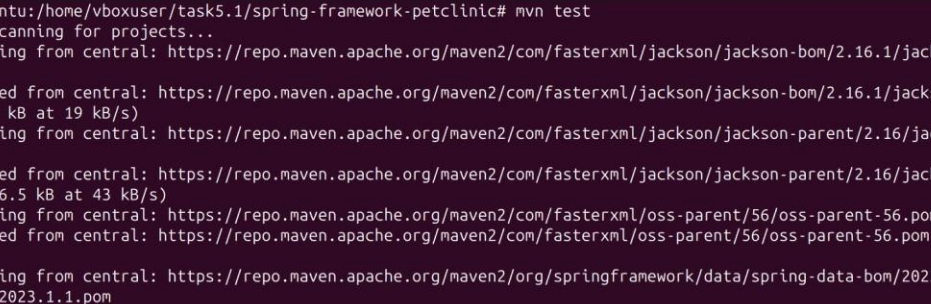
```
root@Ubuntu: /home/vboxuser/task5.1/spring-framework-petclinic# mvn --version
Apache Maven 3.8.7
Maven home: /usr/share/maven
Java version: 17.0.14, vendor: Ubuntu, runtime: /usr/lib/jvm/java-17-openjdk-amd64
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "6.11.0-19-generic", arch: "amd64", family: "unix"
root@Ubuntu: /home/vboxuser/task5.1/spring-framework-petclinic#
```

Executes unit tests to ensure the application works correctly.

Code:

```
mvn test
```

Screenshot:



```
root@Ubuntu: /home/vboxuser/task5.1/spring-framework-petclinic
root@Ubuntu: /home/vboxuser/task5.1/spring-framework-petclinic# mvn test
[INFO] Scanning for projects...
Downloading from central: https://repo.maven.apache.org/maven2/com/fasterxml/jackson/jackson-bom/2.16.1/jackson-bom-2.16.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/com/fasterxml/jackson/jackson-bom/2.16.1/jackson-bom-2.16.1.pom (18 kB at 19 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/com/fasterxml/jackson/jackson-parent/2.16/jackson-parent-2.16.pom
Downloaded from central: https://repo.maven.apache.org/maven2/com/fasterxml/jackson/jackson-parent/2.16/jackson-parent-2.16.pom (6.5 kB at 43 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/com/fasterxml/oss-parent/56/oss-parent-56.pom
Downloaded from central: https://repo.maven.apache.org/maven2/com/fasterxml/oss-parent/56/oss-parent-56.pom (24 kB at 139 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/data/spring-data-bom/2023.1.1/spring-data-bom-2023.1.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/data/spring-data-bom/2023.1.1/spring-data-bom-2023.1.1.pom (5.5 kB at 46 kB/s)
[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/jacoco/jacoco-maven-plugin/0.8.11/jacoco-maven-plugin-0.8.11.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/jacoco/jacoco-maven-plugin/0.8.11/jacoco-maven-plugin-0.8.11.pom (4.2 kB at 28 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/jacoco/org.jacoco.build/0.8.11/org.jacoco.build-0.8.11.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/jacoco/org.jacoco.build/0.8.11/org.jacoco.build-0.8.11.pom (44 kB at 243 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/ow2/asm/asm-bom/9.6/asm-bom-9.6.pom
```

7. Clean and Build the Application

mvn clean: Cleans previous builds.

mvn install: Compiles and packages the application.

mvn package: Generates the final JAR/WAR file in the **target/** directory.

Code:

```
mvn clean
```

```
mvn install
```

mvn package

Screenshot:


```
Mar 21 15:59
root@Ubuntu: /home/vboxuser/task5.1/spring-framework-petclinic# mvn clean
[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 3.4 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/22/maven-plugins-22.pom (13 kB at 65 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/21/maven-parent-21.pom (26 kB at 133 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/apache/10/apache-10.pom (15 kB at 89 kB/s)
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 129 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 (1.5 kB at 8.2 kB/s)
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 56 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/5/maven-parent-5.pom (15 kB at 80 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/apache/3/apache-3.pom (3.4 kB at 21 kB/s)
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 24 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/sonatype/spice/spice-parent/16/spice-parent-16.pom (8.4 kB at 44 kB/s)
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 58 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 355 kB/s)
[INFO] Deleting /home/vboxuser/task5.1/spring-framework-petclinic/target
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 4.392 s
[INFO] Finished at: 2025-03-21T15:59:18Z
[INFO] -----
root@Ubuntu: /home/vboxuser/task5.1/spring-framework-petclinic#
```

```
Mar 21 16:00
root@Ubuntu: /home/vboxuser/task5.1/spring-framework-petclinic# mvn install
[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-war-plugin/3.4.0/maven-war-plugin-3.4.0.pom (8.4 kB at 8.7 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-war-plugin/3.4.0/maven-war-plugin-3.4.0.jar (83 kB at 228 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-war-plugin/3.4.0/maven-war-plugin-3.4.0.jar (83 kB at 228 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/2.4/maven-install-plugin-2.4.pom (6.4 kB at 34 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/23/maven-plugins-23.pom (9.2 kB at 46 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/22/maven-parent-22.pom (30 kB at 135 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/apache/11/apache-11.pom (15 kB at 84 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/2.4/maven-install-plugin-2.4.jar (226 kB at 355 kB/s)
```

```
root@Ubuntu: /home/vboxuser/task5.1/spring-framework-petclinic# mvn package
[INFO] Scanning for projects...
[INFO] -----< org.springframework.samples:spring-framework-petclinic >-----
[INFO] Building Spring Framework Petclinic 6.1.4
[INFO] -----[ war ]-----
[INFO] --- maven-enforcer-plugin:3.4.1:enforce (enforce-maven) @ spring-framework-petclinic ---
[INFO] Rule 0: org.apache.maven.enforcer.rules.version.RequireMavenVersion passed
[INFO] --- jacoco-maven-plugin:0.8.11:prepare-agent (prepare-agent) @ spring-framework-petclinic ---
[INFO] argLine set to -javaagent:/root/.m2/repository/org/jacoco/org.jacoco.agent/0.8.11/org.jacoco.agent-0.8.11-runtime.jar=destfile=/home/vboxuser/task5.1/spring-framework-petclinic/target/jacoco.exec
[INFO] --- maven-resources-plugin:3.3.1:resources (default-resources) @ spring-framework-petclinic ---
[INFO] Copying 21 resources from src/main/resources to target/classes
[INFO] --- maven-compiler-plugin:3.11.0:compile (default-compile) @ spring-framework-petclinic ---
[INFO] Nothing to compile - all classes are up to date
[INFO] --- maven-resources-plugin:3.3.1:testResources (default-testResources) @ spring-framework-petclinic ---
[INFO] Copying 11 resources from src/test/java to target/test-classes
[INFO] Copying 1 resource from src/test/resources to target/test-classes
[INFO] --- maven-compiler-plugin:3.11.0:testCompile (default-testCompile) @ spring-framework-petclinic ---
[INFO] Nothing to compile - all classes are up to date
[INFO] --- maven-surefire-plugin:3.2.3:test (default-test) @ spring-framework-petclinic ---
[INFO] Using auto detected provider org.apache.maven.surefire.junitplatform.JUnitPlatformProvider
[INFO]
```

8. Verify the Built Application

Navigates to the `target` folder where the compiled application is stored.

Code:

```
cd target
ls
cd ..
```

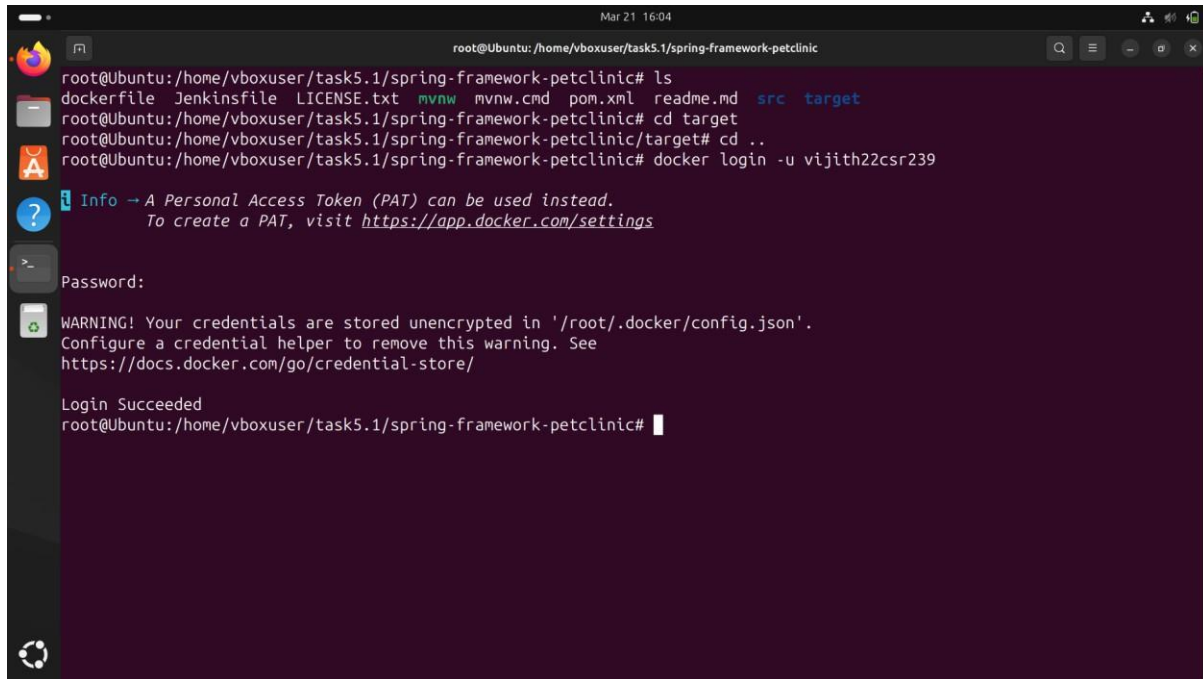
9. Login to Docker

Logs into Docker Hub to push container images.

Code:

```
docker login -u vijith22csr239
```

Screenshot:

A terminal window screenshot showing the Docker login process. The terminal title is 'root@Ubuntu: /home/vboxuser/task5.1/spring-framework-petclinic'. The user runs 'ls' showing files like 'dockerfile', 'Jenkinsfile', 'LICENSE.txt', 'mvnw', 'mvnw.cmd', 'pom.xml', 'readme.md', 'src', and 'target'. Then they run 'cd target' and 'cd ..'. Finally, they run 'docker login -u vijith22csr239'. The terminal shows an info message about PAT, a password prompt, a warning about unencrypted credentials, and a successful login message.

```
root@Ubuntu: /home/vboxuser/task5.1/spring-framework-petclinic# ls
dockerfile Jenkinsfile LICENSE.txt mvnw mvnw.cmd pom.xml readme.md src target
root@Ubuntu: /home/vboxuser/task5.1/spring-framework-petclinic# cd target
root@Ubuntu: /home/vboxuser/task5.1/spring-framework-petclinic/target# cd ..
root@Ubuntu: /home/vboxuser/task5.1/spring-framework-petclinic# docker login -u vijith22csr239

Info → A Personal Access Token (PAT) can be used instead.
To create a PAT, visit https://app.docker.com/settings

Password:
WARNING! Your credentials are stored unencrypted in '/root/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/

Login Succeeded
root@Ubuntu: /home/vboxuser/task5.1/spring-framework-petclinic#
```

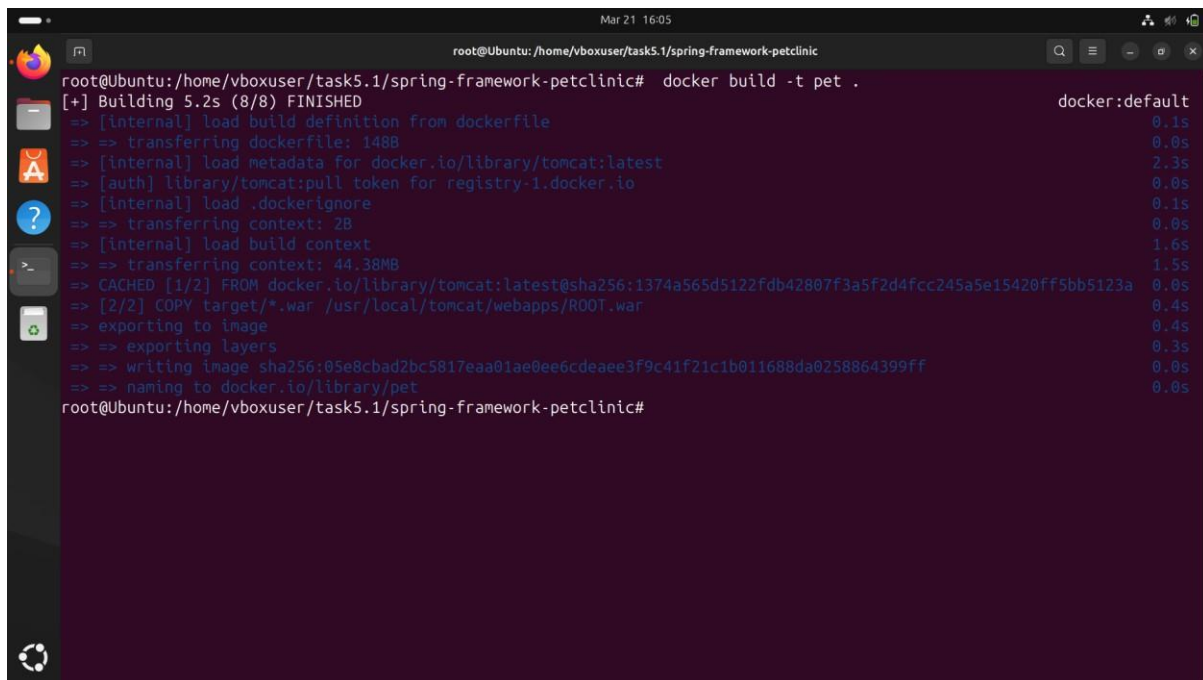
10. Build Docker Image

Builds a Docker image with the tag `pet` from the project directory.

Code:

```
docker build -t pet .
```

Screenshot:



```
root@Ubuntu: /home/vboxuser/task5.1/spring-framework-petclinic# docker build -t pet .
[+] Building 5.2s (8/8) FINISHED
=> [internal] load build definition from dockerfile                                docker:default
=> => transferring dockerfile: 1488                                              0.1s
=> [internal] load metadata for docker.io/library/tomcat:latest                  0.0s
=> [auth] library/tomcat:pull token for registry-1.docker.io                    2.3s
=> [internal] load .dockerignore                                                0.0s
=> => transferring context: 2B                                                  0.1s
=> [internal] load build context                                                0.0s
=> => transferring context: 44.38MB                                             1.6s
=> CACHED [1/2] FROM docker.io/library/tomcat:latest@sha256:1374a565d5122fdb42807f3a5f2d4fcc245a5e15420ff5bb5123a 1.5s
=> [2/2] COPY target/*.war /usr/local/tomcat/webapps/ROOT.war                  0.0s
=> exporting to image                                                            0.4s
=> => exporting layers                                                         0.3s
=> writing image sha256:05e8cbad2bc5817eaa01ae0ee6cdeae3f9c41f21c1b011688da0258864399ff 0.0s
=> => naming to docker.io/library/pet                                         0.0s
root@Ubuntu: /home/vboxuser/task5.1/spring-framework-petclinic#
```


11. Tag and Push Image to Docker Hub

Tags the image for Docker Hub.

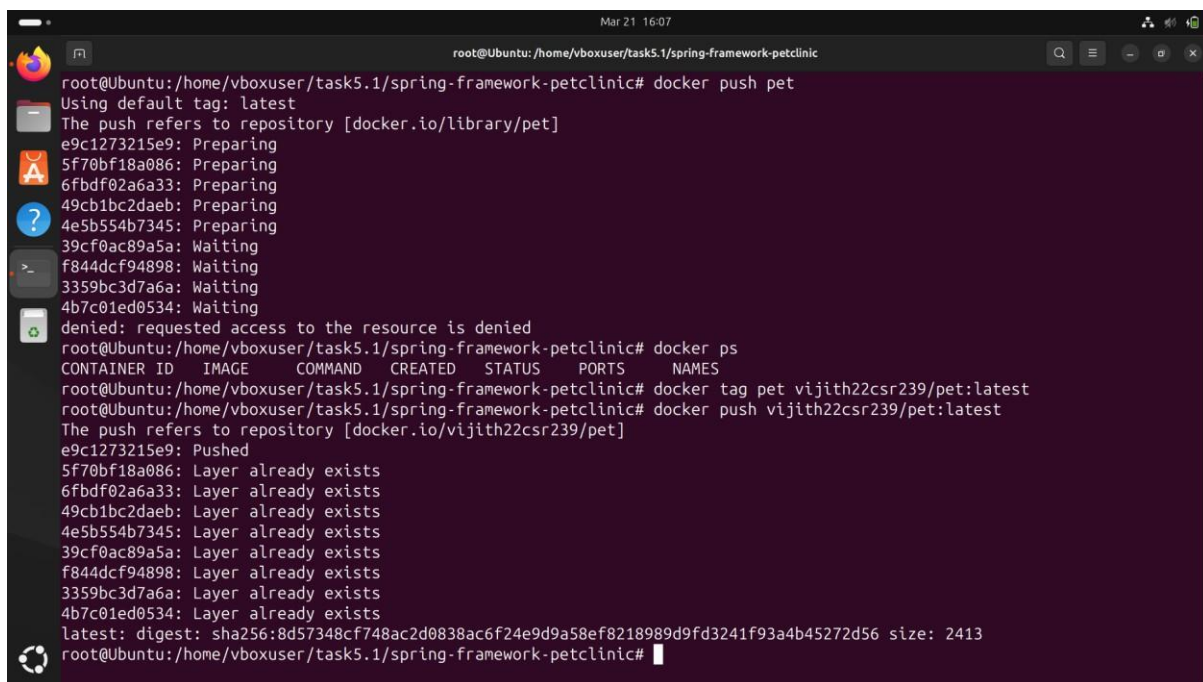
Pushes the image to your Docker Hub repository.

Code:

```
docker tag pet vijith22csr239/pet:latest
```

```
docker push vijith22csr239/pet:latest
```

Screenshot:

A terminal window screenshot showing the execution of Docker commands. The terminal title is 'root@Ubuntu: /home/vboxuser/task5.1/spring-framework-petclinic'. The user runs 'docker push pet', which fails with a 'denied: requested access to the resource is denied' error. Then, the user runs 'docker ps', showing a table of containers. Next, the user runs 'docker tag pet vijith22csr239/pet:latest'. Finally, the user runs 'docker push vijith22csr239/pet:latest', which succeeds, showing the image layers being pushed and the final digest and size of the image.

```
root@Ubuntu: /home/vboxuser/task5.1/spring-framework-petclinic# docker push pet
Using default tag: latest
The push refers to repository [docker.io/library/pet]
e9c1273215e9: Preparing
5f70bf18a086: Preparing
6fbdf02a6a33: Preparing
49cb1bc2daeb: Preparing
4e5b554b7345: Preparing
39cf0ac89a5a: Waiting
f844dcf94898: Waiting
3359bc3d7a6a: Waiting
4b7c01ed0534: Waiting
denied: requested access to the resource is denied
root@Ubuntu: /home/vboxuser/task5.1/spring-framework-petclinic# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS   NAMES
root@Ubuntu: /home/vboxuser/task5.1/spring-framework-petclinic# docker tag pet vijith22csr239/pet:latest
root@Ubuntu: /home/vboxuser/task5.1/spring-framework-petclinic# docker push vijith22csr239/pet:latest
The push refers to repository [docker.io/vijith22csr239/pet]
e9c1273215e9: Pushed
5f70bf18a086: Layer already exists
6fbdf02a6a33: Layer already exists
49cb1bc2daeb: Layer already exists
4e5b554b7345: Layer already exists
39cf0ac89a5a: Layer already exists
f844dcf94898: Layer already exists
3359bc3d7a6a: Layer already exists
4b7c01ed0534: Layer already exists
latest: digest: sha256:8d57348cf748ac2d0838ac6f24e9d9a58ef8218989d9fd3241f93a4b45272d56 size: 2413
root@Ubuntu: /home/vboxuser/task5.1/spring-framework-petclinic#
```

12. Start Minikube

Starts a Minikube cluster for Kubernetes.

Checks if Minikube is running properly.

Code:

```
minikube start
```

```
minikube status
```

13. Verify Kubernetes Nodes

Lists available Kubernetes nodes.

Code:

```
kubectl get nodes
```

ScreenShot:



```
mathav@Mathav-Desktop: ~$ 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 ...
🔄 Restarting existing docker container for "minikube" ...
❗ Failing to connect to https://registry.k8s.io/ from inside the minikube container
💡 To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
🌐 Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
🔍 Verifying Kubernetes components...
   ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟 Enabled addons: default-storageclass, storage-provisioner
🎉 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

14. Deploy the Application on Kubernetes & Expose the Application

Creates a Kubernetes deployment using your Docker image.

Exposes the deployment as a service, making it accessible via Minikube.

Lists all running pods to verify the deployment is successful.


Code:

```
kubectl create deployment pet --image=vijith22csr239/pet --port=8080
```

```
kubectl expose deployment pet --port=8080 --type=NodePort
```

```
kubectl get pods
```

Screenshot:



```

NAMESPACE   NAME   TARGET PORT   URL
default     pod1   80            http://192.168.49.2:30592
★ Starting tunnel for service pod1.
NAMESPACE   NAME   TARGET PORT   URL
default     pod1                http://127.0.0.1:33121
🐞 Opening service default/pod1 in default browser...
🌟 http://127.0.0.1:33121
! Because you are using a Docker driver on linux, the terminal needs to be open to run it.
```

16. Access the Application

Opens the application in the browser via Minikube.

Code:

```
minikube service pet
```

Screenshot:

