SANMATHI S

22ISR045

The process automates the steps to:

1. **Build** a Java app using Maven.
2. **Create** a Docker image containing the app.
3. **Push** the image to a container registry (like DockerHub).
4. **Deploy** the image to a Kubernetes cluster.

Required Files

- **Dockerfile**: Defines how the Docker image is built.
- **deploymentservice.yaml**: Kubernetes deployment configuration.
- **Jenkinsfile**: Automates the CI/CD pipeline.

Jenkins Pipeline Steps

Checkout Code → Clone the GitHub repository.

Build with Maven → Run `mvn clean package -DskipTests`.

Build Docker Image `docker build -t ekartimage -f docker/Dockerfile .`

Push Image to DockerHub → Authenticate and push using

```
>> docker login -u sanmathisedhupathi -p <token>

>> docker tag ekartimage sanmathisedhupathi/myportfolio:ekart

>> docker push sanmathisedhupathi/myportfolio:ekart
```

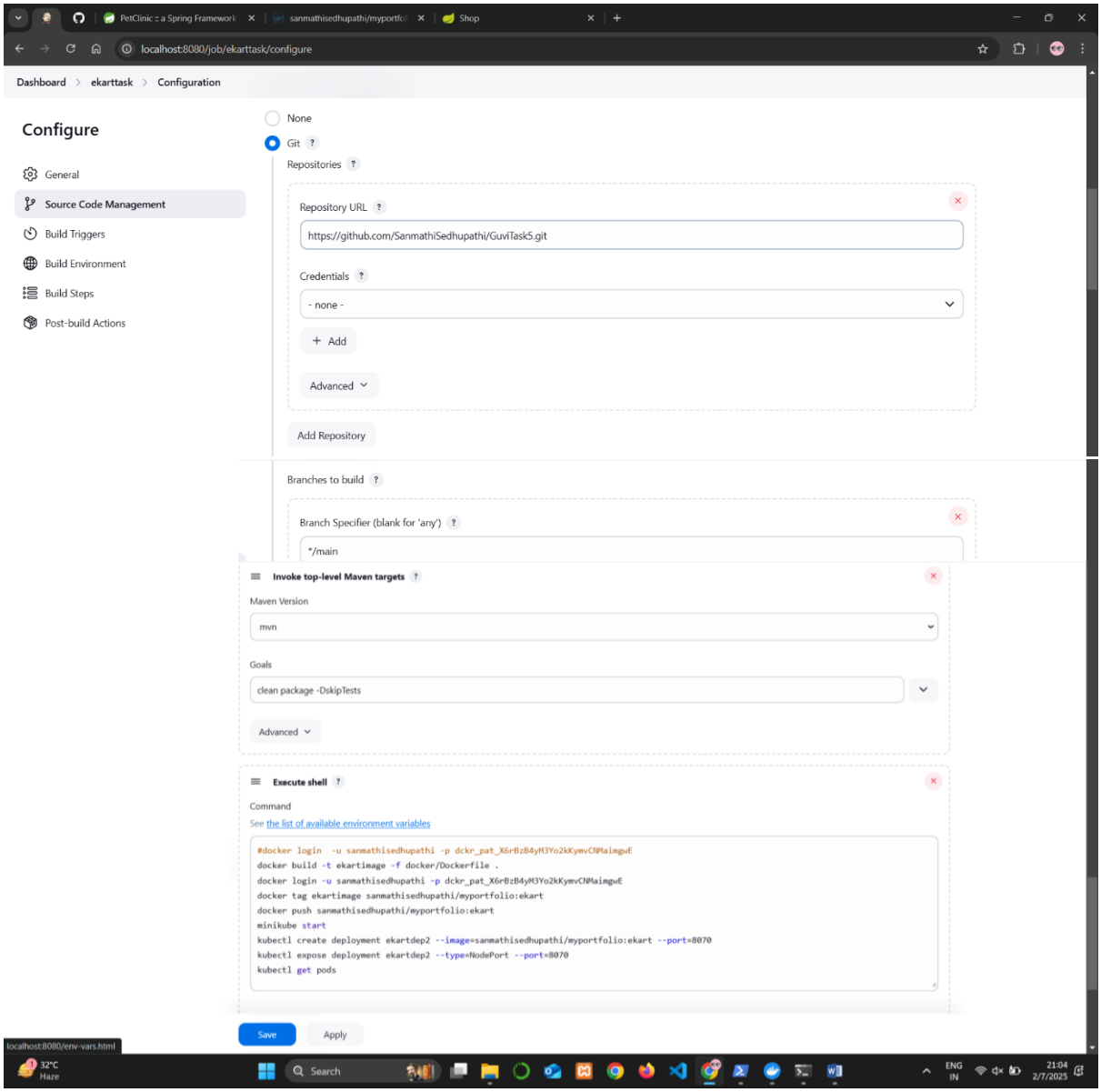
Deploy to Kubernetes Run

```
>> minikube start

>> kubectl create deployment ekartdep2 --image=sanmathisedhupathi/myportfolio:ekart --port=8070

>> kubectl expose deployment ekartdep2 --type=NodePort --port=8070

>> kubectl get pods
```

Get service details:

```
>>kubectl get svc my-java-app-service
```

i. Jenkins


PetClinic :: a Spring Framework
sanmathisedhupathi/myportfol
Shop
localhost:8080/job/ekarttask/configure
Dashboard
ekarttask
Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
None
Git
Repositories
Repository URL
https://github.com/SanmathiSedhupathi/GuviTask5.git
Credentials
- none -
Add
Advanced
Add Repository
Branches to build
Branch Specifier (blank for 'any')
*/main
Invoke top-level Maven targets
Maven Version
mvn
Goals
clean package -DskipTests
Advanced
Execute shell
Command
See the list of available environment variables
#docker login -u sanmathisedhupathi -p dckr_pat_X6rBzB4yM3Yo2kKymvCNMaimgwE
docker build -t ekartimage -f docker/Dockerfile .
docker login -u sanmathisedhupathi -p dckr_pat_X6rBzB4yM3Yo2kKymvCNMaimgwE
docker tag ekartimage sanmathisedhupathi/myportfolio:ekart
docker push sanmathisedhupathi/myportfolio:ekart
minikube start
kubectl create deployment ekartdep2 --image=sanmathisedhupathi/myportfolio:ekart --port=8070
kubectl expose deployment ekartdep2 --type=NodePort --port=8070
kubectl get pods
Save
Apply
localhost:8080/env-vars.html
32°C
Haze
Search
ENG
IN
21:04
2/7/2025

ii. Console Output :BUILD SUCCESS

```
[INFO]
[INFO] --- compiler:3.1:testCompile (default-testCompile) @ shopping-cart ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to /var/lib/jenkins/workspace/ekarttask/target/test-classes
[INFO]
[INFO] --- surefire:2.18.1:test (default-test) @ shopping-cart ---
[INFO] Tests are skipped.
[INFO]
[INFO] --- jar:2.6:jar (default-jar) @ shopping-cart ---
[INFO] Building jar: /var/lib/jenkins/workspace/ekarttask/target/shopping-cart-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot:1.5.3.RELEASE:repackage (default) @ shopping-cart ---
[INFO]
[INFO] --- jacoco:0.8.6:report (jacoco-site) @ shopping-cart ---
[INFO] Skipping JaCoCo execution due to missing execution data file.
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  3.938 s
[INFO] Finished at: 2025-02-07T10:57:56Z
[INFO] ------------------------------------------------------------------------
[ekarttask] $ /bin/sh -xe /tmp/jenkins3226079886023033680.sh
+ docker build -t ekartimage -f docker/Dockerfile .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  48.41MB
```

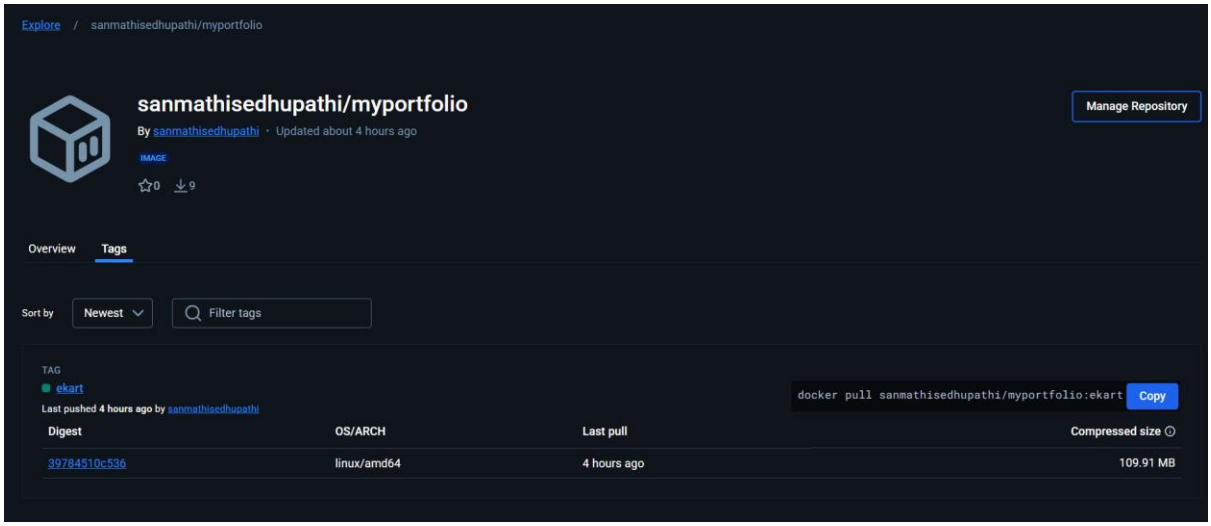
iii. Pushed image to docker hub registry

```
Login Succeeded
+ docker tag ekartimage sanmathisedhupathi/myportfolio:ekart
+ docker push sanmathisedhupathi/myportfolio:ekart
The push refers to repository [docker.io/sanmathisedhupathi/myportfolio]
a93703eac774: Preparing
01bca7cb4826: Preparing
a8cc3712c14a: Preparing
cd7100a72410: Preparing
01bca7cb4826: Layer already exists
cd7100a72410: Layer already exists
a8cc3712c14a: Layer already exists
a93703eac774: Pushed
```

iv. Deploying the image to a Kubernetes cluster

```
+ 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! kubectl is now configured to use "minikube" cluster and "default" namespace by default
+ kubectl create deployment ekartdep2 --image=sanmathisedhupathi/myportfolio:ekart --port=8070
deployment.apps/ekartdep2 created
+ kubectl expose deployment ekartdep2 --type=NodePort --port=8070
service/ekartdep2 exposed
+ kubectl get pods
NAME                                     READY   STATUS             RESTARTS      AGE
ekartdep-d979dfd56-rlhb2                 1/1     Running            2 (80s ago)   15m
react-portfolio-deployment-5d97b58bb-2p92l   0/1     ImagePullBackOff   0             26h
Finished: SUCCESS
```

v. Pushed image into dockerhub registry



vi. Expose a service running in Kubernetes

```
root@DESKTOP-QHV714I:~# sudo su jenkins
jenkins@DESKTOP-QHV714I:/root$ 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 ...
❗  Failing to connect to https://registry.k8s.io/ from both inside the minikube container and host machine
💡  To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networki
ng/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
jenkins@DESKTOP-QHV714I:/root$ kubectl get svc
NAME                         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
ekartdep                     NodePort    10.105.211.68   <none>        80:32470/TCP     4h24m
ekartdep2                    NodePort    10.99.57.185    <none>        8070:30434/TCP   4h8m
kubernetes                   ClusterIP   10.96.0.1       <none>        443/TCP          33h
react-portfolio-deployment   NodePort    10.96.139.56    <none>        80:31953/TCP     30h
jenkins@DESKTOP-QHV714I:/root$ minikube service ekartdep2
|-----------|-----------|-------------|--------------------------|
| NAMESPACE |   NAME    | TARGET PORT |           URL            |
|-----------|-----------|-------------|--------------------------|
| default   | ekartdep2 |        8070 | http://192.168.49.2:30434 |
|-----------|-----------|-------------|--------------------------|
🎉  Opening service default/ekartdep2 in default browser...
👉  http://192.168.49.2:30434
```

vii. opens the service in the default browser