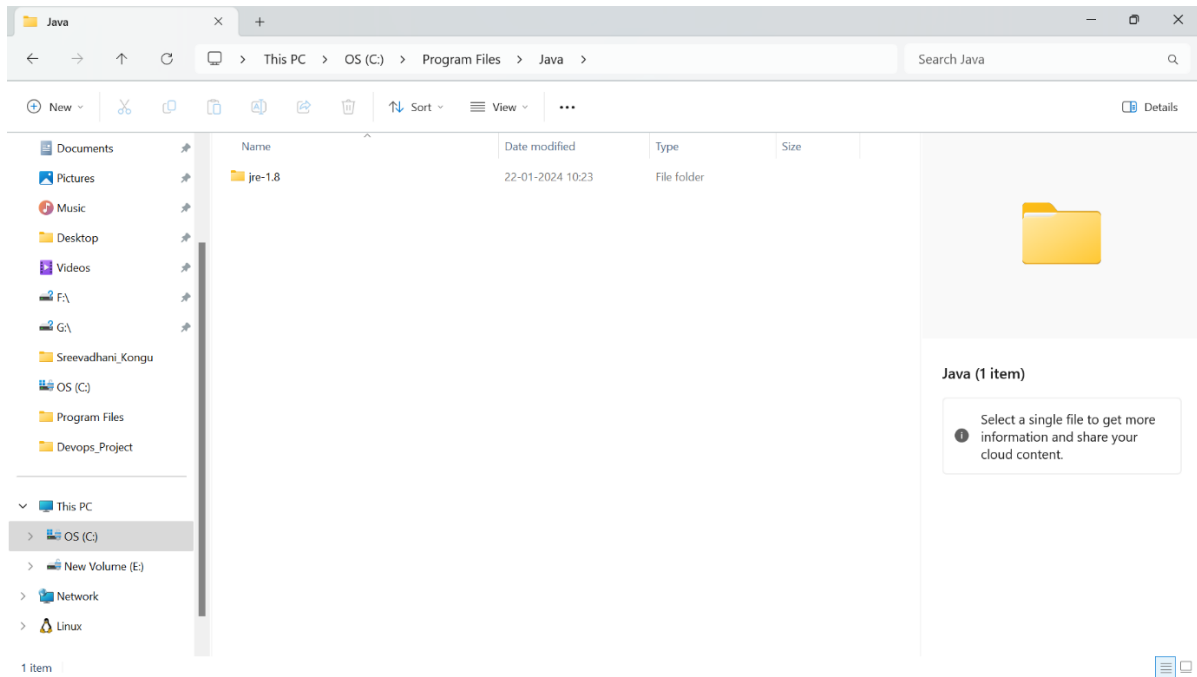


## TASK 5

**Step 1:** Install Java 17 and save it under **Program Files**. If already installed move on to next step.



**Step 2:** Install Maven from its official website

```
sree_ubuntu@sree: ~
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> wsl
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.153.1-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:   https://landscape.canonical.com
 * Support:      https://ubuntu.com/advantage

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.
   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

This message is shown once a day. To disable it please create the
/home/sree_ubuntu/.hushlogin file.
sree_ubuntu@sree: /mnt/c:/WINDOWS/system32$ cd
sree_ubuntu@sree:~$ mvn --version
Apache Maven 3.6.3
Maven home: /usr/share/maven
Java version: 17.0.13, vendor: Ubuntu, runtime: /usr/lib/jvm/java-17-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "5.15.153.1-microsoft-standard-WSL2", arch: "amd64", family: "unix"
sree_ubuntu@sree:~$
```

### Step 3: Install Apache Tomcat 9 from its official website.

```
sree_ubuntu@sree: ~
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> wsl
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.153.1-microsoft-standard-WSL2 x86_64)

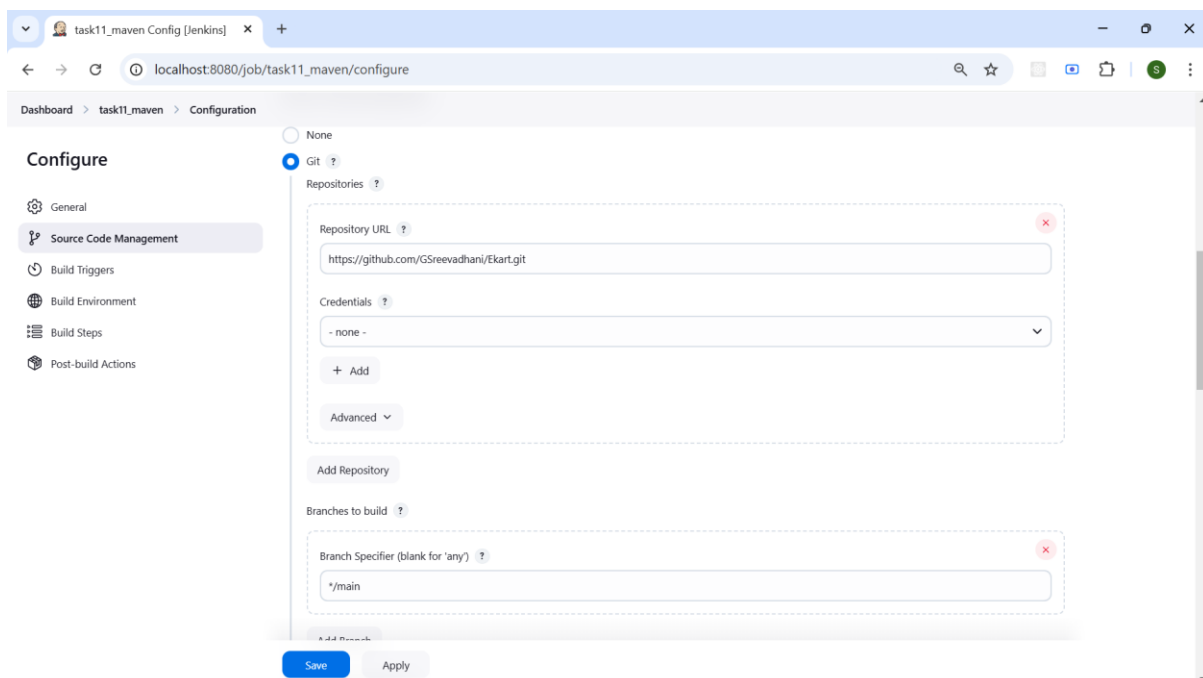
 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.
   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

This message is shown once a day. To disable it please create the
/home/sree_ubuntu/.hushlogin file.
sree_ubuntu@sree:/mnt/c:/WINDOWS/system32$ cd
sree_ubuntu@sree:~$ mvn --version
Apache Maven 3.6.3
Maven home: /usr/share/maven
Java version: 17.0.13, vendor: Ubuntu, runtime: /usr/lib/jvm/java-17-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "5.15.153.1-microsoft-standard-WSL2", arch: "amd64", family: "unix"
sree_ubuntu@sree:~$
```

### Step 4: Fork the Github repo named **Ekart** and create a new freestyle job in Jenkins.

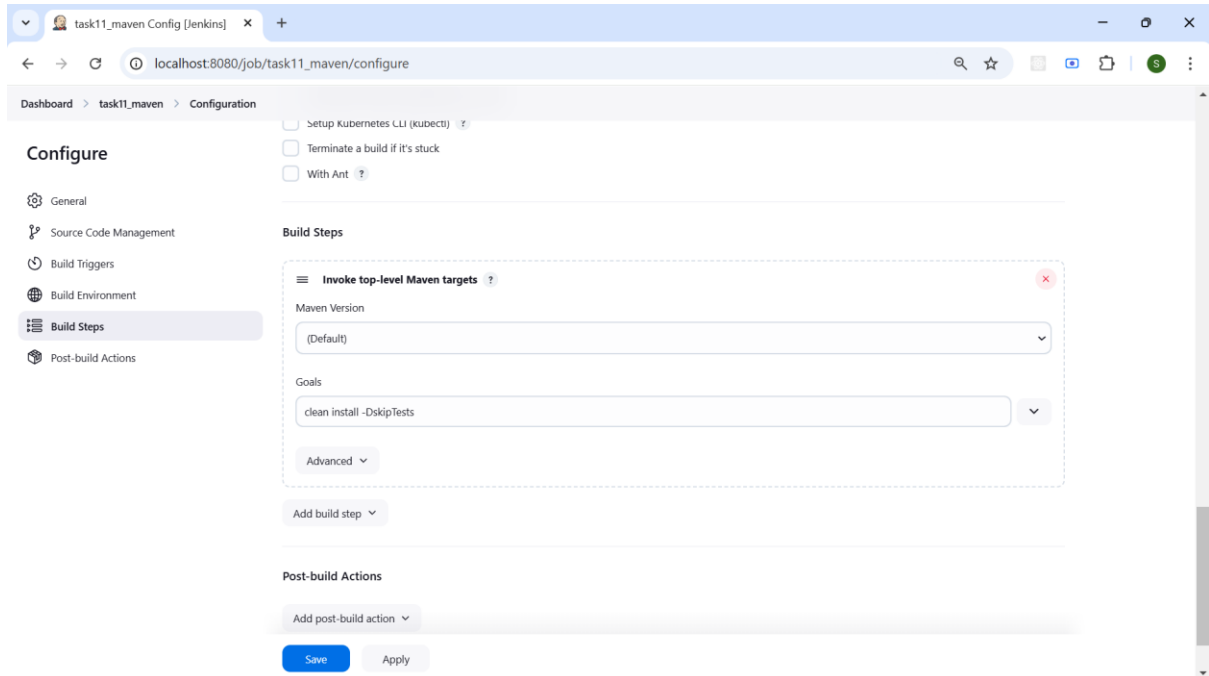
In **source code management**, give github repo URL and change the builds specifier according to the branch you want to build(here, \*/main)



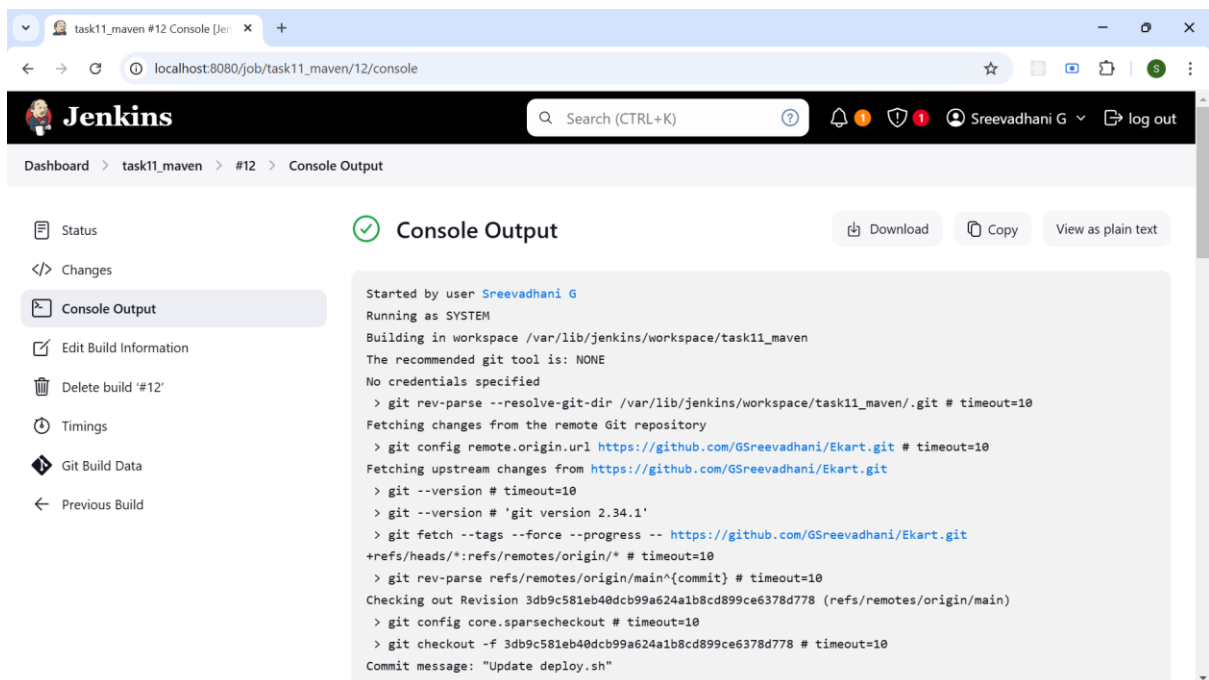
**Step 5: Go to Build Steps and click on Invoke top-level Maven targets.**

In Invoke top-level Maven Targets,

Goals : clean install -DskipTests (generates .war file)



**Step 6: Click on Save and Build Now**



**Step 7: Use `docker build -t maven-img -f docker/Dockerfile` to build a docker image from Dockerfile of name `maven-img`.**

```
sree_ubuntu@sree:/var/lib/jenkins/workspace/task11_maven$ docker build -t maven-img -f docker/Dockerfile .
failed to fetch metadata: fork/exec /usr/local/lib/docker/cli-plugins/docker-buildx: no such file or directory

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 48.43MB
Step 1/6 : FROM openjdk:8u151-jdk-alpine3.7
--> cc2179b8f842
Step 2/6 : EXPOSE 8070
--> Using cache
--> 3de4f7ddcd13
Step 3/6 : ENV APP_HOME /usr/src/app
--> Using cache
--> bbc59dbd2fdb
Step 4/6 : COPY target/shopping-cart-0.0.1-SNAPSHOT.jar $APP_HOME/app.jar
--> 6eb05fd8587c
Step 5/6 : WORKDIR $APP_HOME
--> Running in 187d16a6d700
--> Removed intermediate container 187d16a6d700
--> 7859f45264dc
Step 6/6 : ENTRYPOINT exec java -jar app.jar
--> Running in bfe736d6acb5
--> Removed intermediate container bfe736d6acb5
--> f6ff010180b7
Successfully built f6ff010180b7
Successfully tagged maven-img:latest
sree_ubuntu@sree:/var/lib/jenkins/workspace/task11_maven$
```

**Step 8: Start the minikube.**

Use `kubectl create deployment <deployment_name> --image=<img_name> --port 8070` to deploy the deocker hub image in pod.

And `kubectl expose deployment <deployment_name> --type=NodePort --port=8070` to expose pod to slave node.

```
sree_ubuntu@sree:/var/lib/jenkins/workspace/task11_maven$ minikube start
minikube v1.35.0 on Ubuntu 22.04 (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" ...
* StartHost failed, but will try again: provision: get ssh host-port: get port 22 for "minikube": docker container inspect -f '{{(index (index .NetworkSettings.Ports "22/tcp") 0).HostPort}}' minikube: exit status 1
stdout:

stderr:
template parsing error: template: :1:4: executing "" at <index (index .NetworkSettings.Ports "22/tcp") 0>: error calling index: reflect: slice index out of range

* 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: default-storageclass, storage-provisioner

! /usr/local/bin/kubectl is version 1.30.2, which may have incompatibilities with Kubernetes 1.32.0.
  * Want kubectl v1.32.0? Try 'minikube kubectl -- get pods -A'
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
sree_ubuntu@sree:/var/lib/jenkins/workspace/task11_maven$ kubectl create deployment ekart-repo --image=sreevadhani/ekart-tag --port 8070
deployment.apps/ekart-repo created
sree_ubuntu@sree:/var/lib/jenkins/workspace/task11_maven$ kubectl expose deployment ekart-repo --type=NodePort --port=8070
service/ekart-repo exposed
sree_ubuntu@sree:/var/lib/jenkins/workspace/task11_maven$ kubectl get svc
NAME          TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
ekart-repo    NodePort      10.103.118.19   <none>           8070:31323/TCP   7s
kubernetes    ClusterIP     10.96.0.1       <none>           443/TCP          42h
sree_ubuntu@sree:/var/lib/jenkins/workspace/task11_maven$
```

## Step 9: Open the service.

```
sree_ubuntu@sree:/var/lib/jenkins/workspace/task11_maven$ docker pull sreevadhani/ekart-tag
Using default tag: latest
latest: Pulling from sreevadhani/ekart-tag
Digest: sha256:481f4bce94a0c19ebde856ac991de35c935f459cb2b972fdeafa2ac7d4ad9c63
Status: Image is up to date for sreevadhani/ekart-tag:latest
docker.io/sreevadhani/ekart-tag:latest
sree_ubuntu@sree:/var/lib/jenkins/workspace/task11_maven$ kubectl delete pod -l app=ekart-repo
pod "ekart-repo-8bbdd55c4-bcqhj" deleted
sree_ubuntu@sree:/var/lib/jenkins/workspace/task11_maven$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
ekart-repo-8bbdd55c4-5c2c8          1/1     Running   0           64s
sree_ubuntu@sree:/var/lib/jenkins/workspace/task11_maven$ minikube service ekart-repo

```

NAMESPACE	NAME	TARGET PORT	URL
default	ekart-repo	8070	http://192.168.58.2:31323

```

# Starting tunnel for service ekart-repo.

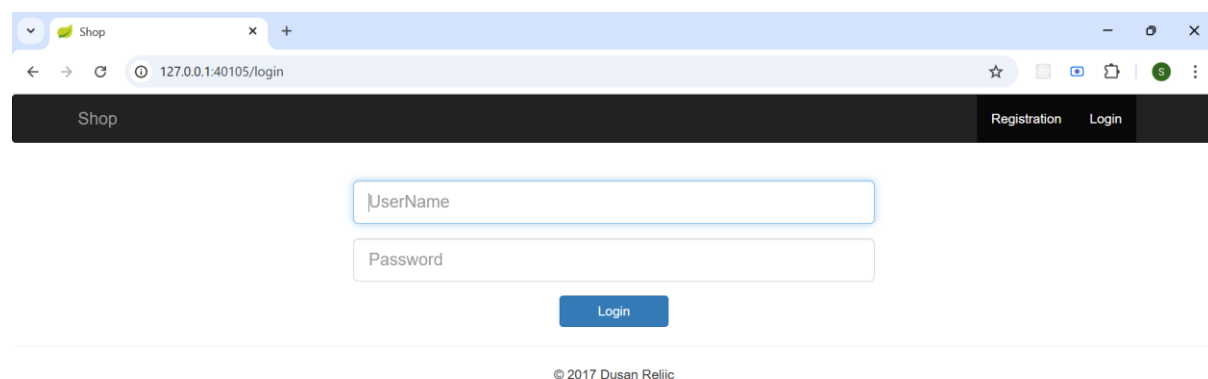
```

NAMESPACE	NAME	TARGET PORT	URL
default	ekart-repo		http://127.0.0.1:40105

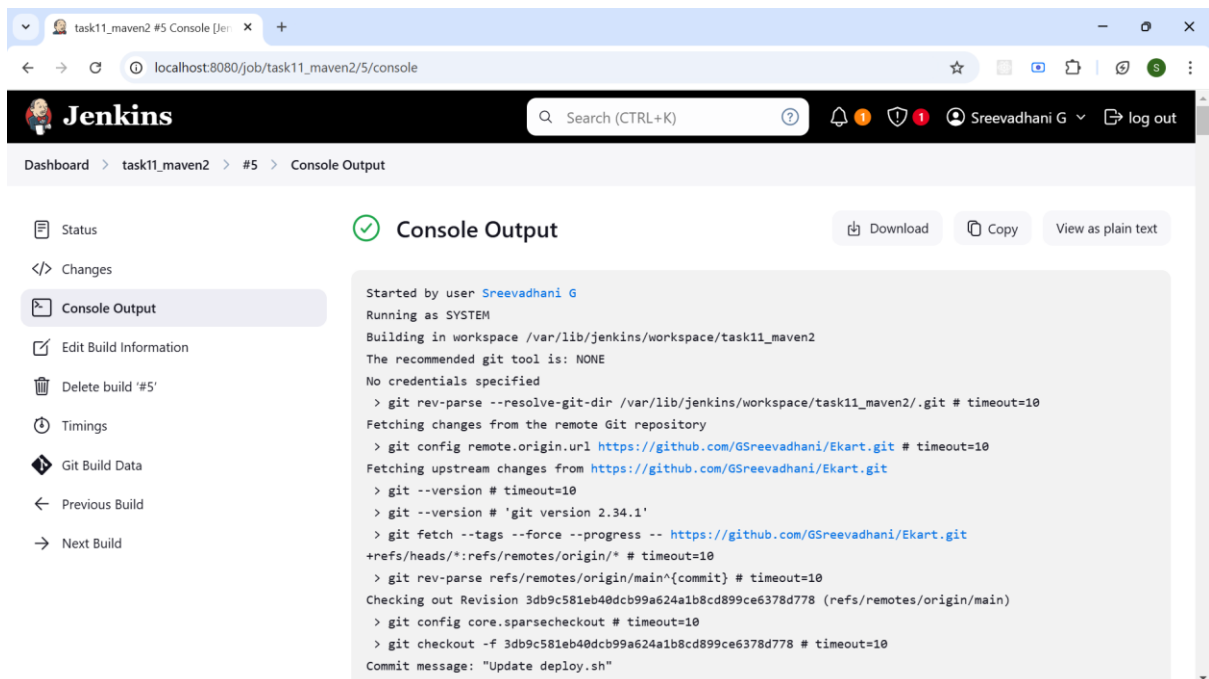
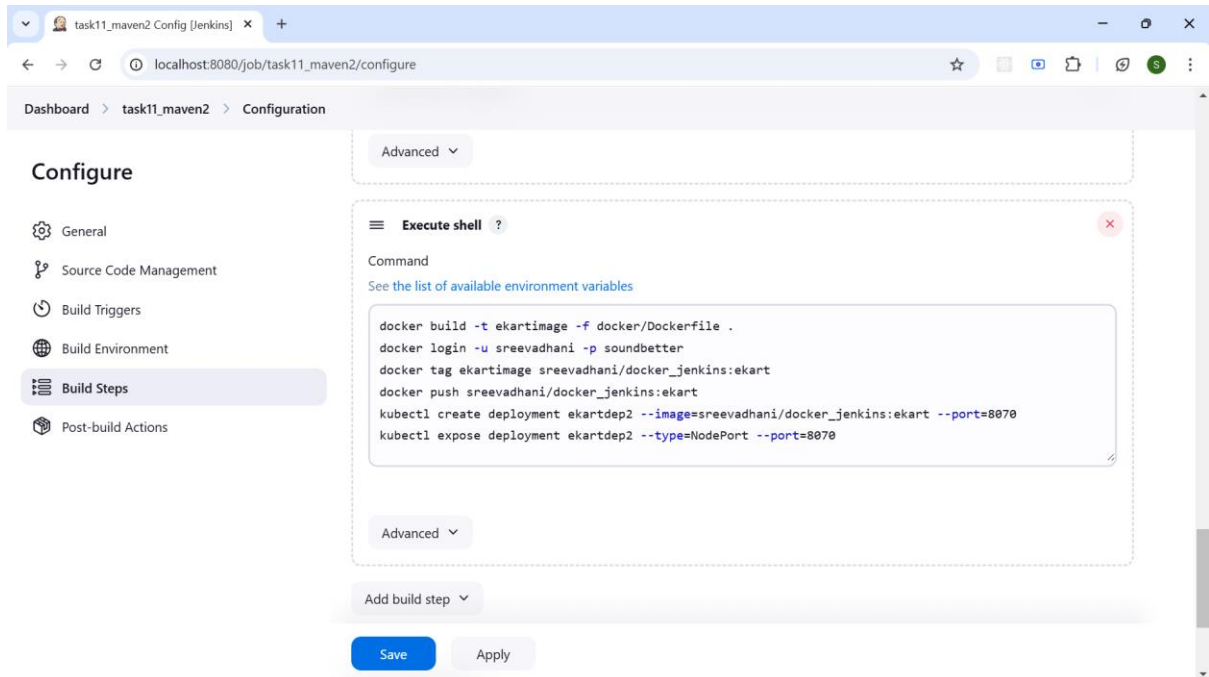
```

# Opening service default/ekart-repo in default browser...
🔗 http://127.0.0.1:40105
! Because you are using a Docker driver on linux, the terminal needs to be open to run it.

```



## BY ENTERING THE BUILD AND DEPLOYMENT COMMANDS IN THE EXECUTE SHELL OF JENKINS:



UserName

Password

Login