DEVOPS

ASSIGNMENT 3

STEPS:

Step 1: Clone the Repository

Step 2: Build the Docker Image

docker build -t e-commerce-app .

Step 3: Start Minikube

minikube start --force

minikube status

Step 4: Load the Docker Image into Minikube

minikube image load e-commerce-app

Verify the image is loaded:

minikube image list # Ensure "e-commerce-app" is listed

Step 5: Deploy the Application

kubectl apply -f deployment.yml

kubectl get deployments

kubectl get pods

If you need a NodePort service, apply it:

kubectl apply -f Nodeport.yaml

Step 6: Fix Image Pull Issues (if necessary)

If Kubernetes tries to pull the image from a registry instead of using the local image, patch the deployment kubectl patch deployment react-ecommerce-deployment --type='json' p='[{"op": "replace", "path": "/spec/template/spec/containers/0/imagePullPolicy", "value": "Never"}]'

Step 7: Expose the Service & Access the App

minikube ip

minikube service react-ecommerce-service

Step 8:

Push to Git:

git init # If not already initialized git

add Dockerfile deployment.yml

Nodeport.yaml

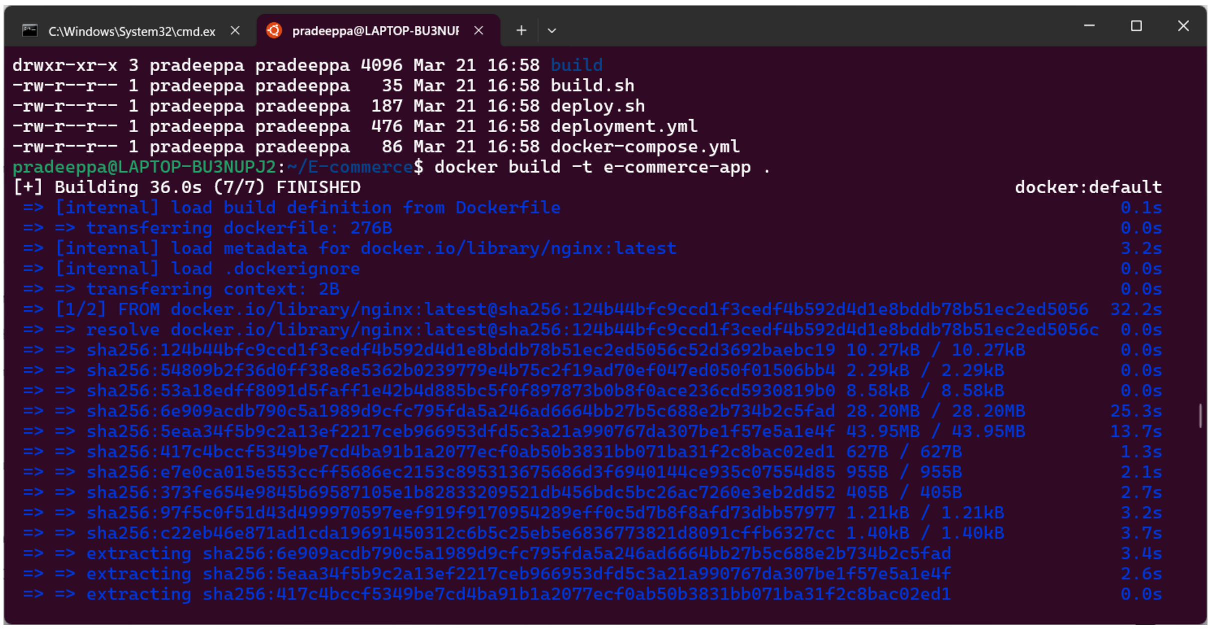
git commit -m "Kubernetes deployment for React e-commerce app"

git remote add origin

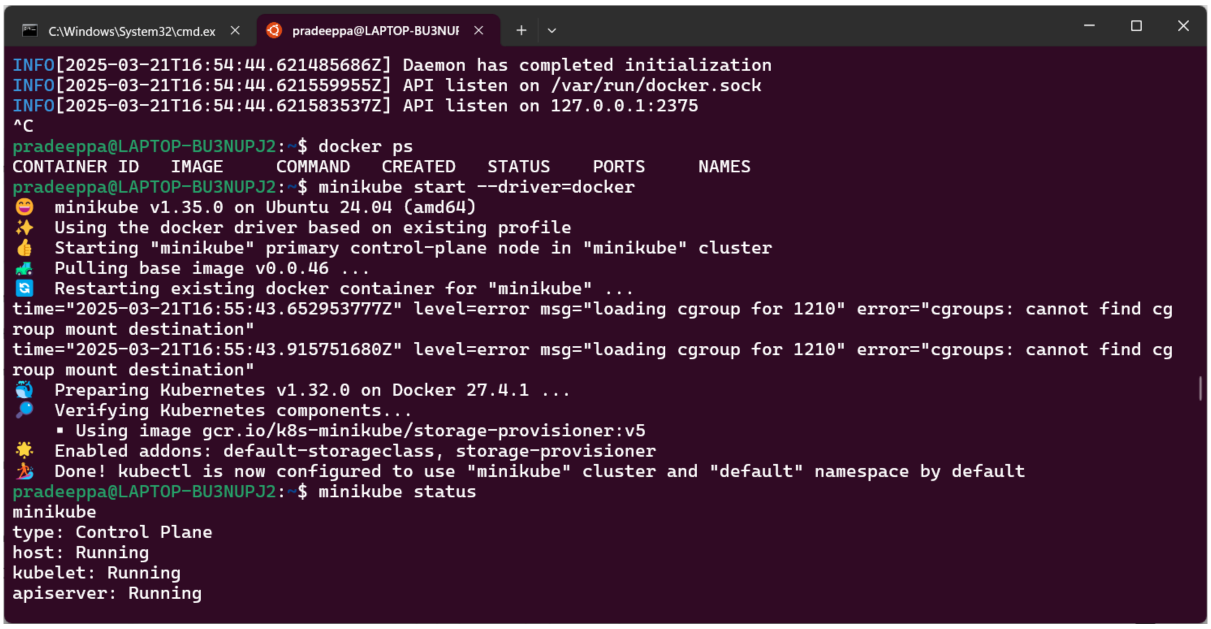
git branch -M main

git push -u origin main

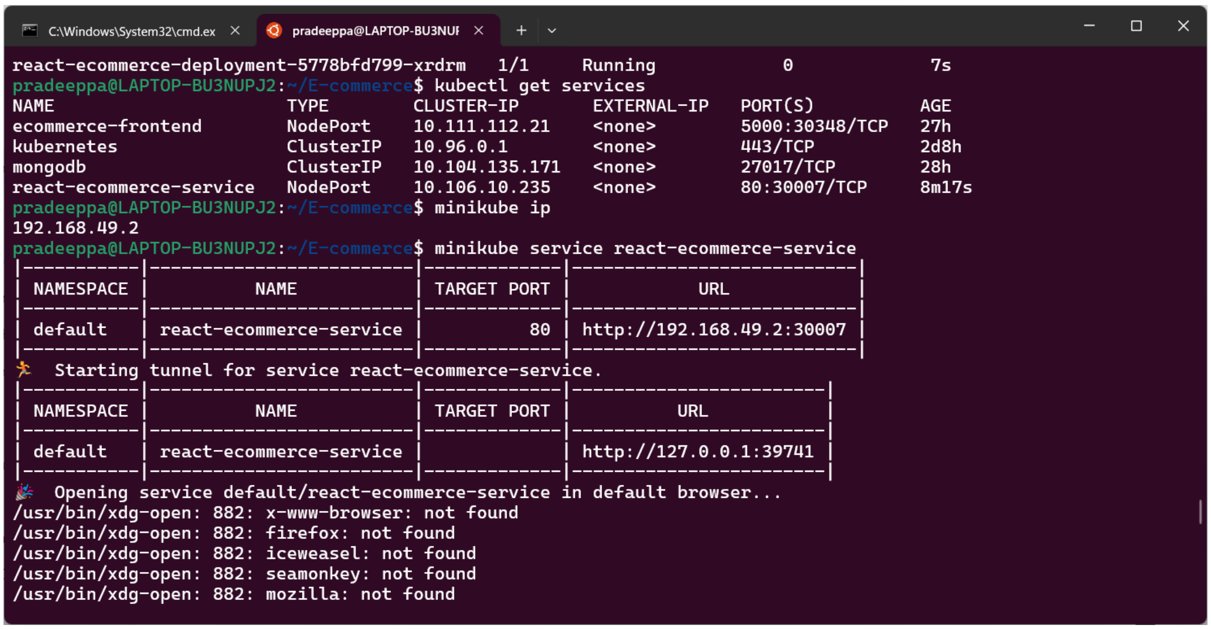
OUTPUT:



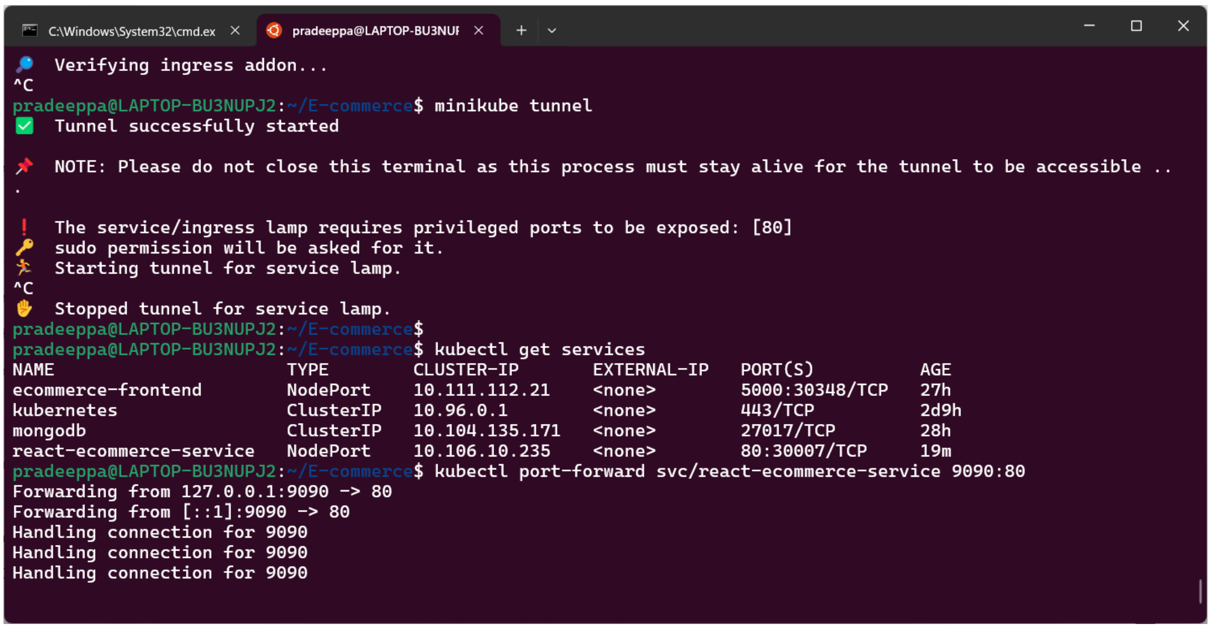
```
drwxr-xr-x 3 pradeeppa pradeeppa 4096 Mar 21 16:58 build
-rw-r--r-- 1 pradeeppa pradeeppa   35 Mar 21 16:58 build.sh
-rw-r--r-- 1 pradeeppa pradeeppa  187 Mar 21 16:58 deploy.sh
-rw-r--r-- 1 pradeeppa pradeeppa  476 Mar 21 16:58 deployment.yml
-rw-r--r-- 1 pradeeppa pradeeppa   86 Mar 21 16:58 docker-compose.yml
pradeeppa@LAPTOP-BU3NUPJ2:~/E-commerce$ docker build -t e-commerce-app .
[+] Building 36.0s (7/7) FINISHED                                                    docker:default
 => [internal] load build definition from Dockerfile                                          0.1s
 => => transferring dockerfile: 276B                                                          0.0s
 => [internal] load metadata for docker.io/library/nginx:latest                               3.2s
 => [internal] load .dockerignore                                                             0.0s
 => => transferring context: 2B                                                               0.0s
 => [1/2] FROM docker.io/library/nginx:latest@sha256:124b44bfc9ccd1f3cedf4b592d4d1e8bddb78b51ec2ed5056  32.2s
 => => resolve docker.io/library/nginx:latest@sha256:124b44bfc9ccd1f3cedf4b592d4d1e8bddb78b51ec2ed5056c  0.0s
 => => sha256:124b44bfc9ccd1f3cedf4b592d4d1e8bddb78b51ec2ed5056c52d3692baebc19 10.27kB / 10.27kB   0.0s
 => => sha256:54809b2f36d0ff38e8e5362b0239779e4b75c2f19ad70ef047ed050f01506bb4 2.29kB / 2.29kB     0.0s
 => => sha256:53a18edff8091d5faff1e42b4d885bc5f0f897873b0b8f0ace236cd5930819b0 8.58kB / 8.58kB     0.0s
 => => sha256:6e909acdb790c5a1989d9cfc795fda5a246ad6664bb27b5c688e2b734b2c5fad 28.20MB / 28.20MB  25.3s
 => => sha256:5eaa34f5b9c2a13ef2217ceb966953dfd5c3a21a990767da307be1f57e5a1e4f 43.95MB / 43.95MB  13.7s
 => => sha256:417c4bccf5349be7cd4ba91b1a2077ecf0ab50b3831bb071ba31f2c8bac02ed1 627B / 627B         1.3s
 => => sha256:e7e0ca015e553ccff5686ec2153c895313675686d3f6940144ce935c07554d85 955B / 955B         2.1s
 => => sha256:373fe654e9845b69587105e1b82833209521db456bdc5bc26ac7260e3eb2dd52 405B / 405B         2.7s
 => => sha256:97f5c0f51d43d499970597eef919f9170954289eff0c5d7b8f8afd73dbb57977 1.21kB / 1.21kB     3.2s
 => => sha256:c22eb46e871ad1cda19691450312c6b5c25eb5e6836773821d8091cffb6327cc 1.40kB / 1.40kB     3.7s
 => => extracting sha256:6e909acdb790c5a1989d9cfc795fda5a246ad6664bb27b5c688e2b734b2c5fad            3.4s
 => => extracting sha256:5eaa34f5b9c2a13ef2217ceb966953dfd5c3a21a990767da307be1f57e5a1e4f            2.6s
 => => extracting sha256:417c4bccf5349be7cd4ba91b1a2077ecf0ab50b3831bb071ba31f2c8bac02ed1            0.0s
```



```
INFO[2025-03-21T16:54:44.621485686Z] Daemon has completed initialization
INFO[2025-03-21T16:54:44.621559955Z] API listen on /var/run/docker.sock
INFO[2025-03-21T16:54:44.621583537Z] API listen on 127.0.0.1:2375
^C
pradeeppa@LAPTOP-BU3NUPJ2:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
pradeeppa@LAPTOP-BU3NUPJ2:~$ minikube start --driver=docker
😄  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" ...
time="2025-03-21T16:55:43.652953777Z" level=error msg="loading cgroup for 1210" error="cgroups: cannot find cg
roup mount destination"
time="2025-03-21T16:55:43.915751680Z" level=error msg="loading cgroup for 1210" error="cgroups: cannot find cg
roup mount destination"
🐳  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
pradeeppa@LAPTOP-BU3NUPJ2:~$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
```



```
react-ecommerce-deployment-5778bfd799-xrdrm   1/1     Running            0              7s
pradeeppa@LAPTOP-BU3NUPJ2:~/E-commerce$ kubectl get services
NAME                       TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
ecommerce-frontend         NodePort    10.111.112.21    <none>        5000:30348/TCP   27h
kubernetes                 ClusterIP   10.96.0.1        <none>        443/TCP          2d8h
mongodb                    ClusterIP   10.104.135.171   <none>        27017/TCP        28h
react-ecommerce-service    NodePort    10.106.10.235    <none>        80:30007/TCP     8m17s
pradeeppa@LAPTOP-BU3NUPJ2:~/E-commerce$ minikube ip
192.168.49.2
pradeeppa@LAPTOP-BU3NUPJ2:~/E-commerce$ minikube service react-ecommerce-service
|-----------|-------------------------|-------------|---------------------------|
| NAMESPACE |          NAME           | TARGET PORT |            URL            |
|-----------|-------------------------|-------------|---------------------------|
| default   | react-ecommerce-service |          80 | http://192.168.49.2:30007 |
|-----------|-------------------------|-------------|---------------------------|
🏃  Starting tunnel for service react-ecommerce-service.
|-----------|-------------------------|-------------|------------------------|
| NAMESPACE |          NAME           | TARGET PORT |          URL           |
|-----------|-------------------------|-------------|------------------------|
| default   | react-ecommerce-service |             | http://127.0.0.1:39741 |
|-----------|-------------------------|-------------|------------------------|
🎉  Opening service default/react-ecommerce-service in default browser...
/usr/bin/xdg-open: 882: x-www-browser: not found
/usr/bin/xdg-open: 882: firefox: not found
/usr/bin/xdg-open: 882: iceweasel: not found
/usr/bin/xdg-open: 882: seamonkey: not found
/usr/bin/xdg-open: 882: mozilla: not found
```

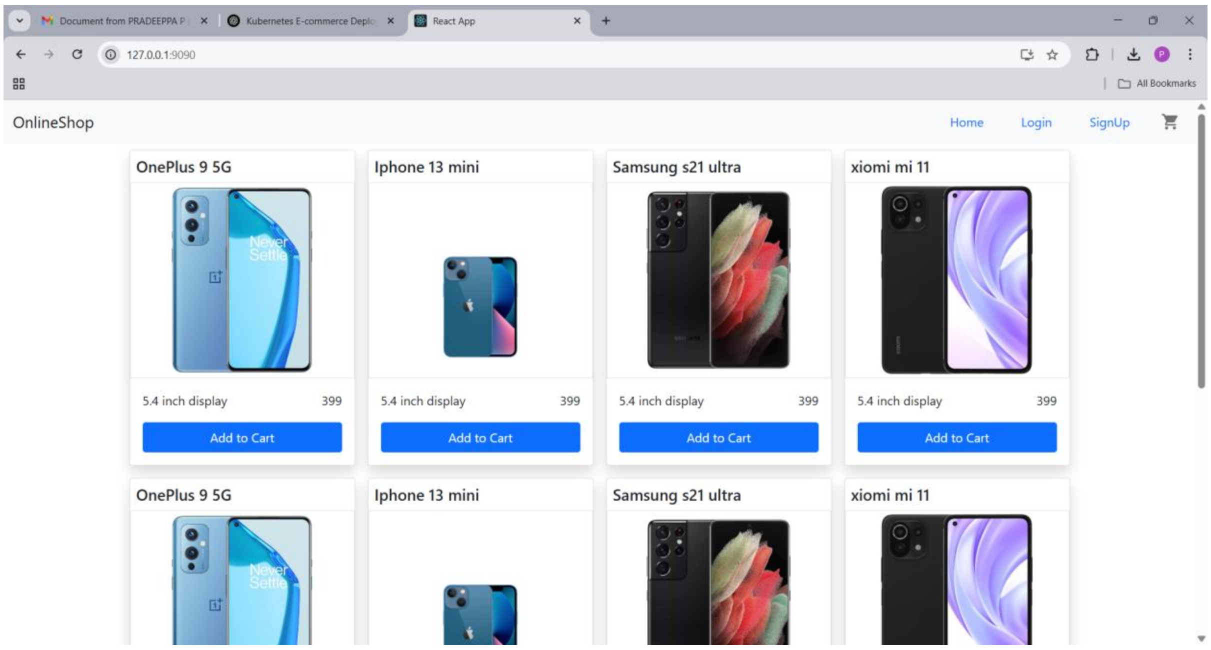


```
C:\Windows\System32\cmd.ex    pradeeppa@LAPTOP-BU3NUI

🔎  Verifying ingress addon...
^C
pradeeppa@LAPTOP-BU3NUPJ2:~/E-commerce$ minikube tunnel
✅  Tunnel successfully started

📌  NOTE: Please do not close this terminal as this process must stay alive for the tunnel to be accessible ..
.

❗  The service/ingress lamp requires privileged ports to be exposed: [80]
🔑  sudo permission will be asked for it.
🏃  Starting tunnel for service lamp.
^C
✋  Stopped tunnel for service lamp.
pradeeppa@LAPTOP-BU3NUPJ2:~/E-commerce$
pradeeppa@LAPTOP-BU3NUPJ2:~/E-commerce$ kubectl get services
NAME                      TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
ecommerce-frontend        NodePort    10.111.112.21    <none>        5000:30348/TCP   27h
kubernetes                ClusterIP   10.96.0.1        <none>        443/TCP          2d9h
mongodb                   ClusterIP   10.104.135.171   <none>        27017/TCP        28h
react-ecommerce-service   NodePort    10.106.10.235    <none>        80:30007/TCP     19m
pradeeppa@LAPTOP-BU3NUPJ2:~/E-commerce$ kubectl port-forward svc/react-ecommerce-service 9090:80
Forwarding from 127.0.0.1:9090 -> 80
Forwarding from [::1]:9090 -> 80
Handling connection for 9090
Handling connection for 9090
Handling connection for 9090
```


Document from PRADEEPPA P
Kubernetes E-commerce Deplo
React App
127.0.0.1:9090
All Bookmarks
OnlineShop
Home
Login
SignUp
OnePlus 9 5G
Iphone 13 mini
Samsung s21 ultra
xiomi mi 11
5.4 inch display
399
Add to Cart
5.4 inch display
399
Add to Cart
5.4 inch display
399
Add to Cart
5.4 inch display
399
Add to Cart
OnePlus 9 5G
Iphone 13 mini
Samsung s21 ultra
xiomi mi 11