

TASK – 3 MINIKUBE DEPLOYMENT TASK

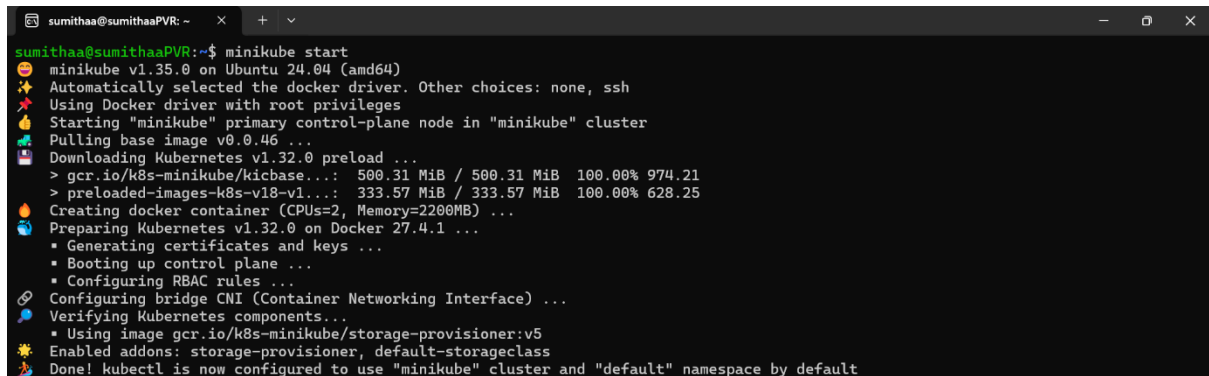
NAME:SUMITHAA P V R

ROLL NO:22CSR212

STEP 1: Start Minikube

Start the Minikube cluster using the following command:

minikube start



```
sumithaa@sumithaaPVR:~$ minikube start
minikube v1.35.0 on Ubuntu 24.04 (amd64)
Automatically selected the docker driver. Other choices: none, ssh
Using Docker driver with root privileges
Starting "minikube" primary control-plane node in "minikube" cluster
Pulling base image v0.0.46 ...
Downloading Kubernetes v1.32.0 preload ...
> gcr.io/k8s-minikube/kicbase...: 500.31 MiB / 500.31 MiB 100.00% 974.21
> preloaded-images-k8s-v18-v1...: 333.57 MiB / 333.57 MiB 100.00% 628.25
Creating docker container (CPUs=2, Memory=2200MB) ...
Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
  * Generating certificates and keys ...
  * Booting up control plane ...
  * Configuring RBAC rules ...
Configuring bridge CNI (Container Networking Interface) ...
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
```

This initializes the Minikube cluster using Docker as the driver.

STEP 2: Install Kubectl

Since kubectl is not found,install it with the following command:

sudo snap install kubectl --classic

Alternatively, you can download it using curl:

```
curl -LO "https://dl.k8s.io/release/$(curl -L -s
https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl" sudo install -o root -g root -m
0755 kubectl /usr/local/bin/kubectl
```

STEP 3: Verify kubectl Installation

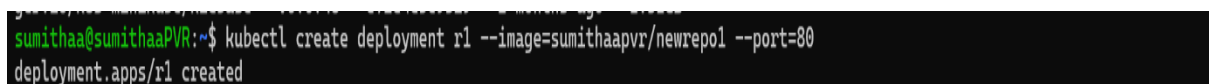
Check the client version to confirm successful installation:

Kubectl version -client

STEP 4: Create a Deployment

Create a deployment named `r1` with the image `sumithaapvr/newrepo1`:

kubectl create deployment r1 --image=sumithaapvr/newrepo1 --port=80



```
sumithaa@sumithaaPVR:~$ kubectl create deployment r1 --image=sumithaapvr/newrepo1 --port=80
deployment.apps/r1 created
```

STEP 5: Expose the Deployment

Expose the deployment as a NodePort service:

kubectl expose deployment r1 --port=80 --type=NodePort

```
sumithaa@sumithaaPVR:~$ kubectl expose deployment.apps/r1 --port=80 --type=NodePort
service/r1 exposed
```

STEP 6: Verify the Pod

Check the running pods:

```
kubectl get pods
```

Step 7: Access the Service

Expose the service using Minikube and get the URL:

```
minikube service r1
```

```
sumithaa@sumithaaPVR:~$ minikube service r1
```

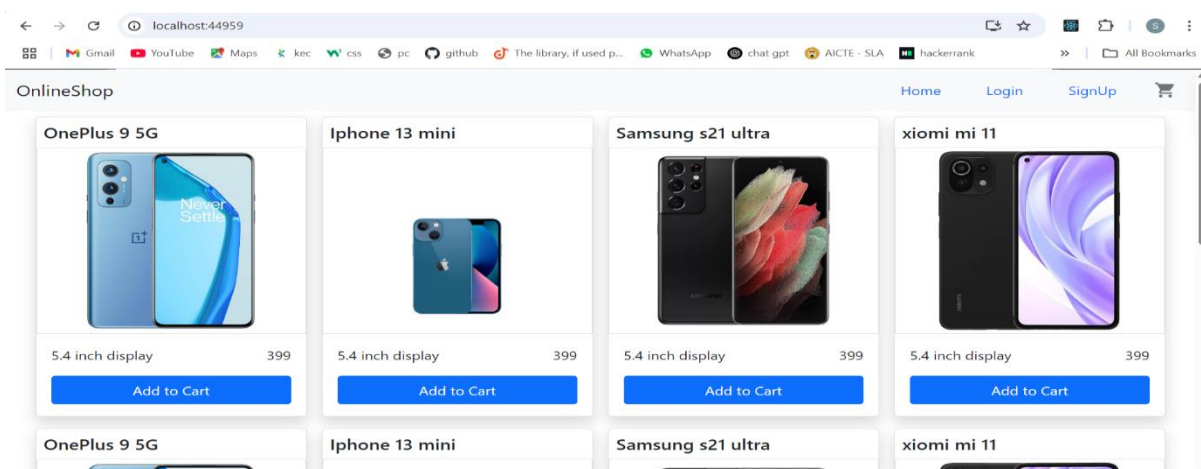
NAMESPACE	NAME	TARGET PORT	URL
default	r1	80	http://192.168.49.2:31394

🔗 Starting tunnel for service r1.

NAMESPACE	NAME	TARGET PORT	URL
default	r1		http://127.0.0.1:44959

🌐 Opening service default/r1 in default browser...
 http://127.0.0.1:44959
 ⚠️ Because you are using a Docker driver on linux, the terminal needs to be open to run it.

STEP 8: Output in the Web Browser



DockerHub:

