

### Task 3

Create a Kubernetes deployment in Minikube using kubectl, expose it as a service, and run Docker image to serve an HTML website.

#### 1. Installing and starting the Minikube.

```
bala2005@Toxin:~$ minikube start
minikube v1.35.0 on Ubuntu 24.04 (amd64)
✳ Automatically selected the docker driver
✳ 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 ...
> preloaded-images-k8s-v18-v1...: 333.57 MiB / 333.57 MiB 100.00% 4.72 Mi
> gcr.io/k8s-minikube/kicbase...: 500.31 MiB / 500.31 MiB 100.00% 4.61 Mi
🔥 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

! /usr/local/bin/kubectl is version 1.30.5, 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
```

#### 2. Creating a Deployment d1 and exposing it.

```
bala2005@Toxin:~$ kubectl create deployment d1 --image=balachandran2005/task2 --port=80
deployment.apps/d1 created
bala2005@Toxin:~$ kubectl expose deployment d1 --type NodePort --port=80
service/d1 exposed
```

#### 3. Verifying whether the deployment is created.

```
bala2005@Toxin:~$ kubectl get svc
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
d1	NodePort	10.97.85.60	<none>	80:30722/TCP	16s
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	2m42s

#### 4. Starting the tunnel and running it.

```
bala2005@Toxin:~$ minikube service d1
```

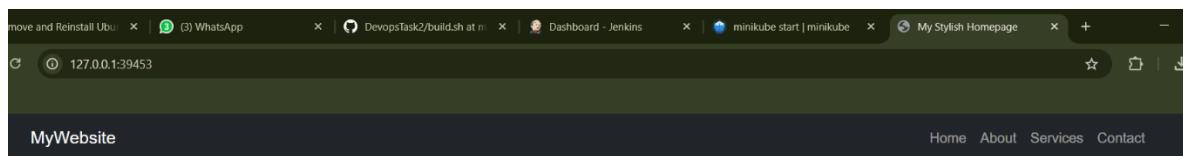
NAMESPACE	NAME	TARGET PORT	URL
default	d1	80	http://192.168.49.2:30722

Starting tunnel for service d1.

NAMESPACE	NAME	TARGET PORT	URL
default	d1		http://127.0.0.1:39453

Opening service default/d1 in default browser...  
http://127.0.0.1:39453  
! Because you are using a Docker driver on linux, the terminal needs to be open to run it.

#### 5. Final output of the URL <http://127.0.0.1:39453>.



Get Started