# ASSIGNMENT 3

# CREATING A MINIKUBE SERVICE TO HANDLE CONTAINERS

## Minikube:

- Minikube is a lightweight Kubernetes tool that allows developers to run a Kubernetes cluster locally.
- It helps in managing and testing containerized applications without needing a full cloud-based cluster.
- A Minikube Service is used to expose applications running in containers to external networks, making them accessible for development and testing.

## Steps to Create a Minikube Service

1. **Start Minikube**
    - Ensure that Minikube is installed.
    - Start the Kubernetes cluster.
2. **Create a Deployment**
    - Define a deployment with container image, replicas, and pod specifications.
3. **Expose the Deployment as a Service**
    - Create a Kubernetes service.
    - Choose a service type (**ClusterIP, NodePort, LoadBalancer**).
4. **Verify the Service**
    - Check if the service is running.
    - Obtain the external URL or port.
5. **Access the Application**
    - Use the Minikube service URL.
    - Access the application via a browser or API.

1. Minikube start and creating a deployment:

```
roshni@Abhinavkrishna:~$ 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 ...
❗  minikube cannot pull kicbase image from any docker registry, and is trying to download kicbase tarball from github r
elease page via HTTP.
❗  It's very likely that you have an internet issue. Please ensure that you can access the internet at least via HTTP,
directly or with proxy. Currently your proxy configure is:

💾  Downloading Kubernetes v1.32.0 preload ...
    > preloaded-images-k8s-v18-v1...:  333.57 MiB / 333.57 MiB  100.00% 2.21 Mi
    > kicbase-v0.0.46-amd64.tar:  1.23 GiB / 1.23 GiB  100.00% 2.24 MiB p/s 9m2
🤷  docker "minikube" container is missing, will recreate.
🔥  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 1.30.2, 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
roshni@Abhinavkrishna:~$ kubectl create deployment task3 --image=roshni178/task --port=80
deployment.apps/task3 created
```

2. Exposing the deployment:

```
roshni@Abhinavkrishna:~$ kubectl expose deployment task3 --type=NodePort --port=80
service/task3 exposed
```

3. Checking for pods:

```
roshni@Abhinavkrishna:~$ kubectl get pods
NAME                     READY   STATUS    RESTARTS   AGE
task3-7967f9cbf8-hpl5w   1/1     Running   0          90s
roshni@Abhinavkrishna:~$
```

4. Checking for deployment:

```
roshni@Abhinavkrishna:~$ kubectl get deployments
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
task3   1/1     1            1           2m52s
roshni@Abhinavkrishna:~$ kubectl get svc
NAME         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP        4m1s
task3        NodePort    10.111.40.248   <none>        80:31210/TCP   39s
```

5. Minikube services:

```
roshni@Abhinavkrishna:~$ minikube service task3
|-----------|-------|-------------|--------------------------|
| NAMESPACE | NAME  | TARGET PORT |           URL            |
|-----------|-------|-------------|--------------------------|
| default   | task3 |          80 | http://192.168.49.2:31210 |
|-----------|-------|-------------|--------------------------|
🏃  Starting tunnel for service task3.
|-----------|-------|-------------|------------------------|
| NAMESPACE | NAME  | TARGET PORT |          URL           |
|-----------|-------|-------------|------------------------|
| default   | task3 |             | http://127.0.0.1:35225 |
|-----------|-------|-------------|------------------------|
🏃  Opening service default/task3 in default browser...
👉  http://127.0.0.1:35225
❗  Because you are using a Docker driver on linux, the terminal needs to be open to run it.
```

6. Output