TASK 3

KUBERNETES

Step 1: Install minikube for linux from its official website.

Use commands like,

curl -LO https://github.com/kubernetes/minikube/releases/latest/download/minikube-linux-amd64

sudo install minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-linux-amd64

to install minikube in Ubuntu.

```
PS C:\WINDOWS\system32> wsl
sree_ubuntu@Sree:/mnt/c/WINDOWS/system32$ cd
sree_ubuntu@Sree:~$ curl -LO https://github.com/kubernetes/minikube/releases/latest/download/minikube-linux-amd64
stall minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-linux-amd64
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:--  0:00:01 --:--:--     0
  0     0    0     0    0     0      0      0 --:--:--  0:00:02 --:--:--     0
100  119M  100  119M    0     0  4059k      0  0:00:30  0:00:30 --:--:-- 9345k
sree_ubuntu@Sree:~$ sudo install minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-linux-amd64
[sudo] password for sree_ubuntu:
sree_ubuntu@Sree:~$ minikube start
😄  minikube v1.35.0 on Ubuntu 22.04 (amd64)
✨  Automatically selected the docker driver. Other choices: none, ssh
📌  Using Docker driver with root privileges
❗  For an improved experience it's recommended to use Docker Engine instead of Docker Desktop.
Docker Engine installation instructions: https://docs.docker.com/engine/install/#server
👍  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% 7.87 Mi
    > preloaded-images-k8s-v18-v1...:  333.57 MiB / 333.57 MiB  100.00% 4.94 Mi
🔥  Creating docker container (CPUs=2, Memory=2200MB) ...
❗  Failing to connect to https://registry.k8s.io/ from inside the minikube container
💡  To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
🐳  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: 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
```

Step 2: Start minikube and deploy docker hub image in pod.

```
sree_ubuntu@Sree:~$ kubectl version --client
Client Version: v1.30.2
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
sree_ubuntu@Sree:~$ kubectl create deployment kube-first --image=sreevadhani/ar --port 80
deployment.apps/kube-first created
sree_ubuntu@Sree:~$ kubectl get pods
NAME                          READY   STATUS    RESTARTS   AGE
kube-first-5b5fc7f47c-7wwp2   1/1     Running   0          53s
```

Step 3: Expose the pod to the slave node

```
sree_ubuntu@Sree:~$ kubectl expose deployment kube-first --type=NodePort --port=80
service/kube-first exposed
sree_ubuntu@Sree:~$ kubectl get svc
NAME         TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
kube-first   NodePort    10.105.110.224   <none>        80:30953/TCP   22s
kubernetes   ClusterIP   10.96.0.1        <none>        443/TCP        35m
```

Step 4: Tunneling the service

```
sree_ubuntu@Sree:~$ minikube service kube-first
|-----------|------------|-------------|---------------------------|
| NAMESPACE |    NAME    | TARGET PORT |            URL            |
|-----------|------------|-------------|---------------------------|
| default   | kube-first |          80 | http://192.168.49.2:30953 |
|-----------|------------|-------------|---------------------------|
🏃  Starting tunnel for service kube-first.
|-----------|------------|-------------|------------------------|
| NAMESPACE |    NAME    | TARGET PORT |          URL           |
|-----------|------------|-------------|------------------------|
| default   | kube-first |             | http://127.0.0.1:37609 |
|-----------|------------|-------------|------------------------|
🎉  Opening service default/kube-first in default browser...
👉  http://127.0.0.1:37609
❗  Because you are using a Docker driver on linux, the terminal needs to be open to run it.
```

Step 5: Check whether it is running