Task01-K8s

OverView:-

1. Create a deployment using a hello world image
2. Create a service for it
3. Create an ingress that routes to this service with a domain like this sub_domain.<your-name>.com

Steps:-

Step 1: Prepare the working space

1.1. Create a Directory called Zerosploit

```
mkdir ~/zerosploit
```

1.2. Create all files needed for the application

```
vim deployment.yaml
```

##Create a deployment.yaml file to specify the pods needed

```
vim ingress.yaml
```

##Create an ingress.yaml file to specify the ingress service

```
vim internal-svc.yaml
```

##Create an internal-svc.yaml file to specify the internal service between pod (ClusterIP)

Step 2: Write the deployment.yaml file

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-world-deployment
  labels:
     app: helloworld
spec:
  replicas: 3
  selector:
     matchLabels:
     app: helloworld
  template:
     metadata:
     labels:
     app: helloworld
     spec:
     containers:
     - name: hello-pod
     image: crccheck/hello-world
     ports:
          - containerPort: 8000
```

This will create 3 replicas for the Hello World pod and expose it on port 8000

Step 3: Write the internal-svc.yaml file

```
apiVersion: v1

kind: Service

metadata:

  name: internal-service

spec:

  selector:

      app: helloworld

  ports:

      - protocol: TCP

      port: 80

      targetPort: 8000
```

This will create a ClusterIP service between the (hello-world-deployment) deployment

ClusterIP will load balance between the 3 replicas and act as an endpoint for all three

Step 4: Write the ingress.yaml file

```
apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

  name: my-ingress

  annotations:

      nginx.ingress.kubernetes.io/rewrite-target: /
```

```
  namespace: mohamedhassan
spec:
  ingressClassName: nginx
  rules:
  - host: subdomain.mohamedhassan.com
     http:
     paths:
     - path: /
           pathType: Prefix
           backend:
           service:
           name: internal-service
           port:
                number: 80
```

This will create a ingres service with a domain called (subdomain.mohamedhassan.com)

ingress uses the internal service (ClusterIP) to load balance between all three replicas

ingress external IP is assigned by a loadbalancer which can be seen in the following:

```
mohamedhassan@master:~/task-03$ kubectl get ingress
NAME         CLASS   HOSTS                          ADDRESS          PORTS   AGE
my-ingress   nginx   subdomain.mohamedhassan.com   198.96.95.206   80      2d23h
```

Step 5: Start Deploying the architecture

5.1. Run the deployment.yaml file

```
Kubectl apply -f deployment.yaml
```

```
Kubectl get deployments
```

```
mohamedhassan@master:~/task-03$ kubectl apply -f deployment.yaml
deployment.apps/hello-world-deployment created
mohamedhassan@master:~/task-03$ kubectl get deployments.apps
NAME                    READY   UP-TO-DATE   AVAILABLE   AGE
hello-world-deployment   3/3     3            3           9s
```

5.2. Run the internal-svc.yaml file

```
kubectl apply -f internal-svc.yaml
```

```
kubectl get service
```

```
mohamedhassan@master:~/task-03$ kubectl apply -f internal-svc.yaml
service/internal-service created
mohamedhassan@master:~/task-03$ kubectl get service
NAME               TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
internal-service   ClusterIP   10.111.13.254   <none>        80/TCP    5s
```

5.3. Run the ingress.yaml file

```
kubectl apply -f ingress.yaml
```

```
kubectl get ingress
```

```
mohamedhassan@master:~/task-03$ kubectl apply -f ingress.yaml
ingress.networking.k8s.io/my-ingress created
mohamedhassan@master:~/task-03$ kubectl get ingress
NAME         CLASS   HOSTS                          ADDRESS   PORTS   AGE
my-ingress   nginx   subdomain.mohamedhassan.com              80      3s
mohamedhassan@master:~/task-03$ kubectl get ingress
NAME         CLASS   HOSTS                          ADDRESS   PORTS   AGE
my-ingress   nginx   subdomain.mohamedhassan.com              80      23s
mohamedhassan@master:~/task-03$ kubectl get ingress
NAME         CLASS   HOSTS                          ADDRESS         PORTS   AGE
my-ingress   nginx   subdomain.mohamedhassan.com   198.96.95.206   80      65s
```

It might take some time for the ingress service to get the external IP

Step 6: Add the Domain name to your local hosts

6.1. Edit the hosts file on your local system

```
sudo Vim /etc/hosts
```

6.2. Add the new domain to the hosts

```
198.96.95.206  subdomain.mohamedhassan.com
```

```
127.0.0.1 localhost
127.0.1.1 mohamed-Inspiron-3581

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
198.96.95.206 subdomain.mohamedhassan.com
~
```

Step 7: Test to see if the architecture is working

7.1. Go to your local browser and search for (subdomain.mohamedhassan.com)