Task 4

Deploying a sample application using Kubernetes

Introduction

This guide outlines the steps to deploy a sample application using Kubernetes. It covers setting up a Deployment and a Service configuration, applying them with kubectl, and accessing the deployed application through Minikube. Additionally, it includes best practices for monitoring, scaling, and troubleshooting Kubernetes deployments.

- Set Up Kubernetes Environment – Install and start Minikube and kubectl.
- Create a Deployment – Define a YAML file to manage application pods.
- Expose the Application – Configure a Service to enable network access.
- Apply Configurations – Use kubectl apply to deploy resources.
- Verify Deployment – Check pod status, logs, and resource utilization.
- Access the Application – Retrieve the Minikube service URL for interaction.
- Scale the Deployment – Adjust replicas for better availability and load balancing.
- Monitor and Troubleshoot – Use kubectl logs, kubectl describe, and dashboards to resolve issues.

Deployment text file:

Code in day4.txt

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: test
  name: test
spec:
  replicas: 1
  selector:
    matchLabels:
      app: test
  template:
    metadata:
      labels:
        app: test
    spec:
      containers:
      - name: test
        image: archita392/task_3
        imagePullPolicy: Always
        ports:
        - containerPort: 80
          name: http
          protocol: TCP
~
~
~
~
~
"day4.txt" 24L, 407B
```

Exposed text file:

```
archita@DESKTOP-U3693UJ:~$ vim expose.txt
```

Code in expose.txt

```
apiVersion: v1
kind: Service
metadata:
  name: test  # Corrected: added name field
  labels:
    app: test
spec:
  selector:
    app: test  # Ensure this matches the Deployment's label
  ports:
    - name: http
      protocol: TCP
      port: 80
      targetPort: 80
  type: NodePort
~
~
~
~
~
~
~
~
~
~
~
~
~
"expose.txt" 16L, 285B
```

```
archita@DESKTOP-U3693UJ:~$ kubectl apply -f day4.txt
deployment.apps/test created
```

```
archita@DESKTOP-U3693UJ:~$ kubectl apply -f expose.txt
service/test created
```

```
archita@DESKTOP-U3693UJ:~$ minikube service test
|-----------|------|-------------|-------------------------|
| NAMESPACE | NAME | TARGET PORT |           URL           |
|-----------|------|-------------|-------------------------|
| default   | test | http/80     | http://192.168.49.2:30580 |
|-----------|------|-------------|-------------------------|
🏃  Starting tunnel for service test.
|-----------|------|-------------|------------------------|
| NAMESPACE | NAME | TARGET PORT |          URL           |
|-----------|------|-------------|------------------------|
| default   | test |             | http://127.0.0.1:46341 |
|-----------|------|-------------|------------------------|
🎉  Opening service default/test in default browser...
👉  http://127.0.0.1:46341
❗  Because you are using a Docker driver on linux, the terminal needs to be open to run it.
^C✋  Stopping tunnel for service test.
```

Final output