Date: 07.02.2025

# **Kubernetes Deployment Guide**

#### Introduction

This document provides a guide on deploying a sample application using Kubernetes. The process includes creating a Deployment and a Service configuration, applying them using kubectl, and accessing the deployed service through Minikube.

#### **UNIX/Linux Commands**

- cat: Short for "concatenate," used to display the content of files.
- **vim**: Opens the Vim text editor, a highly configurable and powerful text editor commonly used in UNIX/Linux environments.

```
ubuntu@LAPTOP-DEQKQVPU:~$ 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 ...

Updating the running docker "minikube" container ...

Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...

Verifying Kubernetes components...

Using image gcr.io/k8s-minikube/storage-provisioner:v5

Enabled addons: storage-provisioner, default-storageclass

Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

## **Kubernetes Commands**

### **Apply Deployment Configuration**

kubectl apply -f t1.txt

```
ubuntu@LAPTOP-DEOKOVPU:~$ cat file1.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: rrragul/sample
        imagePullPolicy: Always
        ports:
        - containerPort: 80
          name: http
          protocol: TCP
   tu@LAPTOP-DEQKQVPU:~$ kubectl apply -f file1.txt
```

This command applies the Deployment configuration and creates the deployment named test.

#### **Apply Service Configuration**

deployment.apps/test unchanged

kubectl apply -f t2.txt

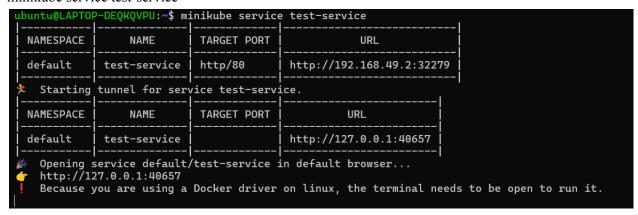
```
ubuntu@LAPTOP-DEQKQVPU:~$ cat file2.txt
apiVersion: v1
kind: Service
metadata:
 name: test-service # Corrected the name field
 labels:
   app: test
spec:
  selector:
   app: test # Ensures it matches the label in the corresponding deployment/pod
 ports:
    - name: http
     port: 80
     protocol: TCP
      targetPort: 80
 type: NodePort # Exposes service on a node port
ubuntu@Harz-PC:~$ kubectl apply -f t2.txt
```

This command applies the Service configuration and creates the service named test-service.

#### **Access the Service**

service/test-service created

#### minikube service test-service



This command starts a tunnel for the service and provides a URL to access the deployed application.

## **Output**

