### **Lab Exercise 9- ClusterIP in Service in Kubernetes**

A ClusterIP is the default type of service in Kubernetes. It provides a stable internal IP address for the service, which is accessible only within the cluster. This is useful for internal communication between services within the cluster.

**Step 1: Set Up Kubernetes Cluster**

* Ensure you have access to a Kubernetes cluster. You can use a local setup with Minikube, kind, or use a cloud-based Kubernetes service.

**Step 2: Deploy a Sample Application**

* Create a deployment for a sample application. Here, we'll use Nginx as our sample application.
* Create a file named **nginx-deployment.yaml** with the following content:

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-deployment

spec:

replicas: 3

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

- name: nginx

image: nginx:latest

ports:

- containerPort: 80

Apply this deployment using:

kubectl apply -f nginx-deployment.yaml

**Step 3: Create a ClusterIP Service**

* Create a service manifest to expose the Nginx application internally using ClusterIP.
* Create a file named **nginx-service.yaml** with the following content:

apiVersion: v1

kind: Service

metadata:

name: nginx-service

spec:

selector:

app: nginx

ports:

- protocol: TCP

port: 80

targetPort: 80

type: ClusterIP

Apply this service using:

kubectl apply -f nginx-service.yaml

**Step 4: Verify the Service Creation**

Check the service to ensure it has been created and has a ClusterIP assigned:

kubectl get services

You should see an output similar to:

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

nginx-service ClusterIP <some-ip> <none> 80/TCP <age>

**Step 5: Access the Service Internally**

To verify that the service is working internally, you can use a temporary Pod to test access to the nginx-service.

curl service-ip:port

You should see the Nginx welcome page content.

**Step 6: Clean Up**

After completing the exercise, clean up the resources created:

kubectl delete deployment nginx-deployment

kubectl delete service nginx-service