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: Create Deployment (Correct Labels)

Create file:

nginx-deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-app
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx   # MUST match template labels
  template:
    metadata:
      labels:
        app: nginx  # MUST match service selector
    spec:
      containers:
      - name: nginx
        image: nginx
        ports:
        - containerPort: 80
```

Save file.

```
kubectl apply -f nginx-deployment.yaml
```

Verify:

```
kubectl get pods --show-labels
```

You should see:

```
app=nginx
```

Step 2: Create ClusterIP Service (Matching Selector)

Create file:

nginx-service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  type: ClusterIP
  selector:
    app: nginx   # MUST match pod label
  ports:
  - port: 80
    targetPort: 80
```

Apply:

```
kubectl apply -f nginx-service.yaml
```

Step 3: Verify It Is Working

Check Service

```
kubectl get svc
```

You should see:

```
TYPE       CLUSTER-IP
ClusterIP   10.x.x.x
```

MOST IMPORTANT CHECK

```
kubectl get endpoints nginx-service
```

Now you should see something like:

```
10.244.x.x:80, 10.244.x.x:80
```

Step 4: Test From Inside Cluster

Run test pod:

```
kubectl run test-pod –image=busybox –rm -it – sh
```

Inside pod:

```
wget -qO- http://nginx-service
```

You should see:

```
Welcome to nginx!
```

Exit:

```
exit
```

Step 5: Test From Outside (It Will Fail)

From your local machine:

```
curl http://<CLUSTER-IP>
```

It will NOT work.

That proves:

ClusterIP works only inside cluster.