Using K8s Services

Relevant Documentation

- Service

Lesson Reference

Create a deployment.

```
vi deployment-svc-example.yml
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: deployment-svc-example
spec:
  replicas: 3
  selector:
    matchLabels:
      app: svc-example
  template:
    metadata:
      labels:
        app: svc-example
    spec:
      containers:
      - name: nginx
        image: nginx:1.19.1
        ports:
        - containerPort: 80
```

```
kubectl create -f deployment-svc-example.yml
```

Create a ClusterIP Service to expose the deployment's Pods within the cluster network.

```
vi svc-clusterip.yml
```

```
apiVersion: v1
kind: Service
metadata:
  name: svc-clusterip
spec:
  type: ClusterIP
  selector:
    app: svc-example
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

```
kubectl create -f svc-clusterip.yml
```

Get a list of the Service's endpoints.

```
kubectl get endpoints svc-clusterip
```

Create a busybox Pod to test your service.

```
vi pod-svc-test.yml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-svc-test
spec:
  containers:
  - name: busybox
    image: radial/busyboxplus:curl
    command: ['sh', '-c', 'while true; do sleep 10; done']
```

```
kubectl create -f pod-svc-test.yml
```

Run a command within the busybox Pod to make a request to the service.

```
kubectl exec pod-svc-test -- curl svc-clusterip:80
```

You should see the Nginx welcome page, which is being served by one of the backend Pods created earlier using a Deployment.

Create a NodePort Service to expose the Pods externally.

```
vi svc-nodeport.yml
```

```
apiVersion: v1
kind: Service
metadata:
  name: svc-nodeport
spec:
  type: NodePort
  selector:
    app: svc-example
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
      nodePort: 30080
```

```
kubectl create -f svc-nodeport.yml
```

Test the service by making a request from your browser to `http://<Control Plane Node Public IP>:30080`. You should see the Nginx welcome page.