Exploring Services

Relevant Documentation

- Service

Exam Tips

- Services allow you to expose an application running in multiple Pods.
- ClusterIP Services expose the Pods to other applications within the cluster.
- NodePort Services expose the Pods externally using a port that listens on every node in the cluster.

Lesson Reference

Log in to the **control plane node**.

Create a server Pod.

```
vi service-server-pod.yml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: service-server-pod
  labels:
    app: service-server
spec:
  containers:
  - name: nginx
    image: nginx:stable
    ports:
    - containerPort: 80
```

```
kubectl apply -f service-server-pod.yml
```

Create a ClusterIP Service for the Pod.

```
vi clusterip-service.yml
```

```
apiVersion: v1
kind: Service
metadata:
  name: clusterip-service
spec:
  type: ClusterIP
  selector:
    app: service-server
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 80
```

```
kubectl apply -f clusterip-service.yml
```

Get the cluster IP address for the Service.

```
kubectl get svc clusterip-service
```

Use the cluster IP to test the Service.

```
curl <service cluster IP address>:8080
```

Now let's create a NodePort Service.

```
vi nodeport-service.yml
```

```
apiVersion: v1
kind: Service
metadata:
  name: nodeport-service
spec:
  type: NodePort
  selector:
    app: service-server
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 80
      nodePort: 30080
```

```
kubectl apply -f nodeport-service.yml
```

Test the NodePort Service. The external node port listens directly on the host, so we can access it using `localhost`.

```
curl localhost:30080
```

You can also use the public IP address of any of your Kubernetes nodes to access the service at `http://<Node Public IP address>:30080`.