쿠버네티스 실습 5 - Service

1. Service 란?

- Service 는 쿠버네티스에 배포한 애플리케이션(Pod)을 외부에서 접근하기 쉽게 추상화한 리소스입니다.
  - https://kubernetes.io/ko/docs/concepts/services-networking/service/
- Pod 은 IP 를 할당받고 생성되지만, 언제든지 죽었다가 다시 살아날 수 있으며, 그 과정에서 IP 는 항상 재할당받기에 고정된 IP 로 원하는 Pod 에 접근할 수는 없습니다.
- 따라서 클러스터 외부 혹은 내부에서 Pod 에 접근할 때는, Pod 의 IP 가 아닌 Service 를 통해서 접근하는 방식을 거칩니다.
- Service 는 고정된 IP 를 가지며, Service 는 하나 혹은 여러 개의 Pod 과 매칭됩니다.
- 따라서 클라이언트가 Service 의 주소로 접근하면, 실제로는 Service 에 매칭된 Pod 에 접속할 수 있게 됩니다.

2. Service 생성

- 우선 지난 시간에 생성한 Deployment 를 다시 생성합니다.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
```

```
        image: nginx:1.14.2
        ports:
        - containerPort: 80
```

- `kubectl apply -f deployment.yaml`

- 생성된 Pod 의 IP 를 확인하고 접속을 시도해봅니다.

```
kubectl get pod -o wide
# Pod 의 IP 를 확인합니다.

curl -X GET <POD-IP> -vvv
ping <POD-IP>
# 통신 불가능
```

- 할당된 <POD-IP> 는 클러스터 내부에서만 접근할 수 있는 IP 이기 때문에 외부에서는 Pod 에 접속할 수 없습니다.

- minikube 내부로 접속하면 통신이 되는지 확인해보겠습니다.

```
minikube ssh
# minikube 내부로 접속합니다.

curl -X GET <POD-IP> -vvv
ping <POD-IP>
# 통신 가능
```

- 그럼 이제 위의 Deployment 를 매칭시킨 Service 를 생성해보겠습니다.

```
apiVersion: v1
kind: Service
metadata:
  name: my-nginx
  labels:
    run: my-nginx
spec:
  type: NodePort # Service 의 Type 을 명시하는 부분입니다. 자세한 설명은 추후 말씀드리겠습니다.
  ports:
  - port: 80
    protocol: TCP
  selector: # 아래 label 을 가진 Pod 을 매핑하는 부분입니다.
    app: nginx
```

- Service 를 생성합니다.

```
vi service.yaml
# 파일을 열어 위의 내용을 복사 붙여넣기 합니다.

kubectl apply -f service.yaml

kubectl get service
# PORT 80:<PORT> 숫자 확인

curl -X GET $(minikube ip):<PORT>
# 클러스터 외부에서도 정상적으로 pod 에 접속할 수 있는 것을 확인합니다.
```

- *** Service 의 Type 이란?**
  - **NodePort** 라는 type 을 사용했기 때문에, minikube 라는 kubernetes cluster 내부에 배포된 서비스에 클러스터 외부에서 접근할 수 있었습니다.
    - 접근하는 IP 는 pod 이 떠있는 노드(머신)의 IP 를 사용하고, Port 는 할당받은 Port 를 사용합니다.
  - **LoadBalancer** 라는 type 을 사용해도, 마찬가지로 클러스터 외부에서 접근할 수 있지만, LoadBalancer 를 사용하기 위해서는 LoadBalancing 역할을 하는 모듈이 추가적으로 필요합니다.
  - **ClusterIP** 라는 type 은 고정된 IP, PORT 를 제공하지만, 클러스터 내부에서만 접근할 수 있는 대역의 주소가 할당됩니다.
  - **실무**에서는 주로 kubernetes cluster 에 MetalLB 와 같은 LoadBalancing 역할을 하는 모듈을 설치한 후, **LoadBalancer** type 으로 서비스를 expose 하는 방식을 사용합니다.
    - NodePort 는 Pod 이 어떤 Node 에 스케줄링될 지 모르는 상황에서, Pod 이 할당된 후 해당 Node 의 IP 를 알아야 한다는 단점이 존재합니다.