쿠버네티스 실습 4 - Deployment

1. Deployment 란?

- Deployment(디플로이먼트)는 Pod와 Replicaset에 대한 **관리**를 제공하는 단위입니다.
  - https://kubernetes.io/ko/docs/concepts/workloads/controllers/deployment/
- **관리**라는 의미는 Self-healing, Scaling, Rollout(무중단 업데이트) 과 같은 기능을 포함합니다.
- 조금 어렵다면 Deployment 는 Pod을 감싼 개념이라고 생각할 수 있습니다.
  - Pod 을 Deployment 로 배포함으로써 여러 개로 복제된 Pod, 여러 버전의 Pod 을 안전하게 관리할 수 있습니다.
- Deployment 의 자세한 구조는 생략하겠습니다.

2. Deployment 생성

- 간단한 Deployment 의 예시입니다.

```
apiVersion: apps/v1 # kubernetes resource 의 API Version
kind: Deployment # kubernetes resource name
metadata: # 메타데이터 : name, namespace, labels, annotations 등
을 포함
  name: nginx-deployment
  labels:
    app: nginx
spec: # 메인 파트 : resource 의 desired state 를 명시
  replicas: 3 # 동일한 template 의 pod 을 3 개 복제본으로 생성합니다.
  selector:
    matchLabels:
      app: nginx
```

```
  template: # Pod 의 template 을 의미합니다.
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx # container 의 이름
        image: nginx:1.14.2 # container 의 image
        ports:
        - containerPort: 80 # container 의 내부 Port
```

- 위의 스펙대로 Deployment 를 하나 생성해보겠습니다.

```
vi deployment.yaml
# 위의 내용을 복사 후 붙여넣습니다.

kubectl apply -f deployment.yaml
```

---

3. Deployment 조회

- 생성한 Deployment 의 상태를 확인합니다.

```
kubectl get deployment
# 다음과 같은 메시지가 출력됩니다.
# NAME               READY   UP-TO-DATE   AVAILABLE   AGE
# nginx-deployment   0/3     3            0           10s

kubectl get deployment,pod
```

- 시간이 지난 후, deployment 와 함께 3 개의 pod 이 생성된 것을 확인할 수 있습니다.

```
kubectl describe pod <pod-name>
```

- pod 의 정보를 자세히 조회하면 `Controlled By` 로부터 Deployment 에 의해 생성되고 관리되고 있는 것을 확인할 수 있습니다. (ReplicaSet)

---

4. Deployment Auto-healing

- pod 하나를 삭제해보겠습니다.

```
kubectl delete pod <pod-name>
```

- 기존 pod 이 삭제되고, 동일한 pod 이 새로 하나 생성된 것을 확인할 수 있습니다. (AGE 확인)

```
kubectl get pod
```

---

5. Deployment Scaling

- replica 개수를 늘려보겠습니다.

```
kubectl scale deployment/nginx-deployment --replicas=5

kubectl get deployment

kubectl get pod
```

- replica 개수를 줄여보겠습니다.

```
kubectl scale deployment/nginx-deployment --replicas=1

kubectl get deployment

kubectl get pod
```

---

5. Deployment 삭제

- deployment 를 삭제합니다.

```
kubectl delete deployment <deployment-name>

kubectl get deployment
```

```
kubectl get pod
```

- Deployment 의 Control 을 받던 pod 역시 모두 삭제된 것을 확인할 수 있습니다.
- 혹은 `-f` 옵션으로 YAML 파일을 사용해서 삭제할 수도 있습니다.

```
kubectl delete -f <YAML-파일-경로>
```