[6] Deployment & Pod Scale

1. Pod Scale out

Pod Scale Out

- Expand the number of running pods in “eshop-order” to 5.
  - 작업 클러스터: k8s
  - namespace: devops
  - deployment: eshop-order

Reference

docs에서 reference 검색 → scale

Kubectl Reference Docs

실습

```
# 작업 클러스터 이동
[user@console ~]$ kubectl config use-context k8s

# devops라는 namespace가 있는지 확인
[user@console ~]$ kubectl get namespaces devops

# namespace가 devops인 곳에서 deployment가 있는지 확인
# 확인했을 때 2개의 pod가 돌아가고 있는거 확인
[user@console ~]$ kubectl get deployments.app -n devops
NAME           READY
eshop-order    2/2
```

```
# Deployment의 ReplicaSet을 5로 늘려준다
[user@console ~]$ kubectl scale deployment eshop-order -n devops --replicas=5

# 확인했을 때 5개의 pod가 돌아가고 있는거 확인
[user@console ~]$ kubectl get deployments.app -n devops
NAME          READY
eshop-order   5/5
```

2. deployment 생성하고 scaling하기

deployment 생성하고 scaling하기

- create a deployment as follows:
  - 작업 클러스터: k8s
  - TASK:
    - name: webserver
    - 2 replicas
    - label: app_env_stage=dev
    - container name: webserver
    - container image: nginx:1.14
  - Scale Out deployment
    - Scale the deployment webserver to 3 pods

실습

```
# 조건에 맞게 deployment 생성
# webserver.yaml 파일로 YAML 파일 생성
[user@console ~]$ kubectl create deployment webserver --image=nginx:1.14 --replicas=2 --dry-run=client -o yaml > webserver.yaml

# webserver.yaml 파일 수정
[user@console ~]$ vi webserver.yaml

matchLabels:
app: webserver
되어있는 부분을 아래처럼 수정하기
```

matchLabels:

app_env_stage: dev

로 수정하기

밑에 metadata에 해당하는 labels 부분도 수정하기

container부분에 name webserver로 수정하기

:wq

수정한 yaml파일 실행

[user@console ~]$ kubectl apply -f webserver.yaml

확인

[user@console ~]$ kubectl get deployments -o wide

3개로 Scale Out

[user@console ~]$ kubectl scale deployments webserver --replicas=3

확인 (label 정보까지 포함해서)

[user@console ~]$ kubectl get pods --show-labels

기출문제 (10번)

Task: Scale the deployment webserver to 3 pods.

- 답안

  kubectl get deployments

  --> webserver라는 deployment가 있는지 확인

  kubectl scale deployment webserver --replicas=3

기출문제 (14번)

Task: Scale the deployment presentation to 6 pods.

- □□

```
# presentation□□□ deployment□ □□□ □□
kubectl get deployments presentation
```

```
kubectl scale deployment presentation --replicas=6
```

□□□□ (65□)

Create a deployment spec file that will: Launch 7 replicas of the nginx Image with the label app_runtime_stage=dev deployment name: kual00201.

Save a copy of this spec file to /opt/KUAL00201/spec_deployment.yaml (or /opt/KUAL00201/spec_deployment.json). When you are done, clean up (delete) any new Kubernetes API object that you produced during this task.

- □□

```
kubectl create deploy kual00201 --image=nginx --dry-run=client -o yaml >
/opt/KUAL00201/spec_deployment.yaml
```

```
vi /opt/KUAL00201/spec_deployment.yaml


apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app_runtime_stage: dev
  name: kual00201
spec:
  replicas: 7
  selector:
    matchLabels:
      app_runtime_stage: dev
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
```

```
      app_runtime_stage: dev
  spec:
    containers:
    - image: nginx
      name: nginx
      resources: {}
status: {}


:wq
```

```
kubectl apply -f /opt/KUAL00201/spec_deployment.yaml
```

```
# 만들어졌는지 확인
kubectl get deploy kual00201
```

```
# 만든 것 삭제하기
rm -rf /opt/KUAL00201/spec_deployment.yaml


kubectl delete deploy kual00201
```

Revision #1
Created 30 May 2023 05:26:47 by 와지
Updated 20 June 2023 13:20:12 by 와지