[실습 4] 클러스터에 동일한 설정 배포

LAB

Kubernetes Cluster

- 실습을 위한 쿠버네티스 클러스터 구성 정보 확인

```
# LAB004 실습을 위한 경로로 이동
$ cd ~/labhome/lab004

$ kubectl cluster-info
Kubernetes master is running at https://192.168.99.100:8443
KubeDNS is running at https://192.168.99.100:8443/api/v1/namespaces/kube-
system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.

# 만약 LAB 실행 중 문제가 있을 경우 아래 두가지 명령어를 이용해 복구할 수 있습니다.
$ labctl --help
Please use corret option [restore|rebuild]

   labctl restore: Quick lab restore
   labctl rebuild: Complete lab rebuild
```

ConfigMap

- 예제 파일을 이용해 ConfigMap 생성 연습
  - 애플리케이션 클러스터에 ConfigMap 으로 동일 설정 배포 및 업데이트 확인

```
# LAB004 실습을 위한 경로로 이동
$ cd ~/labhome/lab004

$ cat redis-config
maxmemory 2mb
maxmemory-policy allkeys-lru
```

```
$ kubectl create configmap example-redis-config --from-file=./redis-config
configmap/example-redis-config created

$ kubectl get configmaps
NAME                   DATA      AGE
example-redis-config   1         16s

$ kubectl describe cm example-redis-config
Name:         example-redis-config
Namespace:    default
Labels:       <none>
Annotations:  <none>

Data
====
redis-config:
----
maxmemory 2mb
maxmemory-policy allkeys-lru

Events:  <none>

$ kubectl get configmap example-redis-config -o yaml
apiVersion: v1
data:
  redis-config: |
    maxmemory 2mb
    maxmemory-policy allkeys-lru
kind: ConfigMap
metadata:
  creationTimestamp: 2018-08-20T07:09:31Z
  name: example-redis-config
  namespace: default
  resourceVersion: "1104"
  selfLink: /api/v1/namespaces/default/configmaps/example-redis-config
  uid: fc5cd4e0-a447-11e8-b754-080027ef3e31

$ cat redis-pod.yml
apiVersion: v1
kind: Pod
metadata:
  name: redis
spec:
  containers:
  - name: redis
    image: kubernetes/redis:v1
    env:
    - name: MASTER
      value: "true"
    ports:
    - containerPort: 6379
    resources:
```

```
      limits:
        cpu: "0.1"
    volumeMounts:
    - mountPath: /redis-master-data
      name: data
    - mountPath: /redis-master
      name: config
  volumes:
    - name: data
      emptyDir: {}
    - name: config
      configMap:
        name: example-redis-config
        items:
        - key: redis-config
          path: redis.conf

# Redis 서버 Pod 배포
$ kubectl create -f redis-pod.yml
pod/redis created

$ kubectl get pod
NAME      READY     STATUS    RESTARTS   AGE
redis     1/1       Running   0          4m

# 현재 배포된 Redis pod 에 터미널을 붙여 Redis 설정 내용 확인
$ kubectl exec -it redis redis-cli
127.0.0.1:6379> CONFIG GET maxmemory
1) "maxmemory"
2) "2097152"
127.0.0.1:6379> CONFIG GET maxmemory-policy
1) "maxmemory-policy"
2) "allkeys-lru"
127.0.0.1:6379>
```

Secret

- 예제 파일을 이용해 Secret 생성 연습
    - 애플리케이션 클러스터에 Secret 으로 인증 정보 배포 및 인증 과정 확인

```
# LAB004 실습을 위한 경로로 이동
$ cd ~/labhome/lab004
$ pwd
/home/jhlee/labhome/lab004

$ echo -n 'guestbook-python' | base64
Z3Vlc3Rib29rLXB5dGhvbg==
$ echo -n 'fd8d83i8dfw72r7d2' | base64
ZmQ4ZDgzaThkZnc3MnI3ZDI=
```

```
$ cat guestbook-secret.yml
apiVersion: v1
kind: Secret
metadata:
  name: guestbook-secret
data:
  username: Z3Vlc3Rib29rLXB5dGhvbg==
  password: ZmQ4ZDgzaThkZnc3MnI3ZDI=

$ kubectl create -f guestbook-secret.yml

$ kubectl get secrets
NAME                  TYPE                                  DATA      AGE
default-token-wwjtp   kubernetes.io/service-account-token   3         4h
guestbook-secret      Opaque                                2         18s

$ kubectl describe secrets guestbook-secret
Name:         guestbook-secret
Namespace:    default
Labels:       <none>
Annotations:  <none>

Type:  Opaque

Data
====
password:  17 bytes
username:  16 bytes

$ cat secret-test-pod.yml
apiVersion: v1
kind: Pod
metadata:
  name: secret-test-pod
spec:
  containers:
    - name: test-container
      image: nginx
      volumeMounts:
          # 아래 volumes: 에서 추가한 volume 이름과 같은지 확인합니다.
          - name: secret-volume
            mountPath: /etc/secret-volume
  # secret 자료는 Pod 에서 Volume 에 형태로 접근이 가능합니다.
  volumes:
    - name: secret-volume
      secret:
        secretName: guestbook-secret

$ kubectl create -f secret-test-pod.yml
pod/secret-test-pod created

$ kubectl get pod
```

```
NAME              READY     STATUS    RESTARTS   AGE
secret-test-pod   1/1       Running   0          28m

$ kubectl exec -it secret-test-pod /bin/bash

root@secret-test-pod:/# cd /etc/secret-volume
root@secret-test-pod:/etc/secret-volume# ls
password  username
root@secret-test-pod:/etc/secret-volume# cat username; echo; cat password; echo
guestbook-python
fd8d83i8dfw72r7d2
```

References

- https://kubernetes.io/docs/tutorials/configuration/configure-redis-using-configmap/
- https://kubernetes.io/docs/tasks/inject-data-application/distribute-credentials-secure/