[5] Side-car Container Pod 실행

하나의 Pod안에 nginx 어플리케이션 컨테이너가 동작하고 있다.

이 컨테이너는 varlog라는 이름을 가지는 볼륨을 마운트해서 사용하고 있으며 이 볼륨에 access.log, error.log를 기록하고 읽는다. (read, write)

동일한 Pod안데 log 분석 어플리케이션 컨테이너가 하나 더 생성된다.

이 컨테이너는 varlog 볼륨을 마운트해서 사용하고 있으며 이 볼륨에 있는 access.log, error.log를 읽고 분석한다. (read only)

이 때, 이 log분석 컨테이너를 side-car 컨테이너라고 하며 nginx 컨테이너를 main 컨테이너라고 하고 이 구조를 side-car container 구조라고 한다.

Side-car Container Pod

- An existing Pod needs to be integrated into the Kubernetes muilt-in logging architecture (e.g. kubectl logs).
  Adding a streaming sidecar container is a good and common way to accomplish this requirement.
- TASK
  - Add a sidecar container named `sidecar`, using the `busybox` image to the existing pod `eshop-cart-app`.
  - The new sidecar container has to run the following command: */bin/sh -c “tail -n+1 -F /var/log/cart-app.log”*
  - Use a volume, mounted at `/var/log`, to make the log file `cart-app.log` available to the sidecar container.
  - Don’t modify the cart-app.

Reference

docs에서 sidecar container 검색

Logging Architecture

실습

```
# eshop-cart-app pod 확인
[user@console ~]$ kubectl get pod eshop-cart-app

# 현재 running 중인 eshop-cart-app이라는 pod의 yaml 템플릿을 얻어낸 후 eshop.yaml 파일이라는 이름으로 저장
[user@console ~]$ kubectl get pod eshop-cart-app -o yaml > eshop.yaml

[user@console ~]$ ls
eshop.yaml

# docs에 나오는 ex에 나오는 템플릿 이용
# 문제에 맞춰 수정한다 -> 여기서 아래 컨테이너만 존재하는게 아니라 기존 탬플릿에
# 존재하는 컨테이너 아래에 추가하는 것임.
# 즉, 기존 파드에서 빼낸 yaml파일에서
# kind, metadata, spec 순으로 있는데
# spec안에 containers와 같은 라인에 사이드카를 적어주는 것임

spec:
  containers:
  - command:
#여기서부터 원래있던 파드 정보가 있을거고
#volumeMouts, mountPath, name이 있을건데
#거기 바로아래에 (- command:)와 같은 라인 유지
  - name: sidecar
    image: busybox
    args: [/bin/sh, -c, 'tail -n+1 -F /var/log/cart-app.log']
    volumeMounts:
    - name: varlog
      mountPath: /var/log

:wq

# 기존에 있는 pod 삭제
[user@console ~]$ kubectl delete pod eshop-cart.app

# eshop.yaml 파일을 이용해서 pod 생성
[user@console ~]$ kubectl apply -f eshop.yaml

# 생성된 pod 확인
[user@console ~]$ kubectl get pods
```

```
# 명령어를 실행했을 때 log가 확인되면 된다.
# sidecar 라는 이름의 컨테이너의 로그를 확인하겠다는 의미
[user@console ~]$ kubectl logs eshop-cart-app -c sidecar
```

□□TIP!!

Pod YAML파일에 2개의 컨테이너에 대한 정보가 들어가야 한다.

하나는 main container에 대한 내용 (로그 기록)

나머지 하나는 side-car container에 대한 내용 (로그 확인)

기출문제 (62번)

An existing Pod needs to be integrated into the Kubernetes built-in logging architecture (e. g. kubectl logs).

Adding a streaming sidecar container is a good and common way to accomplish this requirement. Task: Add a sidecar container named sidecar, using the busybox Image, to the existing Pod bigcorp-app. The new sidecar container has to run the following command: */bin/sh -c tail -n+1 -f /var/log/big-corp-app.log* Use a Volume, mounted at /var/log, to make the log file big-corp-app.log available to the sidecar container.

Context: k8s

Don’t modify the specification of the existing container other than adding the required volume mount.

- 답안

```
kubectl config use-context k8s
```

```
# bigcorp-app이라는 pod가 있는지 확인
kubectl get pods
```

```
kubectl get pod bigcorp-app -o yaml > bigcorp.yaml
```

```
vi bigcorp.yaml


...
spec:
  containers:
  ...
  - name: sidecar
    image: busybox
    args: [/bin/sh, -c, 'tail -n+1 -f /var/log/big-corp-app.log']
    volumeMounts:
    - name: varlog
      mountPath: /var/log
...


:wq
```

```
kubectl delete pod bigcorp-app
```

```
# yaml pod
kubectl apply -f bigcorp.yaml
```

```
# bigcorp-app pod sidecar
kubectl logs bigcorp-app -c sidecar
```

Revision #1
Created 30 May 2023 05:24:06 by
Updated 20 June 2023 13:20:11 by