비정형 정보 저장을 위한 Persistent Volume(AWS)

End-User 의 비정형 정보를 저장하는 AWS Provisioner 소개 및 컨테이너 마운트 실습

10분 이내 완료

INTO THE LAB

Instruction

Volumes

- /container-orchestration/yaml# 의 경로에서

```
cd volume/aws
```

Volumes : emptyDir

```
kubectl apply -f volume-emptydir.yaml
```

- 지정 컨테이너 접속 후, 파일 생성

```
kubectl exec -it shared-volumes --container redis -- /bin/bash
cd /data/shared
echo test… > test.txt
exit
```

- 다른 컨테이너로 접속 후, 파일 확인

```
kubectl exec -it shared-volumes --container nginx -- /bin/bash
cd /data/shared
ls
```

Volumes : hostPath

```
kubectl apply -f volume-hostpath.yaml
```

- Node 의 Local 디스크 경로를 Pod 에 마운트
  같은 hostPath 에 있는 볼륨은 여러 Pod 사이에서 공유
  Pod 가 삭제되어도 hostPath 에 있는 파일은 유지
  Pod 가 재기동 되어 다른 Node 에서 기동될 경우,
  새로운 Node 의 hostPath 를 사용
  Node 의 로그 파일을 읽는 로그 에이전트 컨테이너 등에 사용가능
  Pod 생성 및 확인 (Pod 내, ls -al /data/shared)

Volumes example : gitRepo

```
kubectl apply -f volume-gitrepo.yaml
```

- Pod 생성시 지정된 Git 리파지토리의 특정 리비전을
  Cloning 하여 디스크 볼륨 생성
  물리적으로는 emptyDir 이 생성되고 Git Clone 수행
  HTML 같은 정적 파일 및 Nodejs 같은 스크립트 기반코드 배포에 유용

StorageClass - Dynamic PV Provisioning

```
kubectl get storageclass
```

- 아래와 같은 결과가 출력되었는지 확인해 줍니다.

```
NAME            PROVISIONER             RECLAIMPOLICY
VOLUMEBINDINGMODE      ALLOWVOLUMEEXPANSION   AGE
gp2 (default)   kubernetes.io/aws-ebs   Delete
WaitForFirstConsumer   false                  2d1h
```

EFS 계정 생성 및 Role 바인딩

- ServerAccount 생성

```
kubectl apply -f efs-sa.yaml
```

- SA(efs-provisioner)에 권한(rbac) 설정

```
kubectl apply -f efs-rbac.yaml
```

efs-provisioner-deploy.yaml 파일에서
value: #`{efs system id}` => 파일 시스템 ID
value: # `{aws region}` => EKS 리전
server: # `{file-system-id}`.efs.`{aws-region}`.amazonaws.com

- provisioner 설치

```
kubectl apply -f efs-provisioner-deploy.yaml
```

- StorageClass 등록, 조회

```
kubectl apply -f efs-storageclass.yaml
kubectl get sc
```

- 아래와 같은 결과가 출력되었는지 확인해 줍니다.

```
NAME            PROVISIONER             RECLAIMPOLICY
VOLUMEBINDINGMODE      ALLOWVOLUMEEXPANSION   AGE
aws-efs         my-aws.com/aws-efs      Delete          Immediate
false                  4s
gp2 (default)   kubernetes.io/aws-ebs   Delete
WaitForFirstConsumer   false                  2d2h
```

pvc 생성

```
kubectl apply -f volume-pvc.yaml
```

- 아래와 같은 결과가 출력되었는지 확인해 줍니다.
  persistentvolumeclaim/aws-efs created

pvc 조회

```
kubectl get pvc
```

- 아래와 같은 결과가 출력되었는지 확인해 줍니다.

```
NAME      STATUS    VOLUME   CAPACITY   ACCESS MODES   STORAGECLASS
AGE
aws-efs   Bound                                       aws-efs
59s
```

Create Pod with PersistentVolumeClaim

```
kubectl apply -f pod-with-pvc.yaml
kubectl get pod
kubectl describe pod mypod
kubectl exec -it mypod -- /bin/bash
cd /mnt/aws
```

df -k 로 **PVC** 마운트 확인

CheckPoints

1. 모든 요구사항을 만족하는가 ☐