



# Kubernetes Storage

⋮

## 1. Kubernetes Volume의 종류

- hostPath

- hostPath

- 노드의 파일시스템의 디렉토리나 파일을 컨테이너에 마운트
- 노드에 디렉토리나 파일을 생성하여 마운트 가능
- hostPath는 type 지시어를 이용해 mount 구성의 요구를 추가할 수 있다.

volumes:

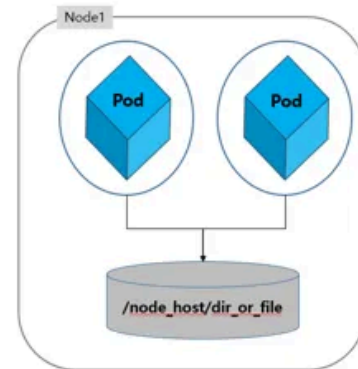
- name: html

hostPath:

path: /hostdir\_or\_file

type: DirectoryOrCreate

DirectoryOrCreate	주어진 경로에 아무것도 없다면, 필요에 따라 kubelet의 소유권, 권한을 0755로 설정한 빈 디렉토리를 생성한다.
Directory	주어진 경로에 디렉터리가 있어야 함
FileOrCreate	주어진 경로에 아무것도 없다면, 필요에 따라 kubelet의 소유권, 권한을 0755로 설정한 file을 생성한다.
File	주어진 경로에 파일이 있어야 함



volumes:

- name: html

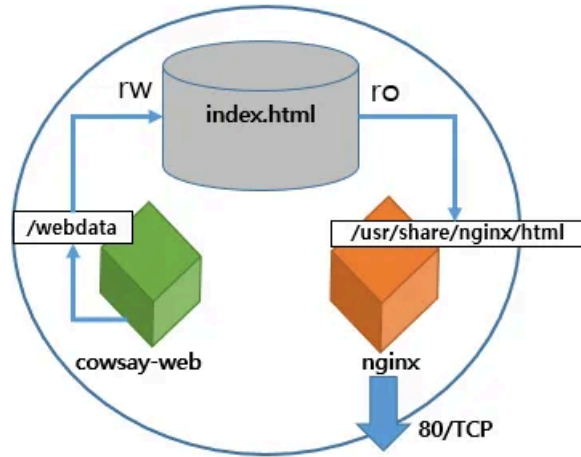
hostPath:

path: /hostdir\_or\_file

- emptyDir

- emptyDir volume

- emptyDir 볼륨은 빈 디렉토리로 시작
- Pod 내부에서 실행중인 애플리케이션은 필요한 모든 파일을 작성
- Pod를 삭제하면 볼륨의 내용이 손실됨
- 동일한 Pod에서 실행되는 컨테이너 간에 파일을 공유할 때 유용



## volume-empty.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: dynamic-web
spec:
  containers:
    - image: smlinux/cowsay-web
      name: web-generator
      volumeMounts:
        - name: html
          mountPath: /webdata
    - image: nginx:1.14
      name: web-server
      volumeMounts:
        - name: html
          mountPath: /usr/share/nginx/html
          readOnly: true
  ports:
    - containerPort: 80
  volumes:
    - name: html
      emptyDir: {}
```

## 2. Persistent Volume & Persistent Volume Claim

- PV(PersistentVolume)

```
pv.yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pvname
spec:
  capacity:
    storage: <storage_size>
  accessModes:
    - ReadWriteOnce
    - ReadOnlyMany
  persistentVolumeReclaimPolicy: Retain
  nfs:
    server: <NFS_Server>
    path: <Share_Storage>
```



Persistent  
Volume  
pool을 생성

### Persistent Volume (PV)



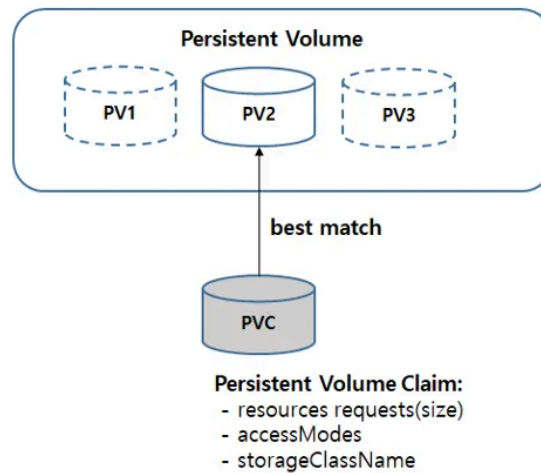
### Persistent Volume

- hostPath
- AWS\_EBS
- GCE\_PD
- NFS

- PVC(PersistentVolumeClaim)

- 디스크 요청

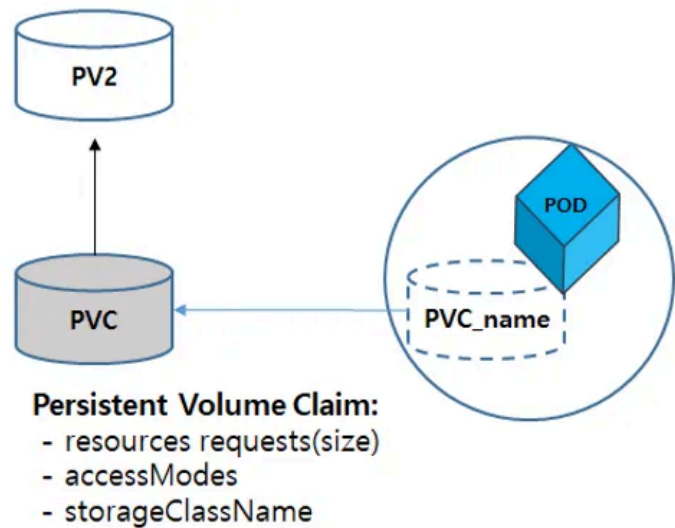
```
pvc.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-name
spec:
  resources:
    requests:
      storage: size
  accessModes:
    - ReadWriteOnce
    - ReadOnlyMany
  storageClassName: "manuel"
```



- Pod에 PersistentVolume 마운트

```
pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: pod
spec:
  containers:
    - image: image
      name: container-name
      volumeMounts:
        - name: volume-name
          mountPath: /mount/dir
  volumes:
    - name: volume-name
      persistentVolumeClaim:
        claimName: pvc-name
```

### Persistent Volume



- ▶ 실습

## • StorageClass

- Kubernetes 클러스터에서 동적 볼륨 프로비저닝(Dynamic Volume Provisioning)을 가능하게 하는 리소스
- 이를 통해 사용자는 특정한 스토리지 요구 사항을 정의하고, 해당 요구 사항에 맞는 볼륨을 자동으로 생성할 수 있음
- Dynamic Volume Provisioning
  - StorageClass에 스토리지 요구사항을 정의
  - **PersistentVolumeClaim**을 통해 특정 스토리지를 요구
  - 쿠버네티스는 정의된 StorageClass를 기반으로 적절한 스토리지 백엔드를 통해 자동으로 Persistent Volume(PV)을 생성해서 할당해줌
- 주요 개념
  - **Provisioner**: 스토리지를 프로비저닝하는 백엔드 드라이버 (예: AWS EBS, GCE PD, NFS 등)
  - **Reclaim Policy**: 스토리지 삭제 정책 ( **Delete** , **Retain** , **Recycle** )
  - **Volume Binding Mode**: 볼륨이 언제 바인딩되는지 결정 ( **Immediate** , **WaitForFirstConsumer** )
  - **Parameters**: 특정 스토리지 백엔드에 대한 설정 (예: EBS 볼륨 유형, 크기, IOPS 등)
- StorageClass 주요 속성

속성	설명
provisioner	스토리지를 제공하는 CSI 드라이버 또는 기본 프로비저너 ( <b>ebs.csi.aws.com</b> , <b>pd.csi.storage.gke.io</b> , <b>csi.vsphere.vmware.com</b> 등)
parameters	백엔드 스토리지 설정 (예: 디스크 유형, 크기, IOPS 등)
reclaimPolicy	삭제 정책 ( <b>Delete</b> : 삭제 시 PV도 제거, <b>Retain</b> : PV를 유지)
volumeBinding Mode	볼륨 할당 시점 <b>Immediate</b> : 즉시 할당. PVC가 생성되자마자, 즉시 PV와 바인딩. <b>WaitForFirstConsumer</b> : Pod가 실행되어 volume mount 요청할 때 할당. 클라우드 환경(AWS, GCP, Azure)에서 다중 가용 영역(AZ) 지원 시 유용
allowVolumeExpansion	볼륨 크기 확장 가능 여부 ( <b>true</b> / <b>false</b> )

- StorageClass 사용 사례
  - AWS EBS: **provisioner: ebs.csi.aws.com**
  - Google Cloud Persistent Disk (GCE PD): **provisioner: pd.csi.storage.gke.io**
  - Azure Disk: **provisioner: disk.csi.azure.com**
  - NFS, GlusterFS 등: 다양한 외부 스토리지 지원 가능

## ◦ StorageClass 관련 명령어

```
# 현재 StorageClass 목록 확인 kubectl get storageclass #특정 StorageClass 상세 정보 조회 kubectl describe storageclass gp2 #StorageClass 삭제 kubectl delete storageclass gp2
```

## ◦ StorageClass 예제

- AWS EBS를 사용하는 StorageClass의 예제

```
apiVersion: storage.k8s.io/v1 kind: StorageClass metadata: name: standard provisioner:
kubernetes.io/aws-ebs parameters: type: gp2 reclaimPolicy: Retain allowVolumeExpansion: true
mountOptions: - debug volumeBindingMode: Immediate
```

- LAB: 간단한 storageclass 생성

```
cat << END > storageclass.yaml apiVersion: storage.k8s.io/v1 kind: StorageClass metadata: name:
app-hostpath-sc provisioner: kubernetes.io/no-provisioner allowVolumeExpansion: true END kubectl
apply -f storageclass.yaml kubectl get storageclasses.storage.k8s.io NAME PROVISIONER
RECLAIMPOLICY VOLUMEBINDINGMODE ALLOWVOLUMEEXPANSION AGE app-hostpath-sc kubernetes.io/no-
provisioner Delete Immediate true 13s
```

- 정적 스토리지클래스
  - Kubernetes에서 **StorageClass**는 일반적으로 동적 볼륨 프로비저닝(Dynamic Volume Provisioning)을 지원
  - 하지만 `kubernetes.io/no-provisioner` 를 사용하면, Kubernetes가 자동으로 PersistentVolume(PV)을 생성하지 않도록 설정할수 있음
  - 즉, 관리자가 직접 PersistentVolume(PV)을 생성해야 함
  - 로컬 SSD, HDD 등 특정 노드의 물리적 스토리지를 사용할 때 사용함

```
apiVersion: storage.k8s.io/v1 kind: StorageClass metadata: name: local-storage provisioner:
kubernetes.io/no-provisioner volumeBindingMode: WaitForFirstConsumer
```

## 2. 기출 문제 문제

### ? PV 생성 4%

- `kubectl config use-context k8s`
- Create a persistent volume with name `app-config` , of capacity `1Gi` and access mode `ReadWriteMany` .
- storageClass: `az-c`  
The type of volume is `hostPath` and its location is `/srv/app-config` .

▶ 답