

18. Volumes

Volume은 왜 필요 한가요?

Container

azx876gke4	34 mb
az9tgu76rre	432 mb
t3r2fff65e45	128.5 kb
d64508f65e	7 kb

Volume은 왜 필요 한가요?

Container	
azx876gke4	34 mb
az9tgu76re	432 mb
t3r2fff65e45	128.5 kb
d64508f65e	7 kb



Image Layer for **read only**

Volume은 왜 필요 한가요?

Container	
Container layer	
azx876gke4	34 mb
az9tgu76re	432 mb
t3r2ff65e45	128.5 kb
d64508f65e	7 kb

→ Container Layer for read write(R/W)

} Image Layer for read only(R/O)

Volume은 왜 필요 한가요?

Container	
Container layer	
azx876gke4	34 mb
az9tgu76rre	432 mb
t3r2fff65e45	128.5 kb
d64508f65e	7 kb

→ Container Layer for **read write**

} Image Layer for **read only**

Container가 **종료** 된다면?

Volume은 왜 필요 한가요?

Container	
Container layer	
azx876gke4	34 mb
az9tgu76rre	432 mb
t3r2fff65e45	128.5 kb
d64508f65e	7 kb

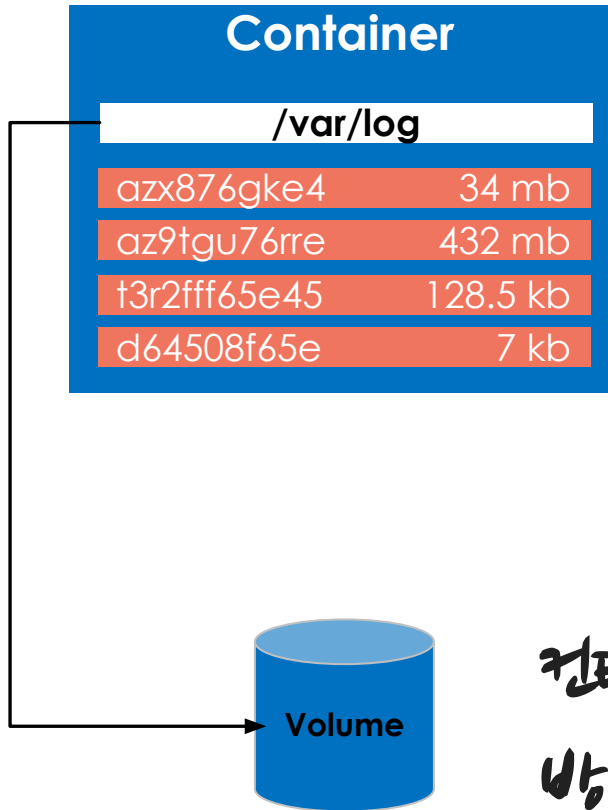
→ Container Layer for **read write**

} Image Layer for **read only**

Container Layer 에 존재 하는 모든
데이터는 **Stateless** 입니다.

Container 가 종료 되면 사라집니다.

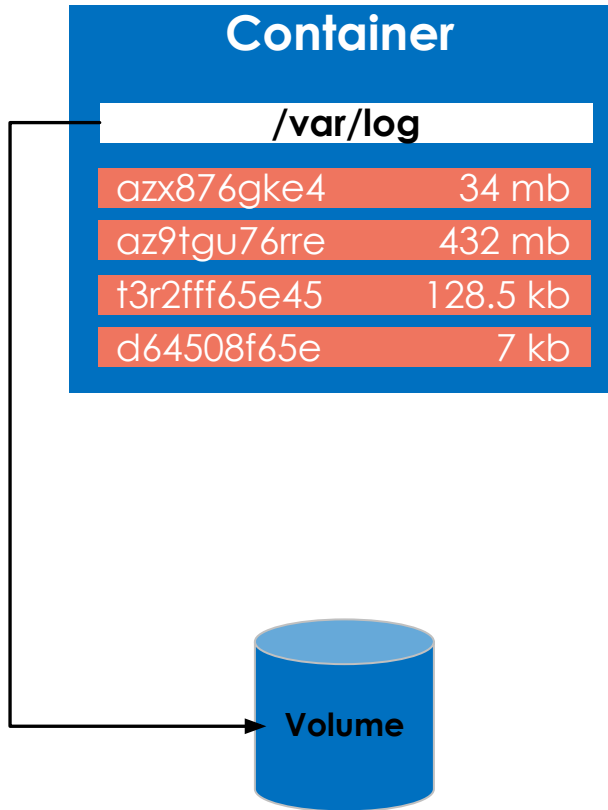
Volume이란 무엇인가?



컨테이너가 종료 되어도 데이터를 유지 할 수 있는
방법은 외부 디스크 볼륨을 마운트 하는 것

노드의 디스크, 클라우드 볼륨, Git 등

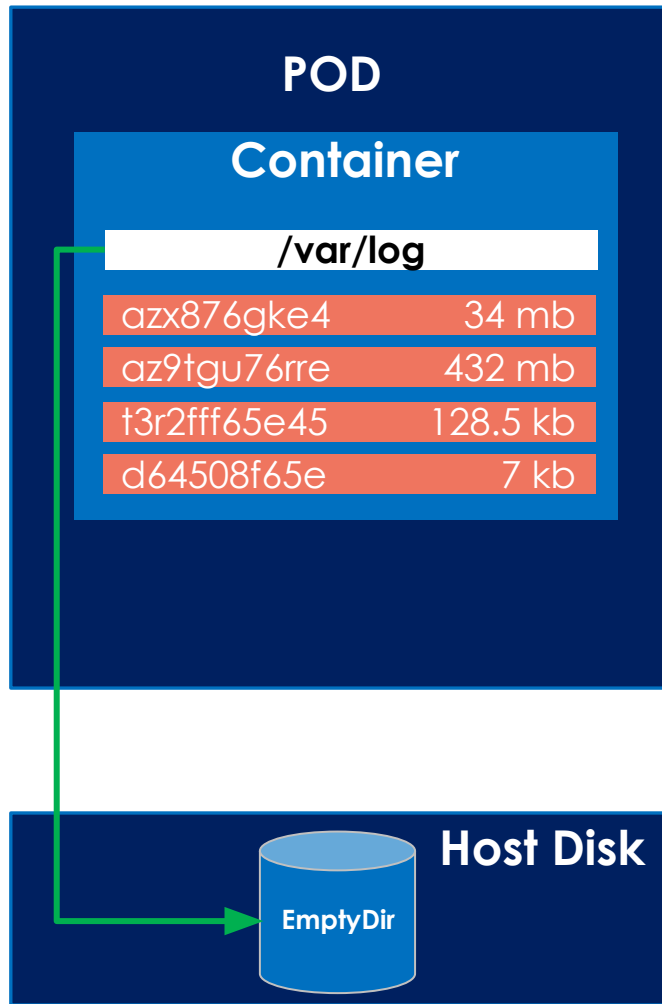
Volume이란 무엇인가?



쿠버네티스 에서 지원 하는 볼륨은 다양한 종류가 있습니다. 지금 부터 하나씩 알아 보도록 하겠습니다.

Volume이란 무엇인가?

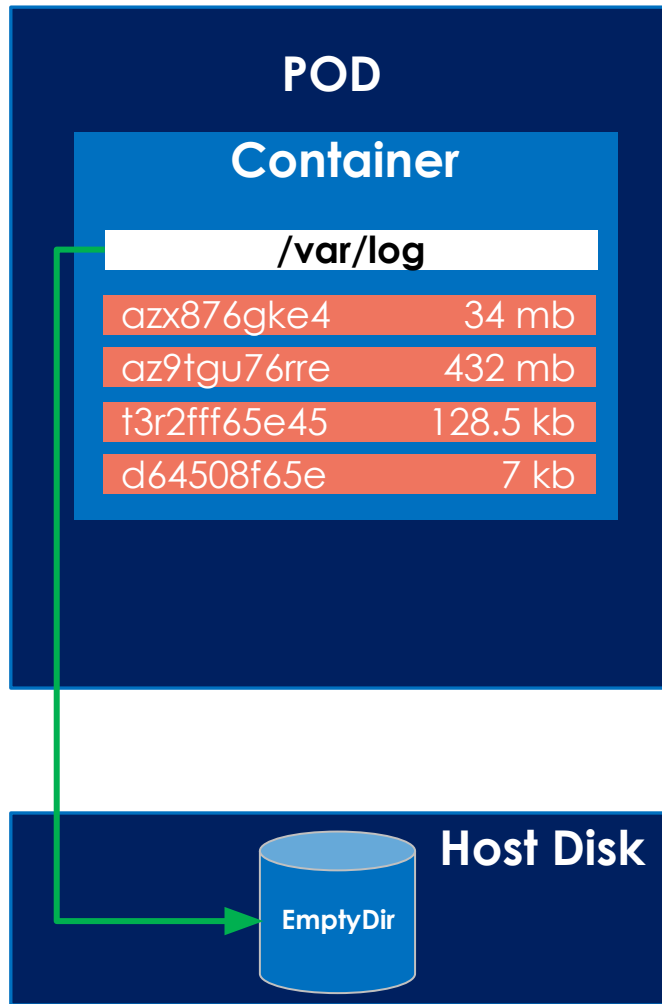
EmptyDir



EmptyDir 은 Pod 가 종료
되면 영구적으로 삭제 됩니다.
즉, Pod 와 동일한 Lifecycle 을 가집니다.

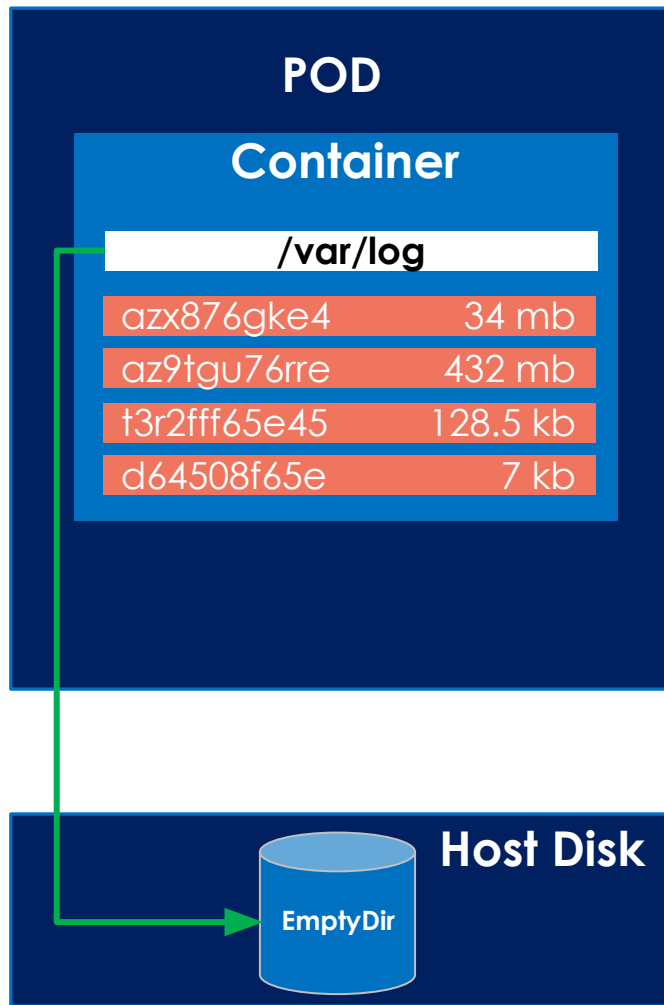
Volume이란 무엇인가?

EmptyDir



EmptyDir은 그럼 무슨 용도로 사용되나?

Volume이란 무엇인가?



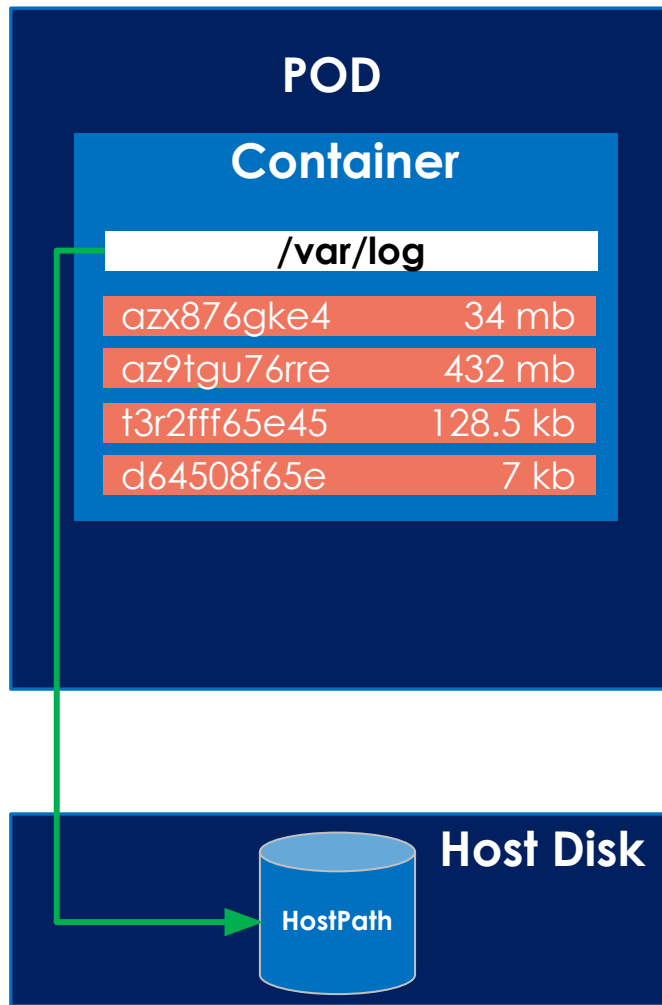
EmptyDir

HostPath

EmptyDir은 그럼 무슨 용도로 사용되나?

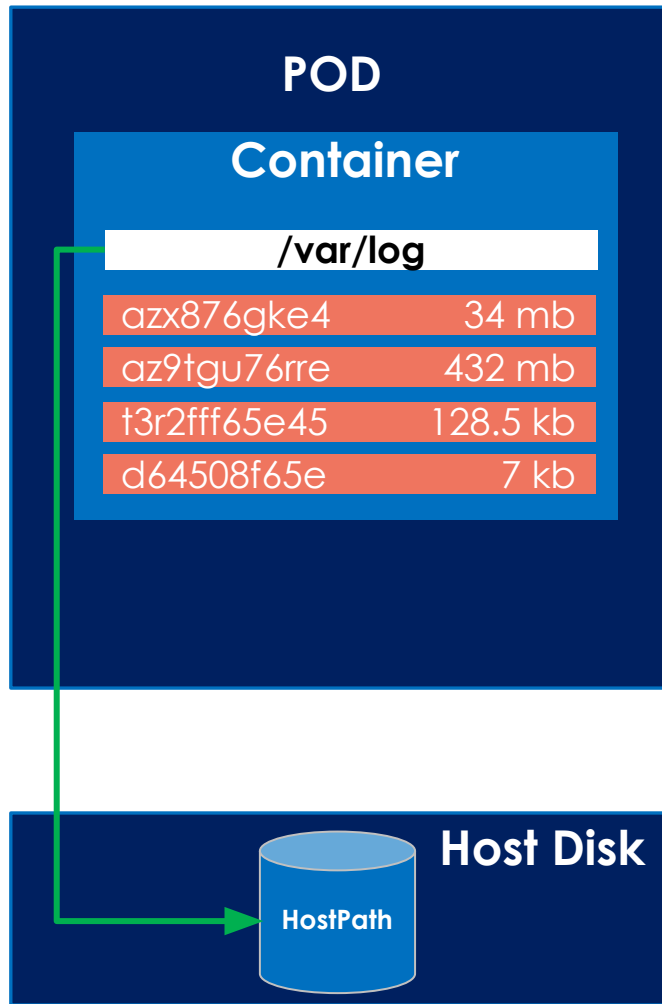
- 대규모 파일 기반 **Sorting** 작업
- Pod내의 컨테이너간 파일교환
- 복구를 위한 **임시파일** 보관

Volume이란 무엇인가?



HostPath는 Pod가 종료
되어도 상태가 유지됩니다.

Volume이란 무엇인가?



EmptyDir

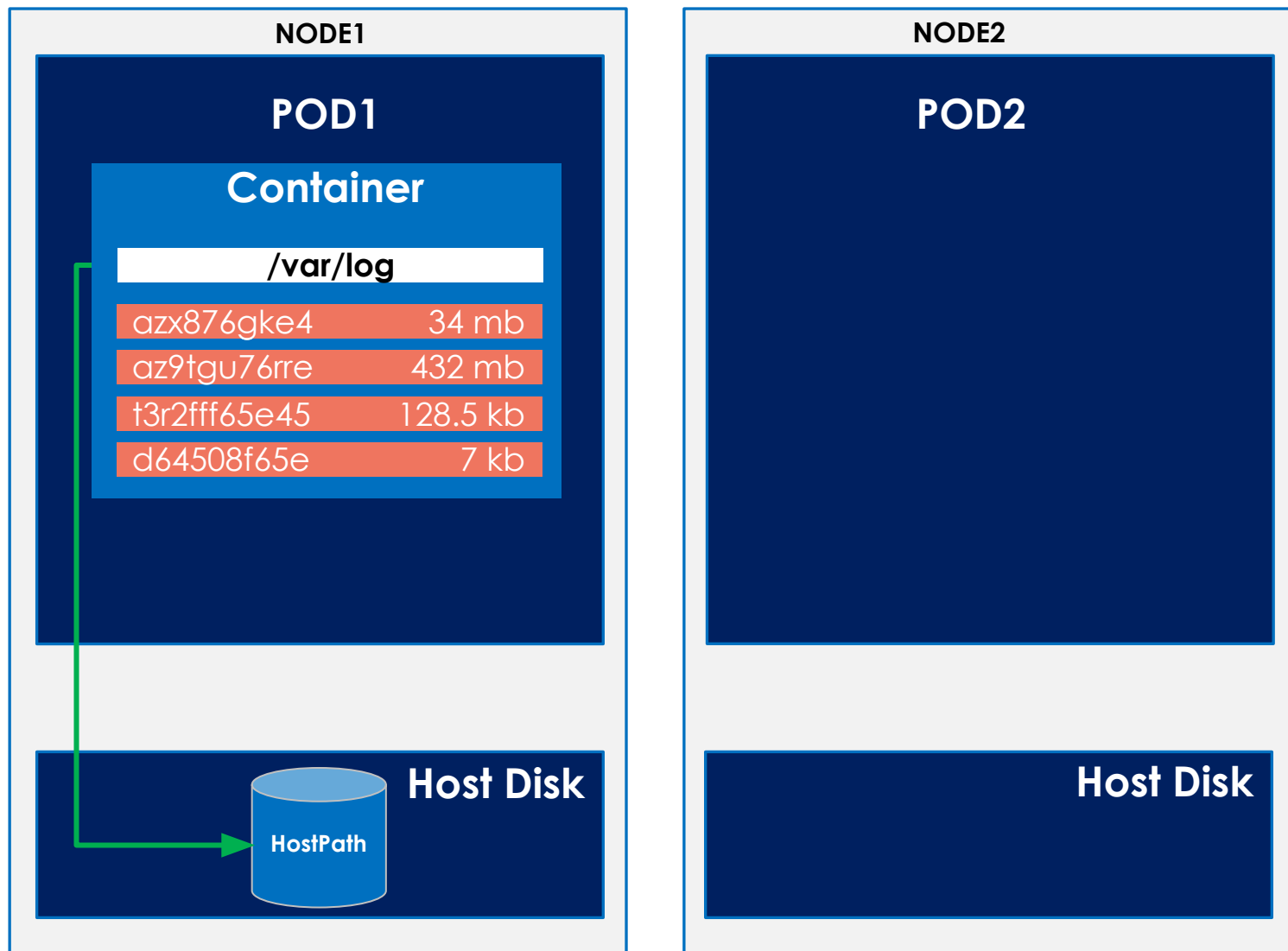
HostPath

HostPath는 Pod가 종료
되어도 **상태가 유지** 됩니다.
다만, POD는 항상 다른 노드로 **이동** 가능성
이 존재 합니다!!!!

Volume 이란 무엇인가?

EmptyDir

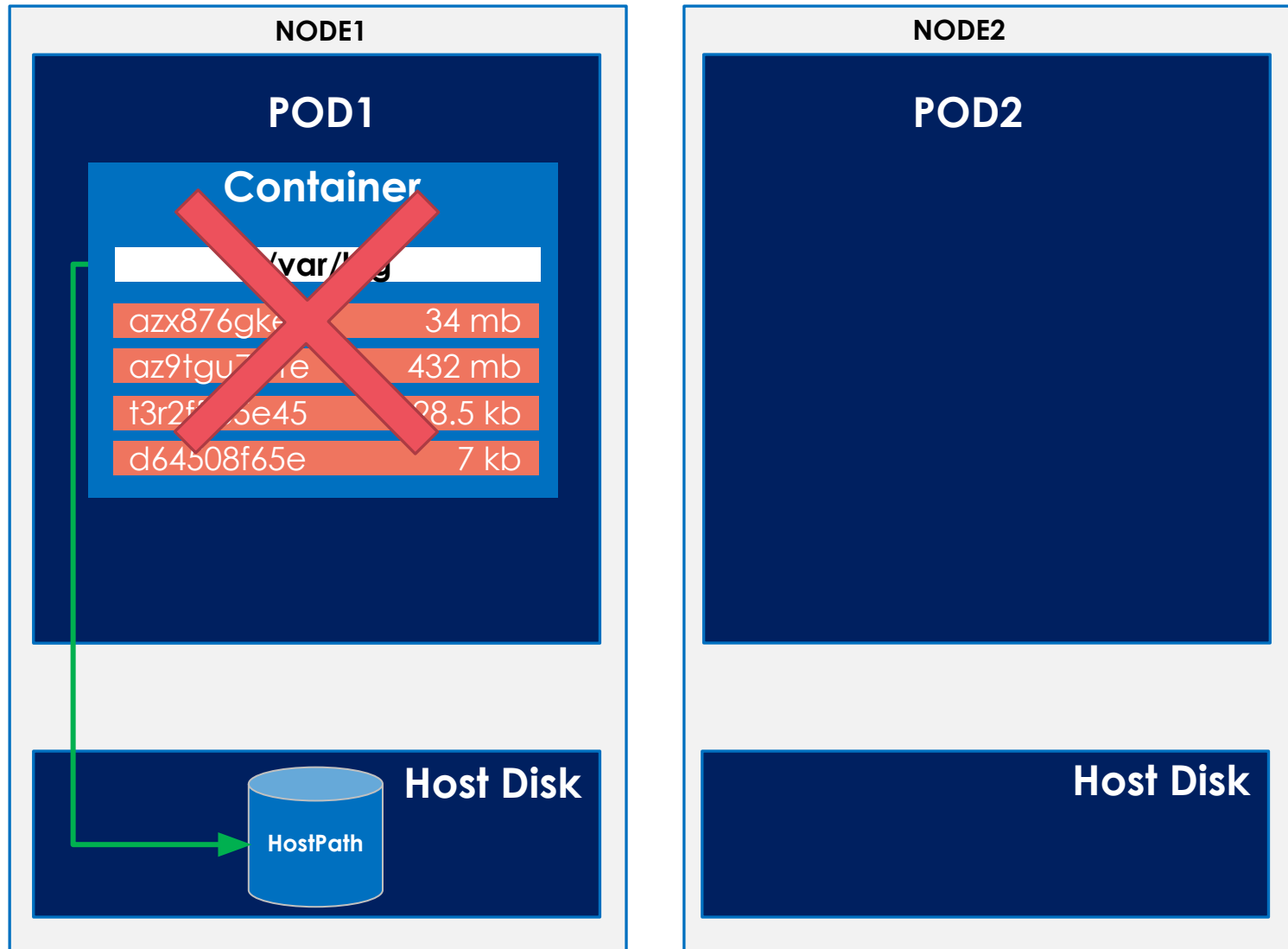
HostPath



Volume 이란 무엇인가?

EmptyDir

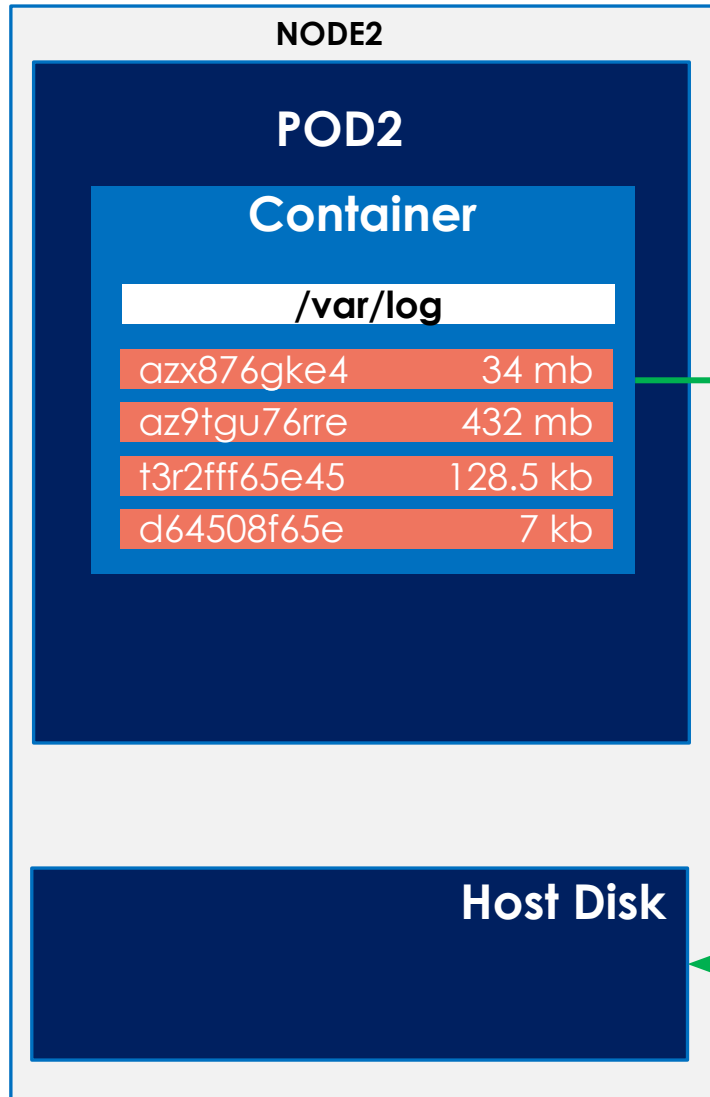
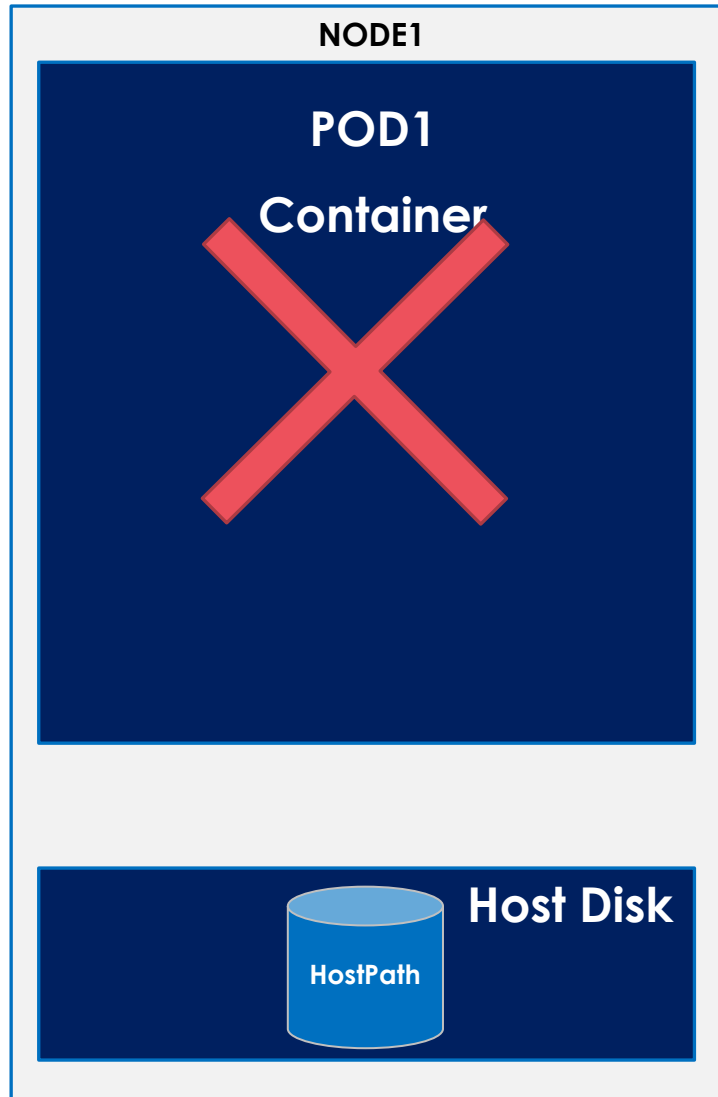
HostPath



Volume이란 무엇인가?

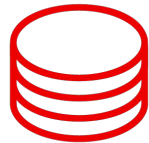
EmptyDir

HostPath



그래서 쿠버네티스에서는
Persistent Volume(PV) 이 존재
합니다.

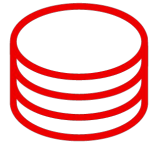
Persistent Volume



물리적 볼륨

Persistent Volume

CSI (Container Storage Interface)



물리적 볼륨

Persistent Volume Access Mode (2024.1 현재)

Volume Plugin	ReadWriteOnce	ReadOnlyMany	ReadWriteMany
AWSElasticBlockStore	✓	-	-
AzureFile	✓	✓	✓
AzureDisk	✓	-	-
CephFS	✓	✓	✓
Cinder	✓	-	-
CSI	depends on the driver	depends on the driver	depends on the driver
FC	✓	✓	-
FlexVolume	✓	✓	depends on the driver
Flocker	✓	-	-
GCEPersistentDisk	✓	✓	-
Glusterfs	✓	✓	✓
HostPath	✓	-	-
iSCSI	✓	✓	-
Quobyte	✓	✓	✓
NFS	✓	✓	✓
RBD	✓	✓	-
VsphereVolume	✓	-	- (works when Pods are collocated)
PortworxVolume	✓	-	✓
ScaleIO	✓	✓	-
StorageOS	✓	-	-

Persistent Volume Access Mode

ReadWriteOnce (**RWO**)

하나의 노드에서 읽기/쓰기로 마운트 합니다. RWO 모드는 **하나의 노드에서 여러개의 POD가 실행중**이라면 볼륨에 접근이 가능 합니다.

ReadOnlyMany (**ROX**)

여러노드에서 읽기 전용으로 마운트된 볼륨

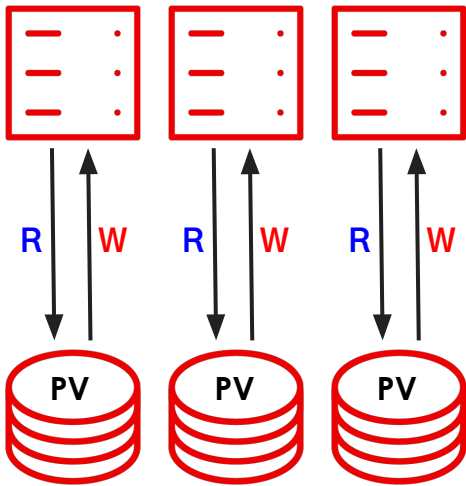
ReadWriteMany (**RWX**)

여러 노드에서 읽기-쓰기로 마운트된 볼륨

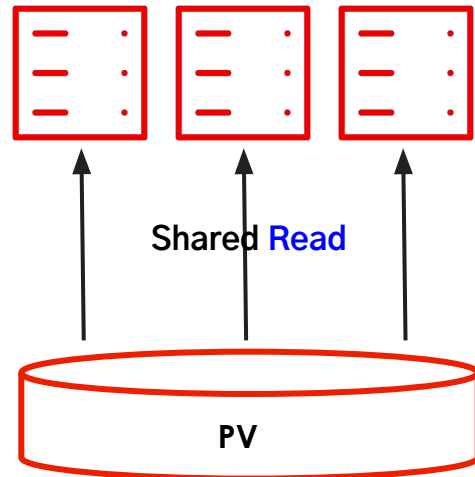
ReadWriteOncePod (**RWOP**)

Persistent Volume Access Mode

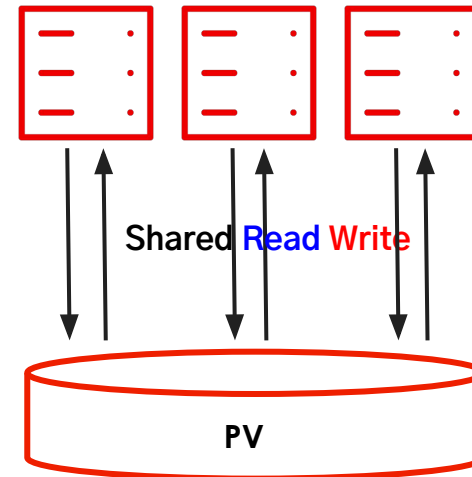
ReadWriteOnce
(RWO)



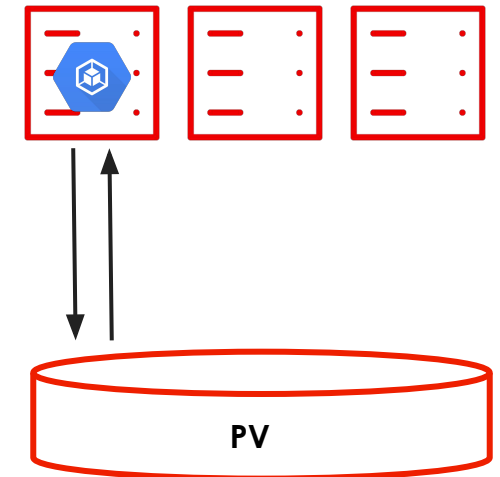
ReadWriteOnce
(ROX)



ReadWriteOnce
(RWX)



ReadWriteOncePOD
(RWOP)



Persistent Volume Access Mode

동적 프로비저닝

정적 프로비저닝

Volume 특징 정리

- Pod 에서 실제 데이터가 있는 디렉토리를 보존 하기 위해 사용
- Pod 의 일부로 정의 되며, Pod 와 라이프사이클을 같이함
- 독립적인 쿠버네티스 오브젝트가 아니며 스스로 생성 하거나 삭제할 수 없다(kind 가 없다)
- 마운트(mount)를 통해서만 사용 할 수 있다
- 실제 호스트에서는 **/var/lib/kubelet/pods/PODUID/volumes/** 위치에서 볼륨 확인 가능

Volume 의 종류

일반

Volume

종류	설명
emptyDir	일시적인 데이터를 저장하는데 사용. 비어있는 단순한 디렉토리
hostPath	노드의 파일 시스템에서 Pod의 디렉토리로 마운트 하는 데 사용
gitRepo	Git 리포지토리를 Checkout 해서 초기화 하는 볼륨
nfs	Pod 에 마운트된 네트워크 파일 시스템
gcePersistentDisk	구글 클라우드 영구 디스크
awsElasticBlockStore	아마존 클라우드 영구 디스크
azureDisk	애저 클라우드 영구 디스크
기타 (cinder,cephfs,iscsi,flocke r,quobyte,glusterfs 등)	다양한 스토리지

특수한

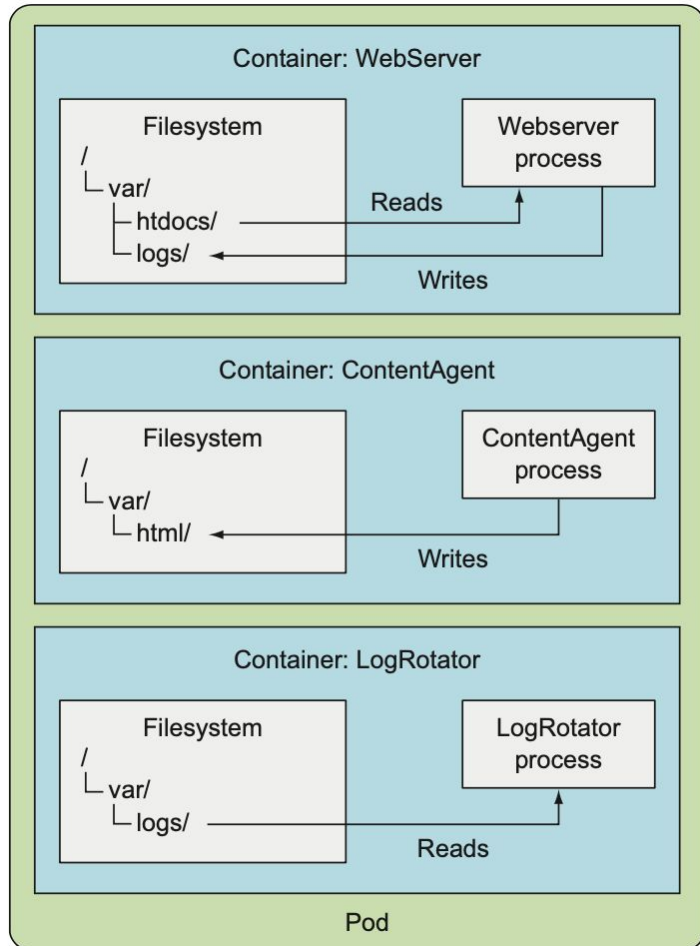
Volume

종류	설명
configMap	설정파일 및 환경 변수 보관 하는 볼륨
secret	기밀 유지가 필요한 설정 파일 이나 환경 변수 보관 하는 볼륨

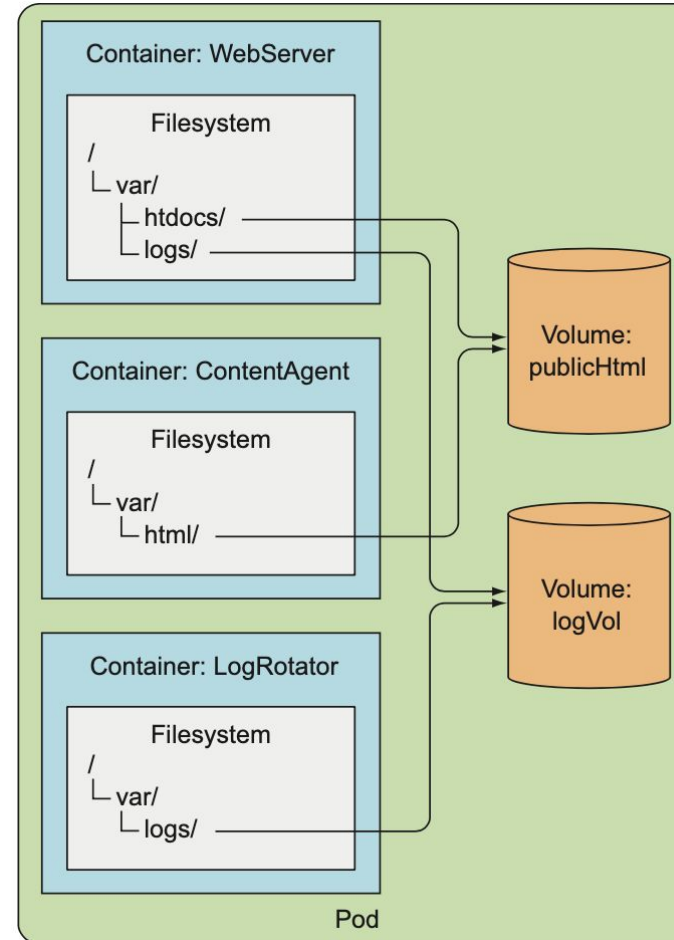
Volume 을 사용한 컨테이너 사이의 데이터 공유

공유 예 : 일반적인 애플리케이션 컨테이너가 로그를 기록하고, 또다른 컨테이너는 로그를 읽어서 로그 서버로 전송

공유 스토리지 사용 하지 않는 경우



볼륨을 공유 하는 경우



emptyDir 볼륨

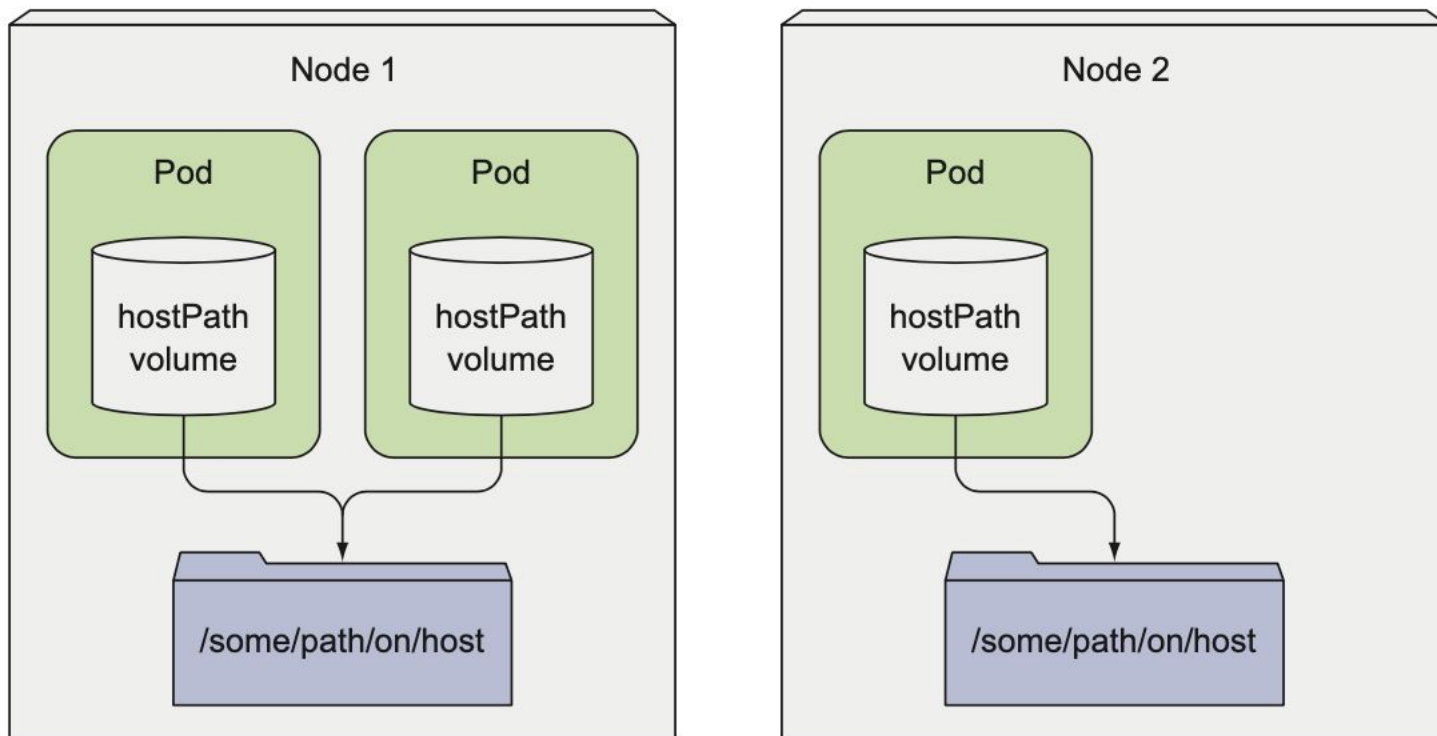
```
spec:
  containers:
    - image: dangtong/fortune
      name: html-generator
      volumeMounts:
        - name: html
          mountPath: /var/htdocs
    - image: nginx:alpine
      name: web-server
      volumeMounts:
        - name: html
          mountPath: /usr/share/nginx/html
          readOnly: true
      ports:
        - containerPort: 80
          protocol: TCP
  volumes:
    - name: html
      emptyDir: {}
```

→ 설정된 볼륨을 마운트함

→ 볼륨 설정

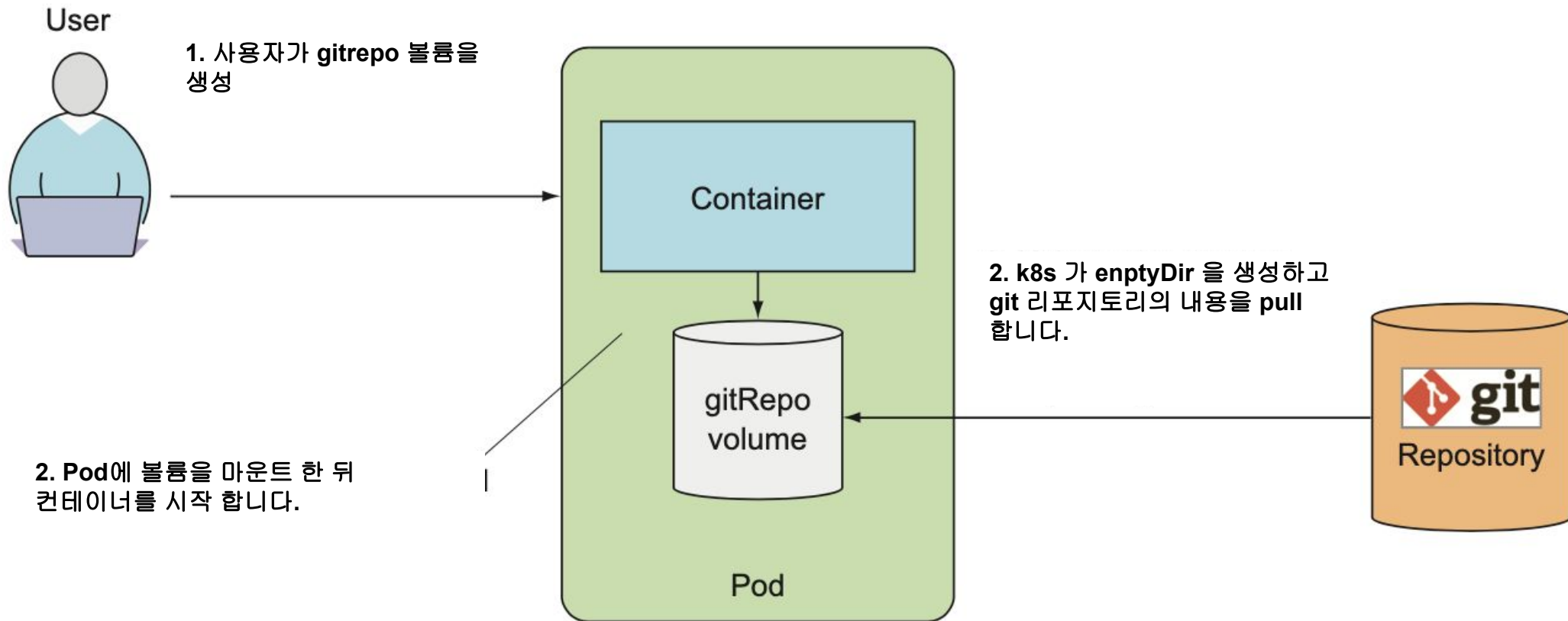
HostPath 볼륨

- 노드의 파일시스템에 있는 특정 파일 또는 디렉토리를 가지는 볼륨
- Pod는 삭제 되도 Host볼륨은 삭제 되지 않음
- Pod를 재 시작 해도, 동일한 노드에 스케줄 되어야 이전 데이터 를 볼 수 있음.
- 일반적인 목적의 Pod 의 경우에 사용 하는 것이 좋지 않음



GitRepo 볼륨

- Git 리포지토리를 복제해 만들어지는 emptyDir 볼륨
- 웹애플리케이션의 정적 콘텐츠를 제공하는 볼륨으로 사용함



GitRepo 볼륨 사용

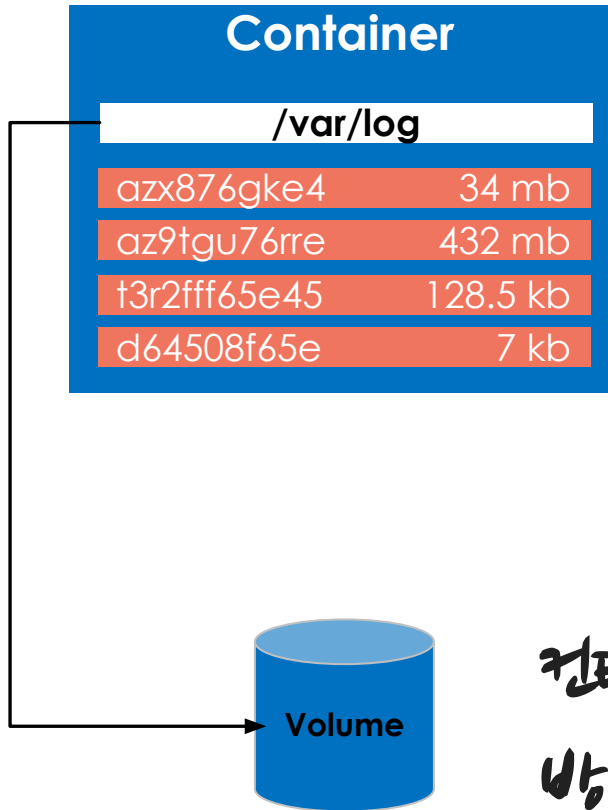
```
spec:
  containers:
  - image: nginx:alpine
    name: web-server
    volumeMounts:
    - name: html
      mountPath: /usr/share/nginx/html
      readOnly: true
  ports:
  - containerPort: 80
    protocol: TCP
```

→ 설정된 볼륨을 마운트함

```
volumes:
- name: html
  gitRepo:
    repository: https://github.com/dangtong76/k8s-website.git
    revision: master
    directory: .
```

→ 볼륨 설정

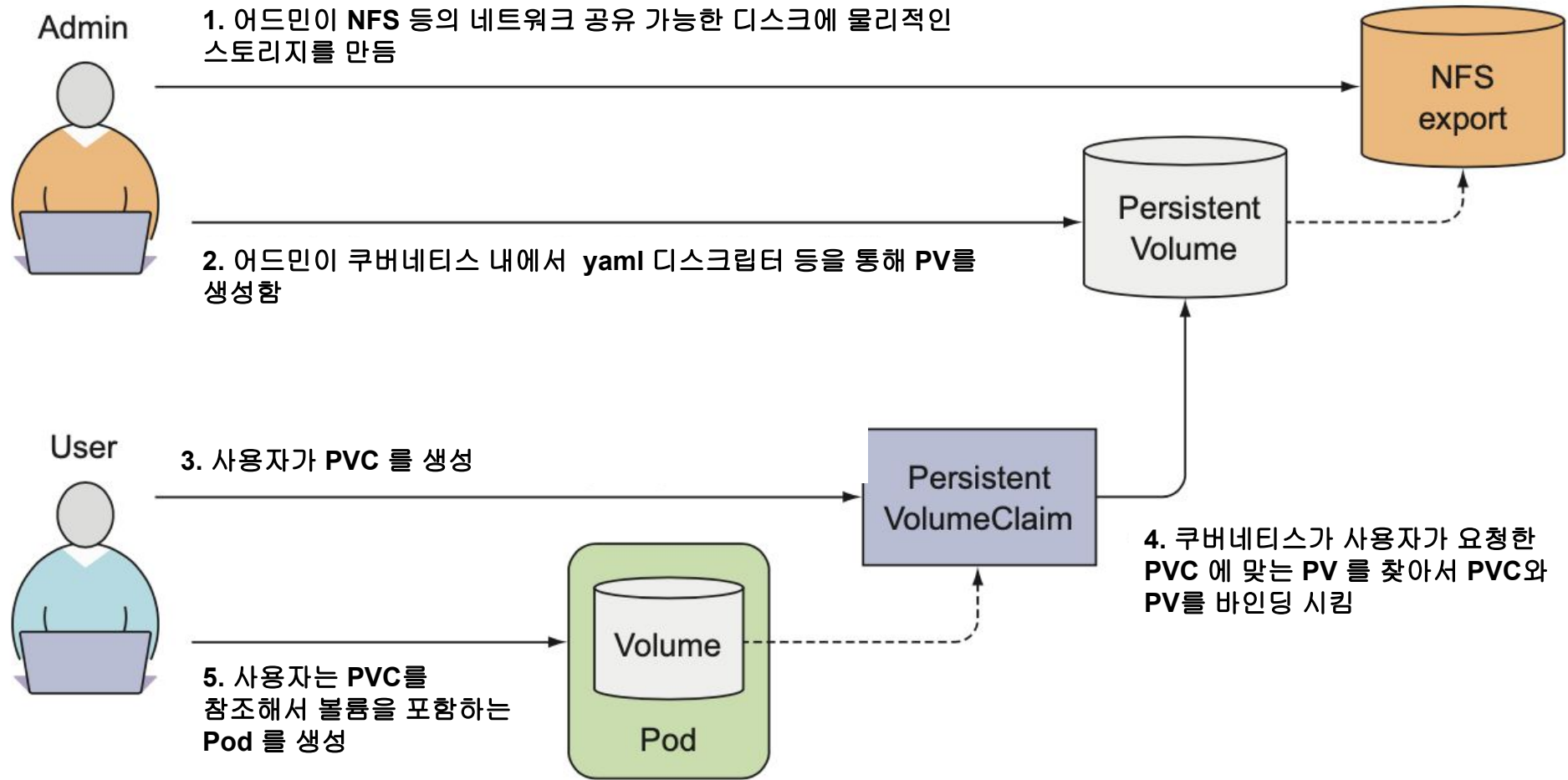
Volume이란 무엇인가?



컨테이너가 종료 되어도 데이터를 유지 할 수 있는
방법은 외부 디스크를 마운트 하는 것

노드의 디스크, 클라우드 볼륨, Git 등

PersistentVolume 과 PersistentVolumeClaim



PersistentVolumeReclaimPolicy 및 AccessMode

PersistentVolumeReclaimPolicy

Policy	설명
Retain(유지)	PVC 를 삭제 해도 PV를 유지 함. 데이터는 살아 있지만 수동으로 정리해야하는 단점
Delete(삭제)	관련된 볼륨 모두 삭제
Recycle(재사용)	볼륨 자체를 삭제 하지는 않고 <code>rm -rf /*</code> 로 모든 데이터만 삭제

AccessMode

Policy	설명
ReadWriteOnce	하나의 node 에서 볼륨을 읽기-쓰기로 마운트할 수 있다
ReadOnlyMany	여러 node 에서 볼륨을 읽기 전용으로 마운트할 수 있다
ReadWriteMany	여러 node 에서 볼륨을 읽기-쓰기로 마운트할 수 있다

PV 및 PVC 사용

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: mongodb-pv
spec:
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteOnce
    - ReadOnlyMany
  persistentVolumeReclaimPolicy: Retain
  gcePersistentDisk:
    pdName: mongodb
    fsType: ext4
```

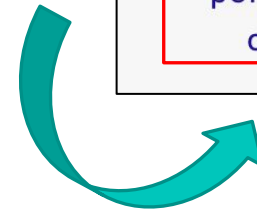
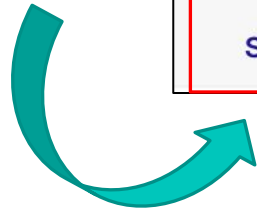
PV

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mongodb-pvc
spec:
  resources:
    request:
      storage: 1Gi
  accessModes:
    - ReadWriteOnce
  storageClassName: ""
```

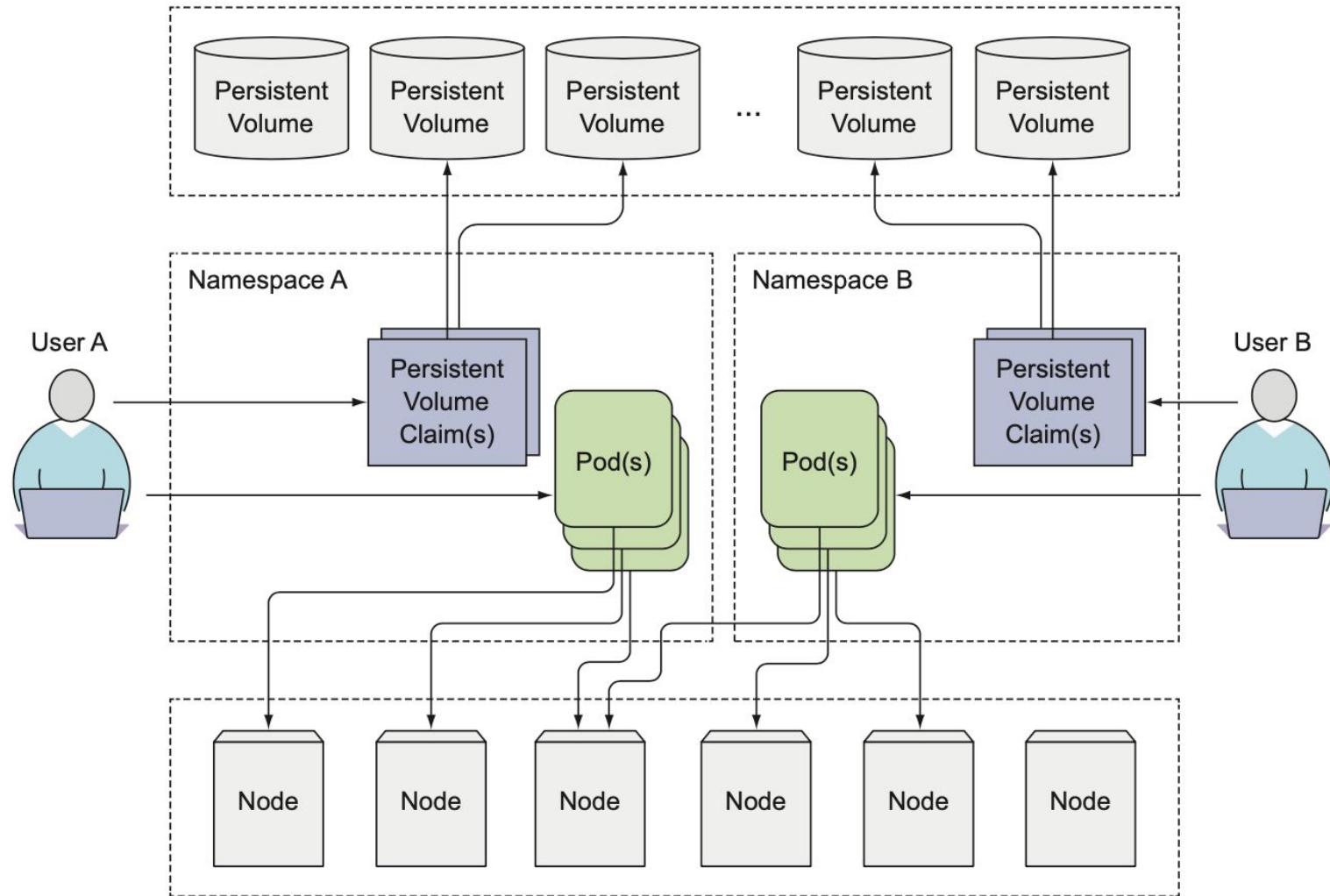
PVC

```
spec:
  containers:
    - image: mongo
      name: mongodb
      volumeMounts:
        - name: mongodb-data
          mountPath: /data/db
      ports:
        - containerPort: 27017
          protocol: TCP
  volumes:
    - name: mongodb-data
      persistentVolumeClaim:
        claimName: mongodb-pvc
```

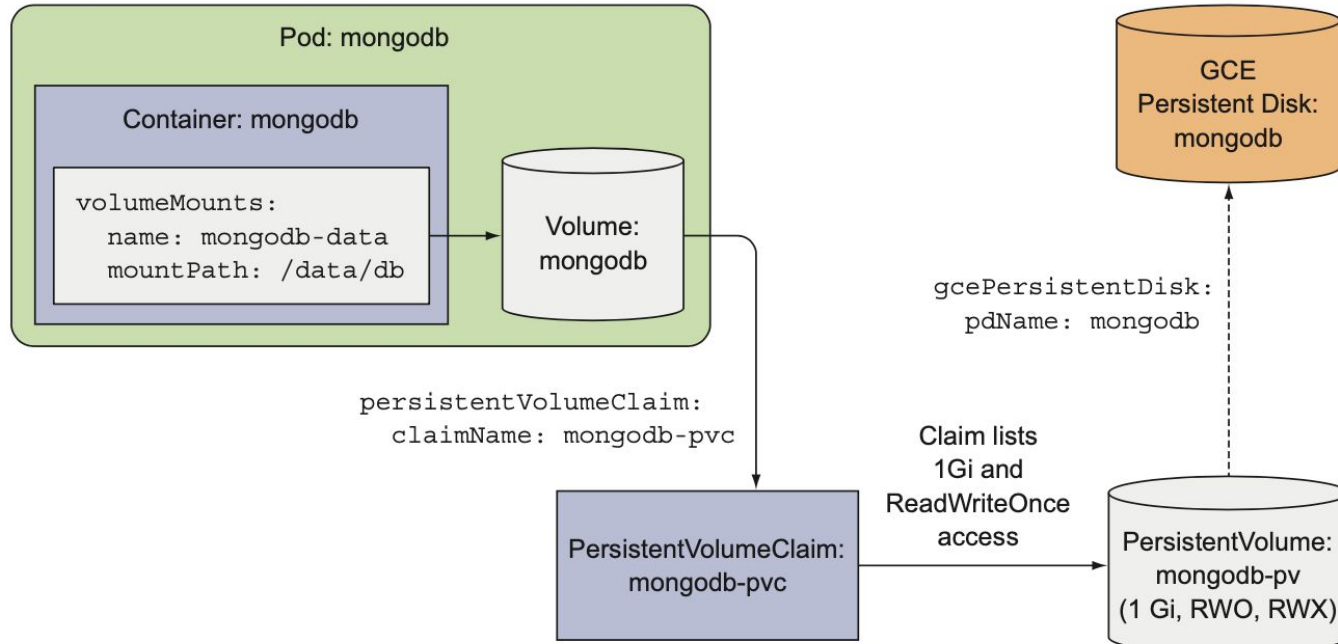
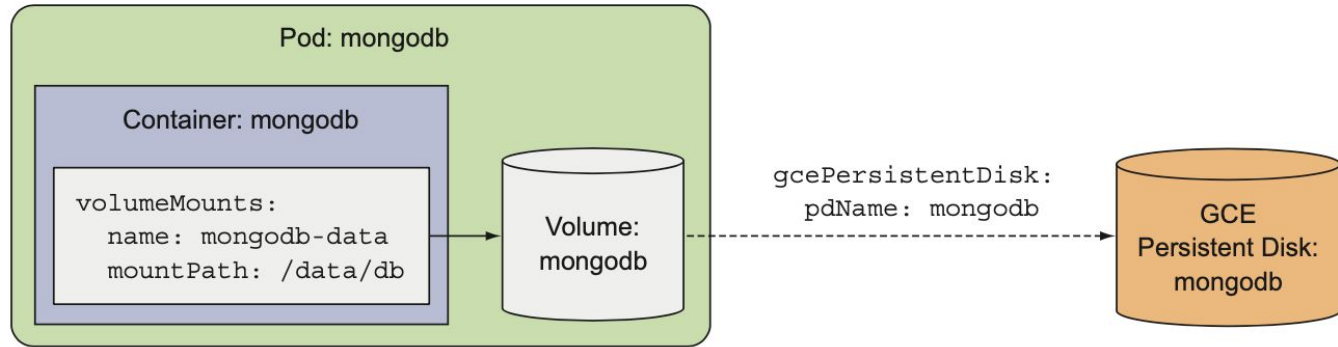
POD



PersistentVolume 와 PersistentVolumeClaim 영역 구분

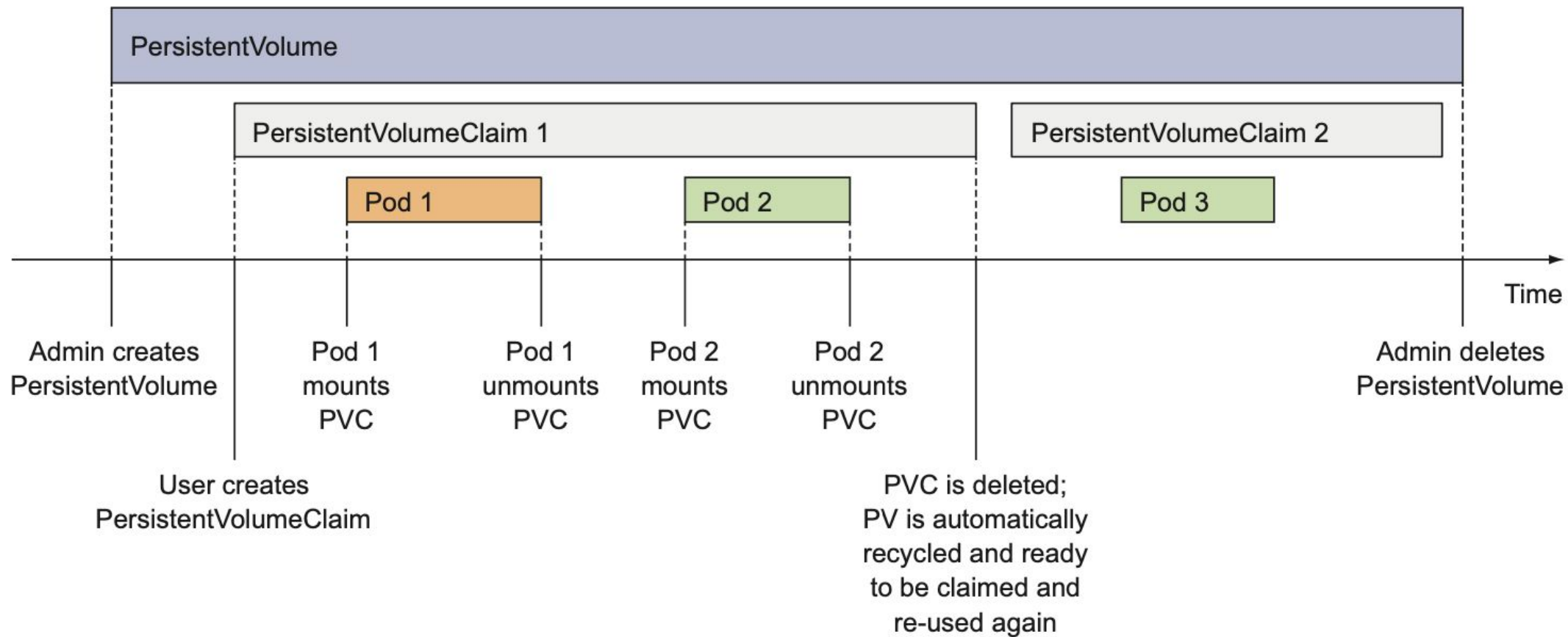


Volume 직접 생성 과 PVC 를 이용한 간접 생성 차이



개발자가 디스크나 인프라에 대해 자세히
알지 못해도 손쉽게 디스크를 사용하고 해지
할 수 있음

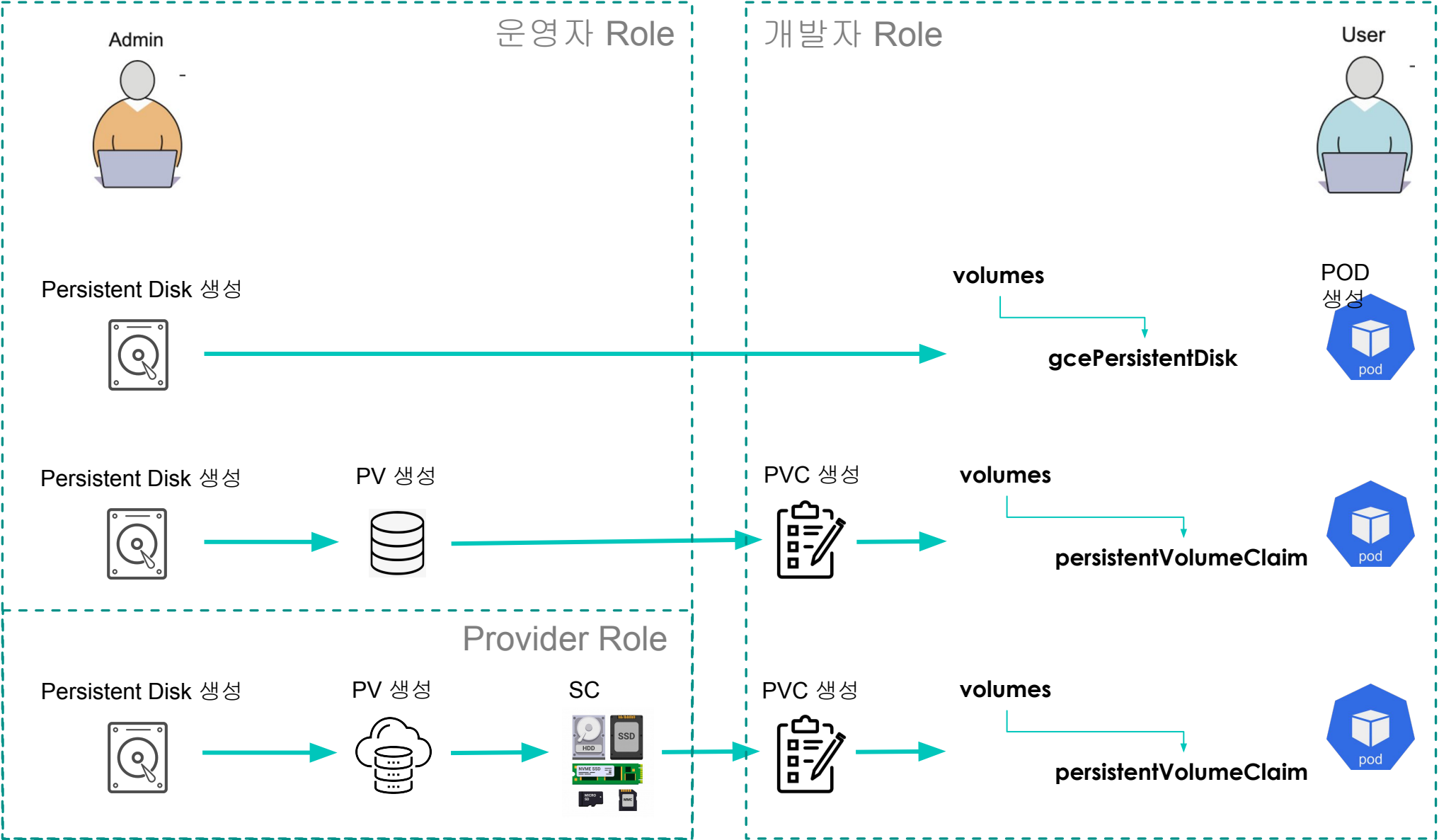
PV 와 PVC 의 수명 정리



PersistentVolume 의 동적 프로비저닝

- 사용자 : PVC 및 POD 만 생성
- 퍼시스턴트볼륨 프로비저너 : PersistentVolume 생성, Default 스토리지 클래스 생성

쿠버네티스에서 볼륨을 사용하는 방법



실습 총정리

