

iSCSI 볼륨 기초 설명



쿠버네티스 볼륨중에 iscsi 에 대해서 기초부터 설명 부탁해



좋습니다. 이제 `Kubernetes Volume` 중에서 **iSCSI(Internet Small Computer Systems Interface)** 를 정리해 드릴게요.

1. iSCSI란 무엇인가?

- **iSCSI**는 네트워크(보통 TCP/IP)를 통해 **원격 스토리지(디스크)** 를 블록 단위로 접근할 수 있게 해 주는 프로토콜이에요.
- 쉽게 말하면, **LAN/WAN** 위에서 동작하는 **SCSI 명령어 세트**.
- 서버가 물리적으로 연결된 디스크처럼 원격 스토리지를 사용할 수 있게 만들어줍니다.

→ 예를 들어, NAS(Network Attached Storage)는 파일 단위 접근이고, iSCSI는 SAN(Storage Area Network)처럼 블록 단위 접근이라 데이터베이스 같은 워크로드에 더 적합해요.

2. 쿠버네티스에서 iSCSI Volume

쿠버네티스 Pod이 **원격 스토리지를 직접 연결해서 사용할 수 있도록** 지원합니다.

- Pod → Volume → iSCSI Target (스토리지 서버)
- Pod 내부에서는 로컬 디스크처럼 보이지만, 실제 데이터는 외부 iSCSI 서버에 저장됩니다.

3. iSCSI 볼륨 특징

- **Persistent Volume(PV)** 과 연결해서 사용하는 것이 일반적입니다.
- Pod이 재시작되더라도 데이터는 **iSCSI 서버에 남아있음**.
- ReadWriteOnce 모드(= 한 번에 하나의 노드만 읽기/쓰기 가능)가 기본.
여러 Pod이 동시에 같은 볼륨을 공유하는 건 까다롭습니다.

- 사용하려면 노드들이 iSCSI 클라이언트 패키지를 설치하고, iSCSI target과 통신할 수 있어야 합니다.
(예: Ubuntu라면 `open-iscsi` 패키지 필요)

4. 간단한 YAML 예시

PersistentVolume과 PVC 예시를 보여드릴게요.

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-iscsi
spec:
  capacity:
    storage: 5Gi
  accessModes:
    - ReadWriteOnce
  iscsi:
    targetPortal: 10.0.0.50:3260 # iSCSI 서버 주소:포트
    iqn: iqn.2023-08.com.example:storage.target01 # 대상 IQN (iSCSI Qualified
Name)
    lun: 0 # Logical Unit Number
    fsType: ext4
    readOnly: false
  persistentVolumeReclaimPolicy: Retain
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-iscsi
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 5Gi
---
apiVersion: v1
kind: Pod
metadata:
```

```
name: pod-iscsi-test
spec:
  containers:
  - name: app
    image: busybox
    command: ["sleep", "3600"]
    volumeMounts:
    - mountPath: /data
      name: iscsi-vol
  volumes:
  - name: iscsi-vol
    persistentVolumeClaim:
      claimName: pvc-iscsi
```

5. 정리

- iSCSI는 네트워크 블록 스토리지.
- 쿠버네티스에서는 `PersistentVolume` 타입 중 하나로 지원.
- 데이터베이스, 고성능 스토리지가 필요한 경우 사용.
- 노드에 iSCSI 클라이언트 세팅이 선행되어야만 동작.

👉 여기까지가 "기초 개념" + "실습에 바로 쓸 수 있는 YAML" 수준이에요.

원하시면 제가 이어서 **NFS, Ceph, Local, HostPath** 같은 다른 **Volume**들과 비교해서 특징을 정리해 드릴 수도 있습니다.

혹시, 비교까지 같이 해드릴까요? 아니면 iSCSI를 더 깊게 (예: multipath, CHAP 인증, 실제 minikube에서 테스트 방법) 파고들까요?