

파일 스토리지, 블록 스토리지, 오브젝트 스토리지

개요

- 파일, 블록 및 오브젝트는 다양한 방식으로 데이터를 보유하고 구성하는 스토리지 형식
- 파일 스토리지 : 데이터를 폴더의 파일 계층 구조로 구성하고 표현
- 블록 스토리지 : 데이터를 임의로 구성해 동일한 크기의 블록으로 나눔
- 오브젝트 스토리지 : 데이터를 관리하고 연결된 메타데이터와 이를 연결



파일 스토리지 (File Storage)

- 파일 수준 또는 파일 기반 스토리지. (일반적으로 흔히 알고 있는 스토리지)
 - 파일과 폴더의 계층 구조
- 데이터가 폴더 안에 단일 정보로 저장
 - 제한된 양의 메타데이터(파일 경로 데이터)를 사용해 검색
- NAS (Network Attached Storage)

장점

- 표준화가 잘 되어 있음.

단점

- 데이터가 많아지면 성능 저하 -> 확장을 위해 scale-out 해야 함.

블록 스토리지 (Block Storage)

- 데이터를 블록 단위로 나누어 저장하는 방식
 - 각 블록은 블록이 저장된 위치를 가리키는 고유의 주소를 가지고 있어, 주소만 알고 있으면 분산 저장된 데이터를 찾아 하나의 데이터로 재구성
- 데이터를 사용자 환경에서 분리해, 분산하도록 저장하는 경우가 많다
- SAN (Storage Area Networking)

장점

- 파일 스토리지와 다르게 단일 데이터 경로에 의존하지 않으므로 신속하게 검색 가능.
- 각 블록은 독립적 / 파티션 분할 가능 -> 다른 운영체제에서 접근 가능

단점

- 비용이 많이 들 수 있다. 메타데이터를 처리하는 기능이 제한적

오브젝트 스토리지 (Object Storage)

- 오브젝트 기반 스토리지
 - 구조화되지 않은 데이터의 대량 저장을 위한 데이터 스토리지 아키텍처
 - 비정형 데이터를 담기에 최적의 스토리지 중 하나
- 파일들이 작게 나뉘어 여러 하드웨어에 분산되는 평면 구조
 - 블록이나 폴더에 저장하는 대신 단일 repo에 보관
- 각각 독립적인 repo이며 메타데이터를 보유함.
 - 상세한 메타데이터와 식별자를 상용하여 데이터 검색
- 오브젝트 스토리지 내 오브젝트는 API를 통해 접근 가능.

- 사용한 만큼 비용 지불
- 확장성 좋음

참고 하면 좋은 글

<https://velog.io/@leesjpr/스토리지File-storage-Block-storage-Object-storage>