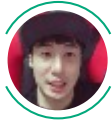


클라우드상 오브젝트 스토리지(Object Storage)란?



Harry The Great

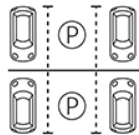
Follow

Dec 12, 2018 · 6 min read

AWS에서 DynamoDB를 사용할때 S3를 사용할때 혹은 Firebase의 Storage, Firestore등을 사용하면 항상 수정은 할 수 없습니다. SDK상에 Update라는 메서드가 있지만 사실상 파일을 덮어쓰고(지우고 다시 넣기) 변경을 처리합니다. 우리가 흔히 써왔던 온프로미스 환경과는 조금 다른점이라 찾아보면 객체 스토리지(Object Storage)의 특성이라고 말합니다. 그래서 찾아보니 생각만큼 이해하기 쉬운 자료는 나오지 않아 직접 정리한 내용을 공유하고싶었습니다.

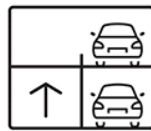
객체스토리지가 조금 더 옳은 표현이라하며 오브젝트 스토리지를 이하 객체스토리지로 기술하였습니다.

우선 객체스토리지를 알기 위해서는 3가지를 알아야합니다. 블록스토리지(Block Storage), 파일스토리지(File Storage), 객체스토리지(Object Storage)



블록 스토리지

'주차장'에 비유 - 엄격하게 정의된 블록에 데이터가 저장됨 - 특정 '공간' 위치를 통한 액세스



파일 스토리지

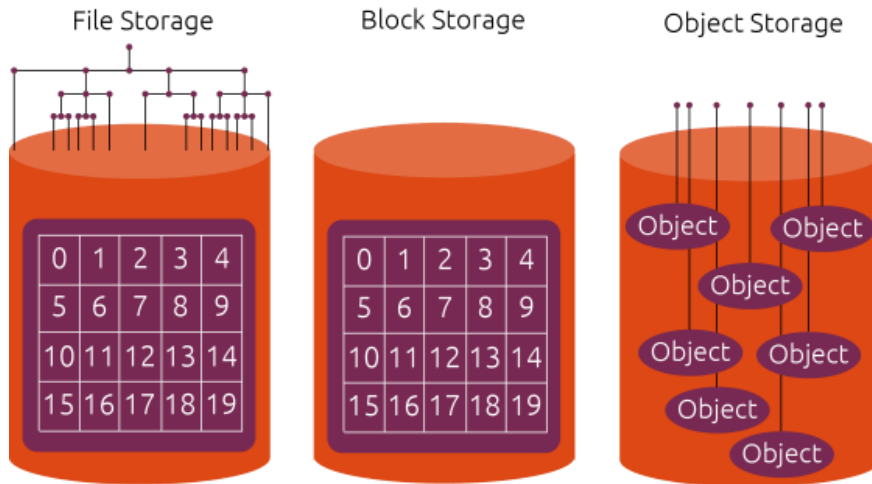
'주차장 건물'에 비유 - 데이터가 계층형 레벨로 정리됨 - 경로를 역추적해서 액세스



오브젝트 스토리지

대리 주차'에 비유 - 스토리지 세부 정보에 대해 걱정할 필요 없음 - 데이터 저장 및 액세스가 용이함

출처) Dell EMC



(역시 설명에는 그림이 있어야...)

파일스토리지

폴더와 파일로 이루어지는 계층구조를 갖는 스토리지입니다. 각 파일은 폴더에 종속되며 폴더 역시 다른 폴더에 종속될 수 있습니다. 캐비닛을 떠올리면 이해하기 쉽습니다. 각각의 파일철들이 캐비닛별로 저장되어있고 언제든 꺼내어 수정하거나 변경할 수 있습니다. 파일철을 찾으려면 어느 캐비닛이 있는지 알고있어야합니다. 마찬가지로 파일을 찾으려면 어느 경로에 있는지 알고있어야합니다. 파일철이 그리 많지 않다면 분류하고 정리하는데 큰 문제가없지만 파일철들이 계속해서 늘어나면 늘어날수록 점점 분류하고 정리하는(파일시스템에서의 인덱싱)이 많아지고 찾기가 힘들어집니다.

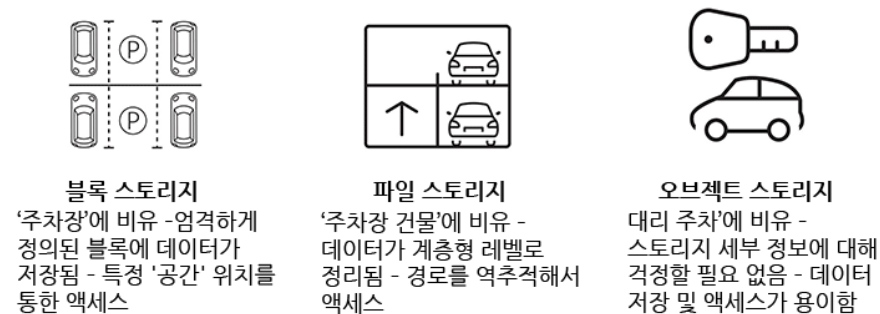
블록스토리지

블록스토리지는 정해진 블록안에 데이터를 저장합니다. 우리가 SQL에서 테이블을 만들때 각 칼럼별로 저장할 데이터를 설정해주고 이후의 데이터 삽입도 맨처음 설정한 값 범위안에서 저장해주는것과 유사합니다. 블록스토리지는 컴퓨터의 C드라이브, D드라이브와 유사합니다. 맨처음 파티션을 나누어주고 그 공간안에서 사용을 할 수 있습니다. 하지만 파일스토리지와 다른점은 파일스토리지는 1개의 경로만 갖는데 반하여 블록스토리지는 여러개의 경로를 가질 수 있습니다. C드라이브와 D드라이브를 네트워크를 통해서 공유하고 여러 사용자가 해당 드라이브를 참조하는것과 유사합니다. (물론 공유는 할 수 있지만 OS와의 연결은 한번에 1개만 가능합니다.) 낮은 IO 레이턴시를 보이기때문에 RDB와 같은 데이터베이스에 아주 적합합니다.

오브젝트 스토리지

그래도 이해가 잘 안간다면...?

블록스토리지와 파일스토리지는 모두 OS단에서 동작하지만 오브젝트 스토리지는 어플리케이션 단에서 동작합니다. 쉽게 말하면 S3나 Cloud Sotrage에서 폴더를 만들거나 다른 버킷으로 파일을 옮긴다고하여도 실제로 블록을 이동하거나 폴더에 종속된게 아닌 단지 사용자에게 그렇게 보이게 해주기만 해줍니다. 즉 논리적인 스토리지라고 볼 수 있습니다. 위의 이미지에서 보듯이 오브젝트 스토리지는 물리적 제약이 없기 때문에 원하는 만큼 공간을 확장시킬 수 있습니다. 저는 이부분에서 많이 헷갈리다가 한가지 좋은 사례를 보고 조금 더 이해할 수 있던 이미지가 있었습니다.



출처) Dell EMC

블록 스토리지는 주차장과 같습니다. 주차장이 꽉차면 더이상 주차할 수 없습니다. 그렇다면 필요한만큼 주차장을 늘려놓아야합니다. 파일 스토리지는 주차타워와 같습니다. 문제는 내 차를 타기 위해서는 주차 타워가 한번 돌아가야합니다. 즉 주차가 많아지면 많아질수록 폴더가 깊어지고 많아질수록 차를 찾기가 힘듭니다. 오브젝트 스토리지는 발렛아저씨들이 알아서 주차를 해줍니다. 강남이나 판교에서 발렛 맡겨 보시면 아시겠지만 이 아저씨들은 정말 공간의 예술을 보여줍니다. 어떠한 공간도 정말 효율적이고 예술적으로 사용하고 공간의 낭비가 하나도 없도록 주차를 해줍니다.

클라우드에서 오브젝트 스토리지를 쓰는 이유는?

역시 확장성과 속도 그리고 저렴한 가격입니다. 오브젝트 스토리지는 오로지 2가지 키값과 데이터만 저장합니다. 발렛을 맡길때도 자동차(데이터) 세워두고 번호표만 줍니다. 어디에 주차되어있는지 알필요는 없습니다. 마찬가지로 오브젝트 스토리지는 RESTful Protocol(HTTP)를 이용하여 Get 혹은 Post로 요청을하면 파일을 내려줍니다. 그 이상은 알 필요가 없습니다. 또한 파일에 대해 가지고 있는 정보가 적기때문에 파일이 아무리 많아져도 블록스토리지나 파일스토리지에 비해서 빠르게 작동합니다. 마지막으로 공간을 아주 효율적으로 사용할 수 있기때문에 저렴합니다.

물론 단점도 존재합니다. 파일의 수정이 불가능합니다. 파일이 수정될 때 트랜잭션을 통해 일관성을 유지하기가 힘들기때문에 덮어쓰는 방법을 이용합니다. 이렇기 때문에 이미지나 동영상같이 수정이 잘 일어나지 않는 정적인 데이터를 호스팅할때 좋습니다. 또한 내구성이 블록스��리지에 비해 떨어지기때문에 내구성을 상당히 요야하는 데이터를 처리하기 힘듭니다.

마치며

점점 데이터의 종류와 양이 많아지고 비용의 부담이 커질수록 많은 클라우드와 데이터센터들이 점점 오브젝트 스토리지로 넘어가고있습니다. 또한 점점 많이 용어가 사용되고있기때문에 누군가 '오브젝트 스토리지가 뭐야?' 라고 말했을때 잘난척 한번은 할 수 있을것 같습니다.

참고 및 인용

<https://korea.emc.com/corporate/glossary/object-storage.htm>

Object Storage vs. Block Storage Services | DigitalOcean

Flexible and scalable data storage is a baseline requirement for most applications and services...

www.digitalocean.com



<https://www.redhat.com/ko/topics/data-storage/file-block-object-storage>

