

Storage Service

Amazon S3: 데이터 저장 장소

1. 객체 스토리지

- Amazon Simple Storage Service(S3): AWS에서 제공하는 객체 스토리지 서비스
 - 스토리지: 데이터 저장 장소
 - 객체: 텍스트 파일, 음성 파일 등의 데이터
- 객체 스토리지
 - 폴더 구조X, **객체 키로 데이터를 고유하게 식별** → 데이터의 입출력/관리 수행
 - 키만으로 데이터를 관리하므로 **간편하게 대용량 데이터 저장/관리 가능**
- S3
 - AWS에서 시스템을 구축할 때 유용하게 사용 가능
 - 객체 키만으로 관리되므로 대량 로그 보존에도 이용 가능

2. S3의 특징

- **용량 무제한**: 객체 당 5TB라는 제약, 객체 수/데이터 용량에는 제한X
- **높은 내구성**: 일반적으로 데이터가 3개 이상의 AZ로 복사, 복사를 통해 내구성↑
- **저렴한 비용**: 사용한 만큼만 발생, 1GB/월 당 약 30원

3. 다른 서비스와 연계

- **확장성, 내구성, 저렴한 비용이라는 장점** → AWS의 다른 서비스와 연계 사용 다수

4. 요금

- 데이터 저장 용량, 데이터 액세스 횟수, 데이터 전송량에 따라 요금 발생

품목	요금(서울 리전)
데이터 저장	0.025USD/GB
데이터 액세스(업로드)	0.0045USD/1,000개 당
데이터 액세스(다운로드)	0.00035USD/1,000개 당
데이터 전송(S3 → 인터넷)	0.126USD/GB

- 데이터 전송 요금이 저장 요금보다 높을 수 있으므로 대량의 데이터 다운로드 시 주의

S3의 기본 용어 및 기본 조작

1. S3에서 사용되는 중요 용어

- 버킷
 - 객체(데이터)를 저장하는 장소, 사용자는 버킷에 객체 저장
 - **버킷의 이름은 전 세계에서 고유해야 함(모든 리전)**
- 객체
 - 버킷에 저장된 데이터 본체
 - 객체 무제한 저장 가능, but 한 객체의 최대 크기 5TB
- 키
 - 객체의 저장 URL 경로, 버킷 이름과 키 이름, 객체 이름을 조합해 고유하게 설정
- S3은 객체 스토리지로 폴더 구조가 아니지만 '/'라는 문자열을 구분자로 판단해 관리
 - 콘솔에서는 폴더 구조처럼 객체 표시

2. S3의 데이터를 브라우저나 프로그램에서 조작

- **데이터 등록/삭제 API(Application Programming Interface) 제공** → 이를 호출해 객체 조작
- API → 조작을 받아들이는 창구와 같은 역할

GET	S3에서 데이터(객체) 다운로드
PUT	- S3에 객체 업로드, 신규 업로드, 갱신(덮어쓰기) - 하나의 PUT 작업은 최대 5GB까지 이므로 더 많은 데이터 업로드 시 멀티 파트 업로드 라는 기능을 활용해 여러 파트로 나눠 최대 5TB의 데이터 업로드 가능
LIST	S3 버킷의 객체 목록 표시, 데이터 추출 가능
COPY	S3 내에서 객체 복사, 다른 버킷과 리전 간 복사 가능
DELETE	S3 내에 있는 임의의 객체 삭제

- 관리 콘솔은 직관적으로 구성되어 쉽게 데이터 다운로드/업로드 가능

S3 데이터를 적절하게 저장

1. 사용하지 않는 객체 저렴하게 저장

- S3는 비용과 가용성이 다른 여러 스토리지 클래스 제공
- 비용 효율성을 중시한 아카이브용 스토리지인 Amazon S3 Glacier 도 존재
- **가용성: S3에 데이터를 업로드/다운로드 등 사용할 수 있는가 하는 관점의 수치**
- 거의 사용하지 X, 보존하고 싶은 데이터 → S3 Glacier Deep Archive
언제든지 사용할 수 있어야 하는 데이터 → 표준
- **수명 주기 설정**
 - 자동으로 스토리지 클래스 변경 가능
 - but 저비용으로 마이그레이션 시 표준 클래스로 돌아갈 수 없음
- S3용 액세스 분석기라는 기능으로 데이터 용량과 접근 상태 확인 후 스토리지 클래스 결정

2. 버전 관리 및 다른 리전으로 복사

- S3의 버전 관리 기능 활성화 → 모든 세대의 객체에 대한 이력 정보 보존
- 객체 갱신 시(PUT) 기존 객체와 다른 새로운 버전 ID 부여
- 버전 관리 설정은 S3 버킷 단위로 가능, 객체 단위 설정은 불가

- 삭제(DELETE) 시 객체가 실제로 삭제되는 것이 아닌 삭제 마커 부여, 객체의 과거 버전 유지
→ 버전 ID 지정 삭제 시 지정한 버전 삭제 가능
- 각 버전은 별개의 객체로 보존되므로 추가 요금 발생 → 수명 주기를 사용해 자동 삭제 가능
- **교차 리전 복제(CRR-Cross Region Replication):** 데이터 저장 위치와 다른 리전에 복사하는 기능
- **재해복구(DR-Disaster Recovery):** 하나의 리전 전체가 재해를 당한 경우를 가정해 설계

S3 데이터를 안전하게 공개

1. S3 버킷에 대한 접근 제어

- 허가받은 사람이 허가받은 객체에 올바르게 접근할 수 있도록 접근 제어를 하는 것이 중요
- **사용자 정책(IAM 정책)**
 - 데이터를 조작하는 쪽을 제어하는 설정
- **접근 제어 목록(ACL)**
 - 객체(데이터) 또는 버킷 단위로 설정 가능한 접근 제어
 - 데이터별로 상세한 설정 가능
- **버킷 설정**
 - 버킷 단위로 설정하는 접근 제어
 - 복잡한 설정 가능, 설정은 JSON 형식으로 기재
- S3에는 퍼블릭 액세스 차단이라는 기능이 있어 의도치 않게 공개 상태가 되는 것을 방지

2. S3를 이용해 웹 사이트 구축

- S3 버킷에 저장된 HTML, CSS, 자바 스크립트 등의 콘텐츠 파일을 이용해 웹사이트 구축 가능
- S3는 암호화된 통신인 HTTPS를 지원하지 않으므로, 민감한 정보가 포함된 사이트를 운영해야 하는 경우 콘텐츠 전송 네트워크(CDN) 서비스인 Amazon CloudFront를 함께 사용해 HTTPS 통신을 할 수 있게 구성

3. 데이터 암호화

- 데이터 암호화 실시 → 데이터와 키 데이터를 조합해 데이터 내용을 타인이 볼 수 없게 함
- 키 데이터가 없으면 데이터를 읽을 수 없음
키 데이터 유출 시 타인이 데이터를 읽을 수 있음 → 키를 안전하게 관리해 데이터 보호
- S3에서는 버킷의 기본 암호화 기능 활성화 시 객체 저장할 때 AWS에서 자동으로 암호화 수행
- 암호화용 키: **AWS Key Management Service(KMS)**에서 생성한 키, 사용자가 관리하는 키
 - KMS 키 사용시 AWS 시스템 내에서 자동으로 암호화 처리 수행 → 사용자는 객체 저장만
- S3에서 제공되는 기능은 객체 저장 시 암호화이므로 S3로 전송 시 암호화하고 싶다면, 미리 데이터를 응용 프로그램에서 암호화해 전송 → 클라이언트 측 암호화
 - 사용자 고유키, KMS 키 사용 가능
- S3의 API에 대한 접근은 기본적으로 HTTPS에서 수행되므로 통신은 모두 암호화

가상 서버 데이터 저장을 위한 EBS

1. 서버 데이터를 저장하는 EBS

- Amazon Elastic Block Store(EBS)
 - EC2와 함께 사용하는 스토리지 서비스
 - EC2의 HDD/SSD와 같은 역할

- EC2에서 실행되는 응용 프로그램의 데이터/로그/설정 정보 등을 저장하는 데 주로 사용
- EBS에서는 필요한 속도에 따라 볼륨 유형 선택 가능
 - EBS의 쓰기/읽기 성능은 **IOPS(Input/Output Per Second)**라는 초당 쓰기/읽기 횟수를 표현하는 단위 사용

볼륨 유형	개요
범용 SSD (gp2, gp3)	- 균형 잡힌 범용 타입, 일반적 용도 사용
프로비저닝된 IOPS SSD (io1, io2, io2 Block Express)	- 고성능 스토리지 필요 시 사용 - 필요한 IOPS를 사용자 지정 가능
처리량 최적화 HDD (st1)	- 저비용 마그네틱 스토리지
Cold HDD(sc1)	- st1보다 훨씬 저렴한 비용의 마그네틱 스토리지 - 접근 빈도가 낮을 때 사용

2. 서버 백업 및 이미지 관리

- 데이터 파손 등의 사고 방지를 위해 **스냅샷**이라는 **백업 기능** 지원
- 스냅샷에는 스냅샷을 만들 당시 EBS에 보존하고 있던 모든 정보 백업
- 스냅샷은 EBS 볼륨으로 복구해 EC2에 다시 연결해 사용할 수 있고 스냅샷과 몇 가지 설정 정보를 조합해 사용자 전용 AMI 생성 가능 → **AMI는 가상 서버 생성에 사용**

3. EBS 요금 예

- EC2 인스턴스는 정지 중에는 요금 미발생, EBS는 용량 확보 후부터 계속 요금 발생
→ EC2가 중지되어 있어도 데이터를 저장하고 있는 EBS에는 계속 요금 발생
- 데이터 보존량뿐만 아니라 IOPS나 데이터 처리 용량에도 요금이 발생하지만 데이터 보존에 비해 저렴 but, 읽기/쓰기 대량 발생 시 주의 필요

데이터 공유, 백업, 전송을 위한 서비스

1. Amazon Elastic File System (EFS)

- 비교적 고속으로 데이터를 전송할 수 있는 NFS(Network File System)라는 프로토콜을 이용해 여러 EC2 인스턴스가 함께 이용할 수 있는 스토리지
- ACL 설정만 잘 돼 있다면 온프레미스 서버나 로컬 PC에서도 마운트해서 사용 가능

2. Amazon FSx

- 파일 서버 구축을 위한 서비스
- Amazon FSx for Windows File Server: 윈도우에서 사용되는 파일 공유 프로토콜인 SMB 이용

Amazon FSx for Lustre: 대규모 클러스터 컴퓨팅, 슈퍼컴퓨터 등에서 사용되는 Lustre라는 고성능 파일 시스템 이용

- AWS Storage Gateway
 - 온프레미스에 서버 기기 혹은 가상 서버에 설치해 온프레미스 기기와 S3, FSx, EBS를 직접 연결하는 서비스
 - 직접 AWS 서비스와 데이터를 교환하는 것보다 빠름

3. AWS Transfer Family

- S3, EFS, HTTP, NFS가 아닌 SFTP, FTPS, FTP와 같은 FTP 기반 프로토콜로 통신하기 위한 서버 구축 서비스
- FTP를 이용하는 시스템을 AWS로 마이그레이션할 때 응용 프로그램의 FTP를 사용하는 부분은 그대로 두고 전송 대상을 가용성 높은 S3 또는 EFS로 변경 가능

4. AWS Backup

- EBS, EFS, FSx 등 스토리지, rds나 DynamoDB 등 데이터베이스의 데이터 백업 서비스
- 백업 규칙 설정 시 백업이 자동 실시
- 백업은 S3에 저장되므로 가용성과 내구성이 매우 높음

5. AWS DataSync

- 온프레미스와 AWS 혹은 AWS 스토리지 서비스 간 데이터 전송을 위한 서비스
- 온프레미스 서버에 DataSync라는 에이전트를 설치해 스토리지 서비스와 간단히 데이터 통신 가능
- 안전하고 빠른 데이터 전송 가능

6. AWS Snow Family

- 데이터 전송을 최대한 빨리 하기 위한 서비스
- 물리 스토리지를 AWS에서 빌려 거기에 데이터를 저장하고 AWS로 반환하면 해당 데이터를 직접 AWS 내의 스토리지에 옮겨주는 서비스