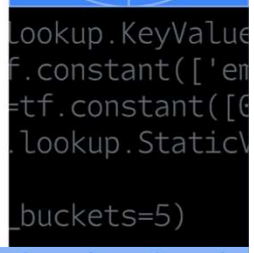
Google Developer Groups

GDG Cloud Team: Google Cloud Associate

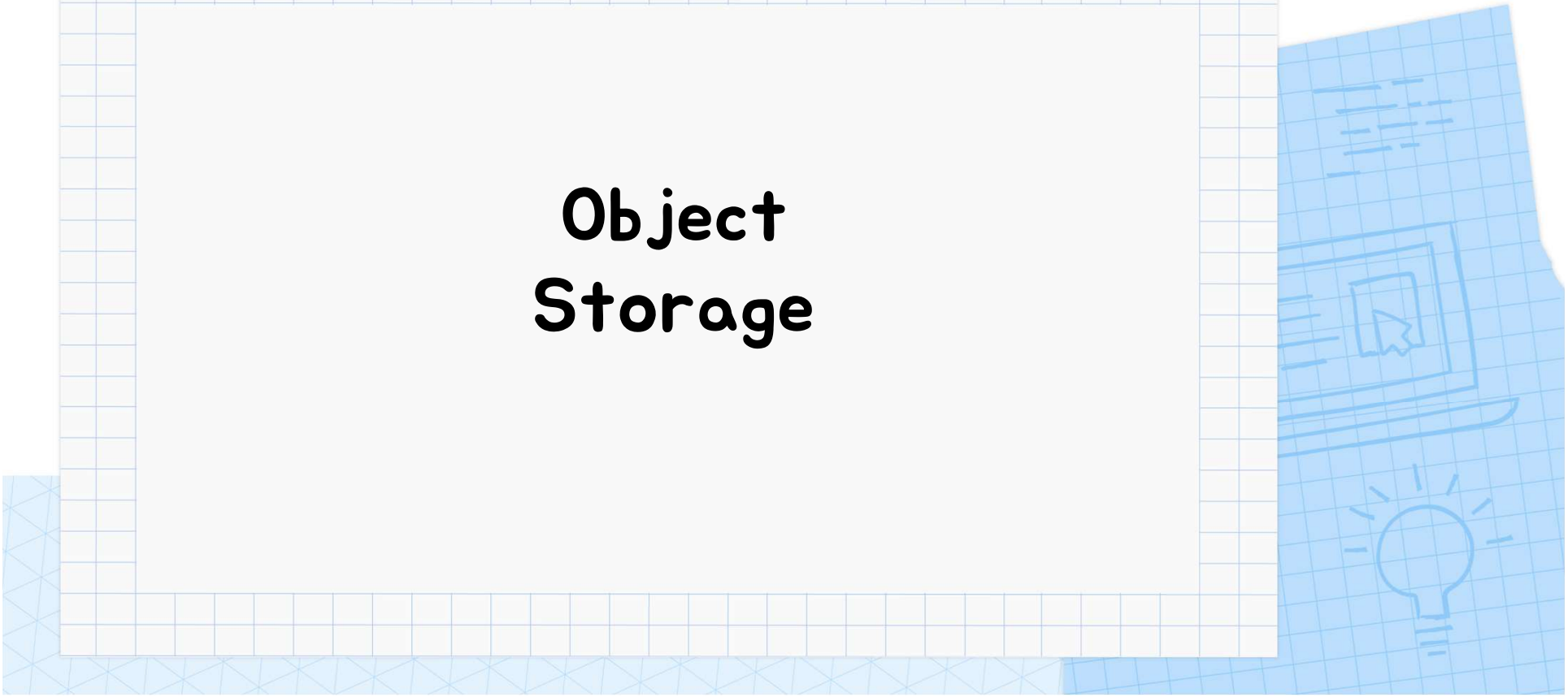


2025/01/13

Hyeongjun Kang

Google Developer Groups On Campus

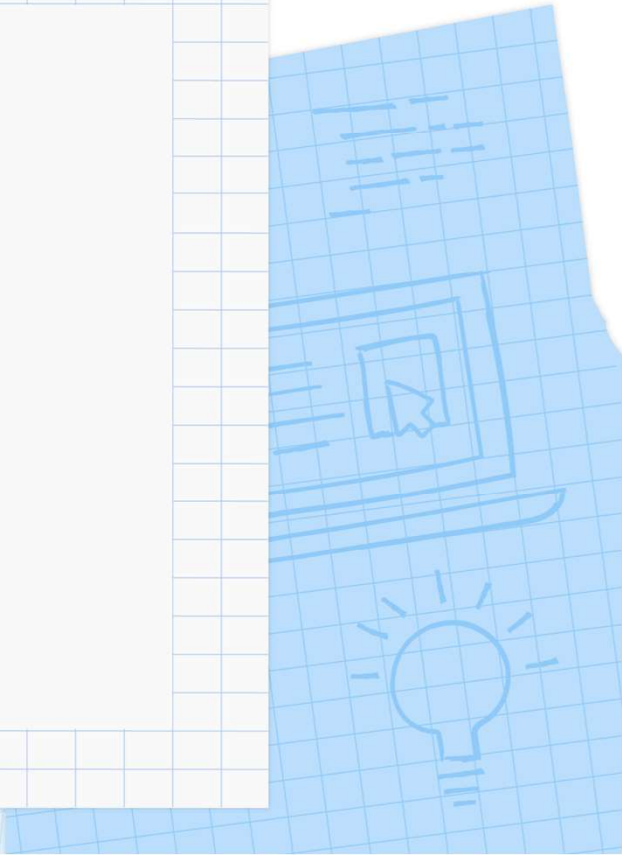
Object Storage



Google Developer Groups On Campus

Cloud storage 특징

- key-value로 접근
- 서버리스, 오토스케일링으로 거의 무한대
- 부분 업데이트 불가 (객체 자체를 변경하는 것만 가능)
- 오브젝트 단위로 접근 가능해서 object storage 라고도 부름
- Rest API 가능
- CLI 가능 근데 gcloud 아니고 gsutil임.
- 모든 파일형식 다 됨 백업 아카이브, 텍스트, 바이너리 등
- 지연율 낮고 튼튼함.



Google Developer Groups On Campus

objects and buckets

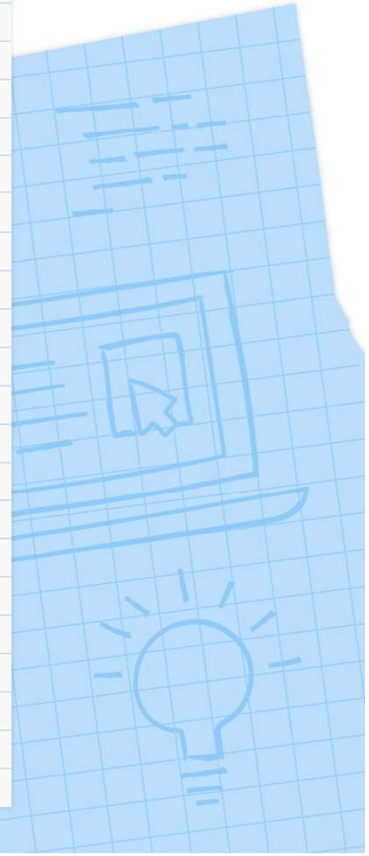
버켓 이름은 url의 일부로 활용되기에 글자 제한 지켜야 하고 중복 안됨.
고유 키가 있고 한개의 최대 오브젝트 사이즈는 5TB, 최소사이즈 없음.

기간별 설정
가장 짧은기간 자주 변경되면 standard

달에 한번꼴이면 nearline

분기에 한번꼴이면 coldline

년에 한번꼴이면 archive



Google Developer Groups On Campus

업로드와 다운로드 방법

업로드/다운로드	설명
simple upload	작은 파일들, metadata 없음
multipart upload	작은 파일들, metadata 있음
resumable upload	큰 파일들, 대부분 권장됨, HTTP 요청 가짐, 오류 시 재요청
streaming transfers	사이즈 모르겠는거
parallel composite uploads	32개로 chunking해 병렬 업로드, 네트워크와 디스크 좋으면 빠름
simple download	간단한 내용들
streaming download	크기 모름
sliced object download	나누어 슬라이스 해 다운로드

Google Developer Groups On Campus

Version

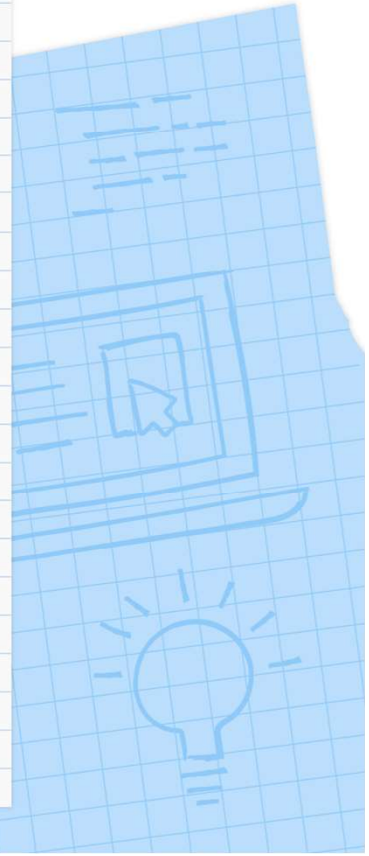
- 삭제 방지, 히스토리 제공이라는 이점

- 버킷 레벨에서 가능하고 온오프 가능

- 가장 최신 버전이 live version, live version을 삭제하면 live version에서 내려오고, live version이 아닌 걸 삭제하면 완전히 삭제됨

- 개체번호+버젼번호로 구별됨

- 옛날 버젼 삭제하면 비용 절감



Google Developer Groups On Campus

라이프 사이클

- 자동으로 저장소 들어있는 것의 age나 CreatedBefore, 라이브여부, 클래스, 가장 최신버젼 숫자 등으로 스토리지의 class를 바꾸거나 삭제하는 행동을 자동화할 수 있다.

- 옮기는 건 왼쪽에서 오른쪽으로 일방향으로만 가능, 한번에 여러칸 뛰어 넘는 건 가능

->	->	->	->
standard multi-regional regional	nearline	coldline	archieve

Google Developer Groups On Campus

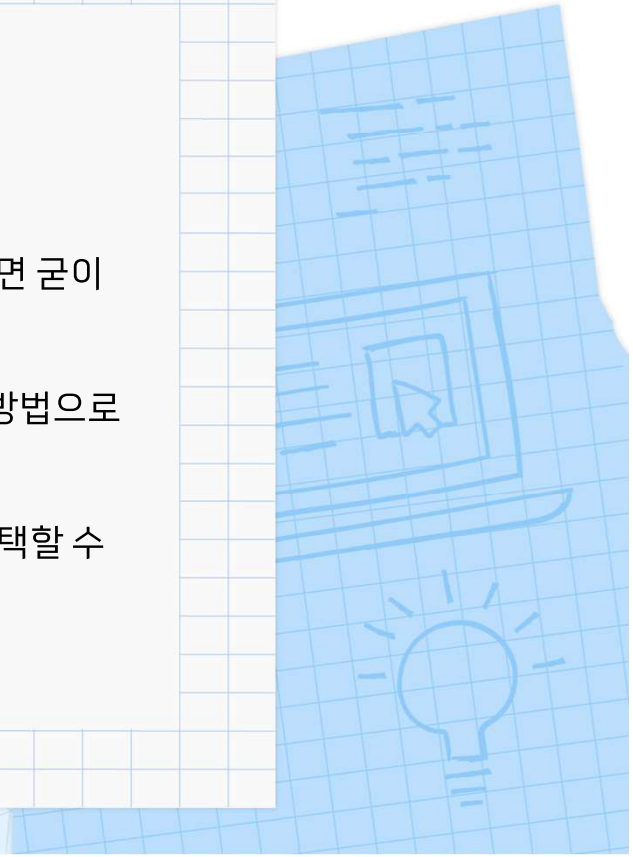
암호화

- cloud storage는 항상 서버단에서 데이터를 암호화 함

- 그 암호화 구성을 google-managed로 구글에서 하는 기본값으로 한다면 굳이 별도 구성 필요 없음

- customer-managed로 한다면 KMS를 이용하여 권한을 얻고 별도의 방법으로 암호화 할 수 있음

- option으로 customer가 upload 하기전에 client단에서 암호화할지 선택할 수 있음. 이경우 클라이언트가 전송하는 중에도 암호화 유지되는 장점 있음



Google Developer Groups On Campus

시나리오

Q: 100GB 넘는 큰 사이즈를 cloud storage에 더 빠르게 업로드 하고 싶어요 A: Parallel composite uploads 사용하세요

Q: 영구적으로 저장하며 접근 안할 것 같음
A: archive 쓰세요

Q: 분기에 한번 정도 접근할 것 같아요
A: Cold Line 쓰세요

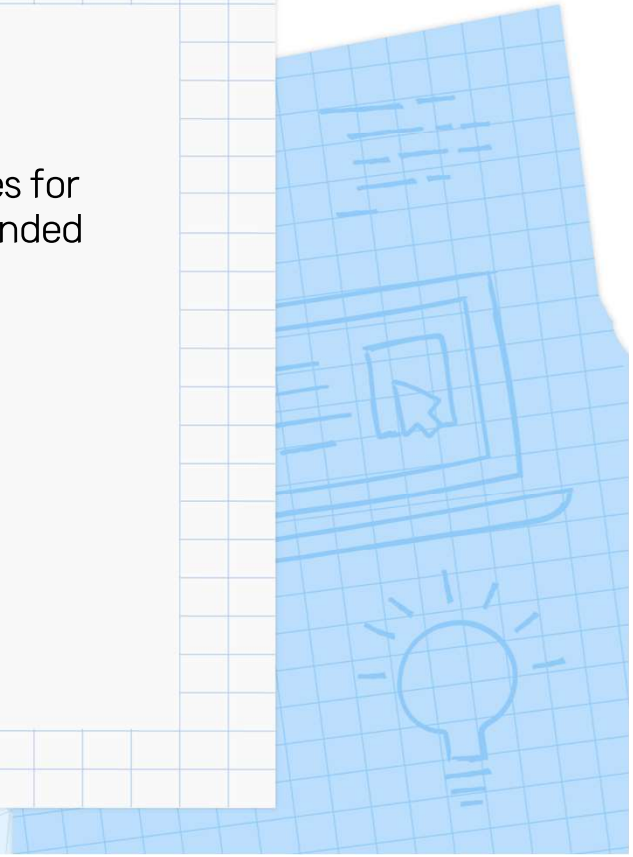
Q: 존재하는 bucket 의 storage class를 바꾸고 싶어요
A: 1.bucket의 기본 storage class를 바꾸고, 2. 오브젝트가 들어있는 버킷을 업데이트 하세요

Google Developer Groups On Campus

Quiz!

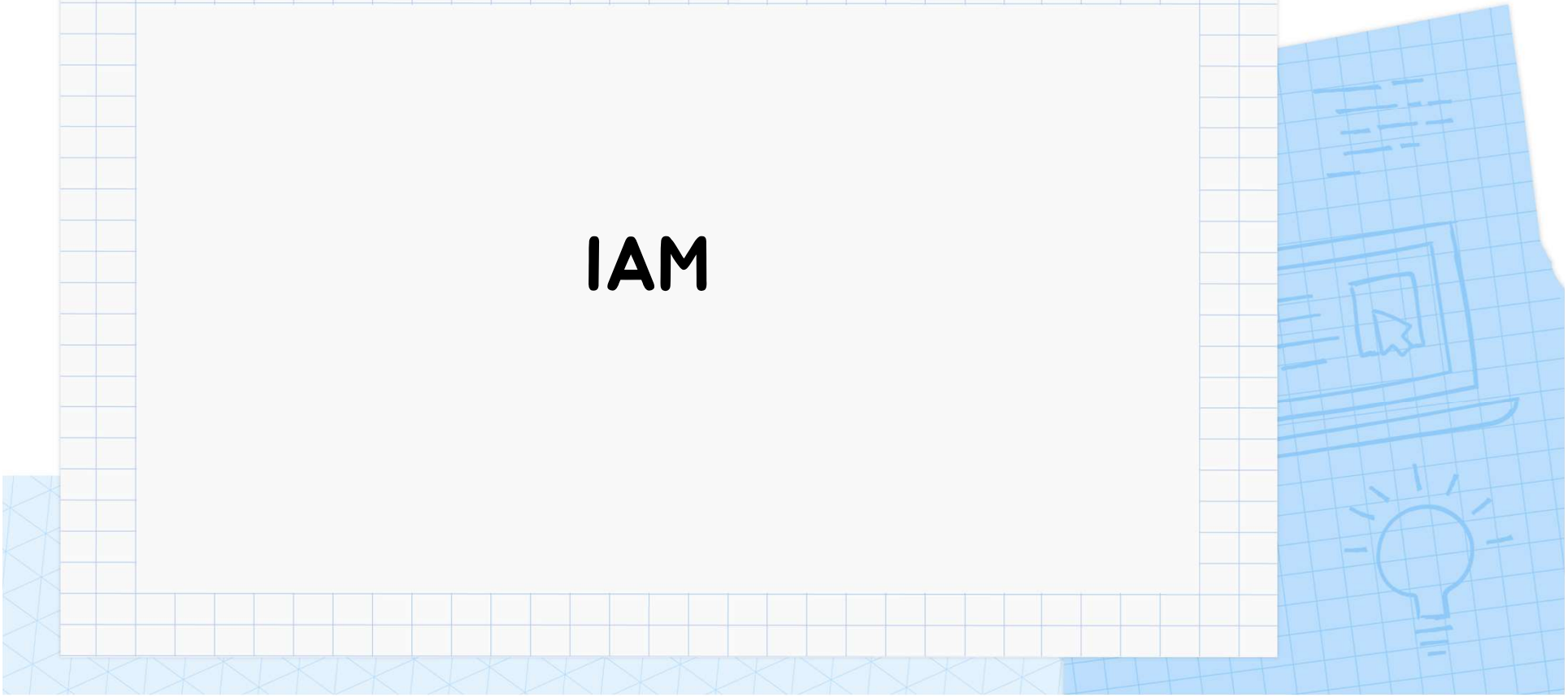
Your company uses Cloud Storage to store application backup files for disaster recovery purposes. You want to follow Google's recommended practices. Which storage option should you use?

A. Multi-Regional Storage
B. Regional Storage
C. Nearline Storage
D. Coldline Storage



Google Developer Groups On Campus

IAM



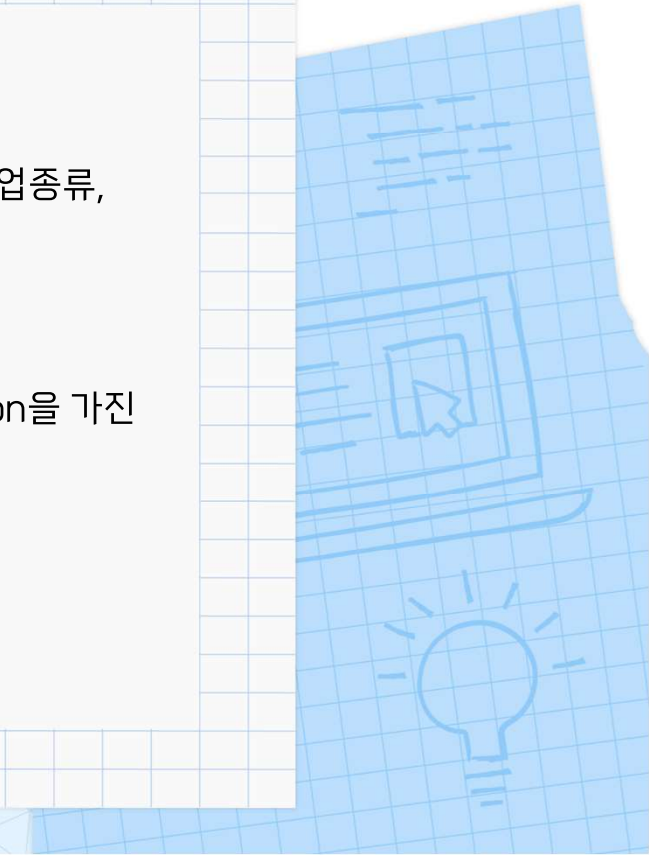
Google Developer Groups On Campus

IAM 이란?

- 사용자나 프로그램의 authentication 과 authorization를 제어함. 작업종류, 시간대, IP등을 제한할 수 있음.

- Role(permission)에 따른 policy(mapping) 함으로 써 결정됨

- 즉, 멤버 하나하나에 바로 permission이 부여되는게 아니라, permission을 가진 role을 만들고 그 role을 각 멤버에게 부여함으로써 생김



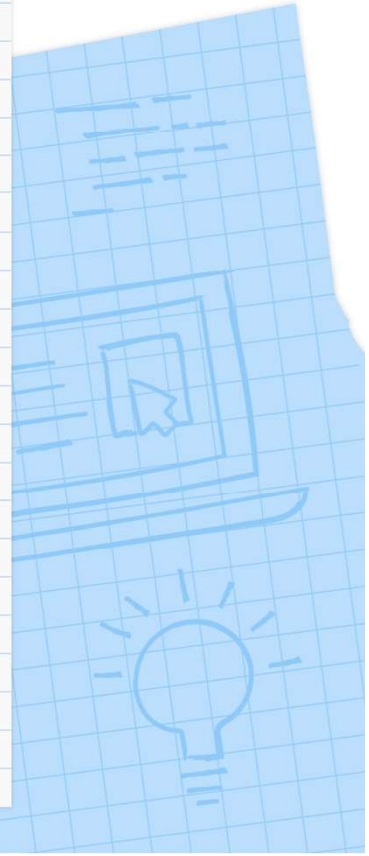
Google Developer Groups On Campus

Roles

Role은 permissions이다,

기본적인 역할의 종류
1. Basic roles
1-1) viewer: read
1-2) editor: read+ edit
1-3) owner: read+ edit+ manage+ billing

2. custom roles: 자기가 새롭게 만든 역할



Google Developer Groups On Campus

Roles

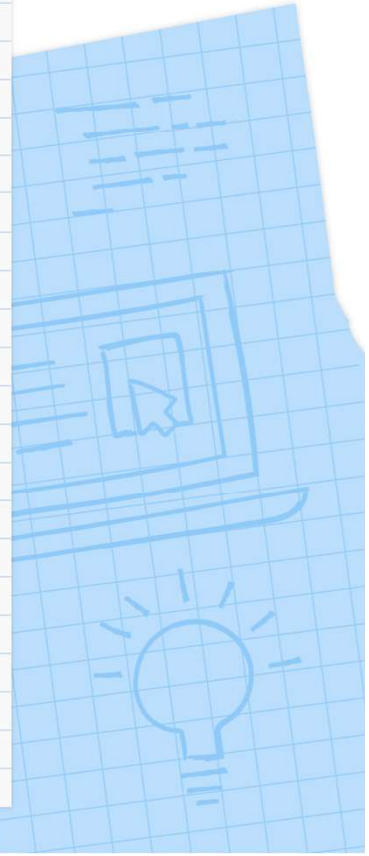
3. predefined roles
- 구글에 의해 미리 정의된 역할로 목적에 따라 다르다.
- Cloud Storage role에는 4가지가 있고 기본적으로 resourcemanager.progects.get/list에 대한 권한을 가진다.

3-1. Storage admin: Storage.buckets와 storage.objects에 대한 모든 권한

3-2. Storage object admin :storage.objects에 대한 모든 권한

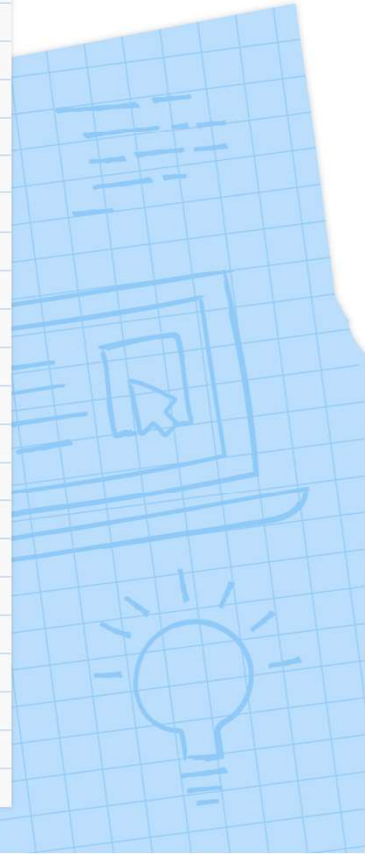
3-3. storage object creator: storage.objects.create 권한

3-4. storage object viewer: storage.objects.get/list 권한



활용

- gcloud projects 나 gcloud iam 으로 명령어가 전개된다.
  - gcloud projects로 시작하는 것들
    - `gcloud projects add-iam-policy-binding` :IAM policy binding
    - `gcloud projects get-iam-policy` : Get IAM policy for a project
    - `gcloud projects remove-iam-policy-binding` :Remove IAM policy binding
    - `gcloud projects set-iam-policy` :Set the IAM policy
    - `gcloud projects delete` : Delete a project
  - gcloud iam
    - `gcloud iam roles describe`
    - `gcloud iam roles create`
    - `gcloud iam roles copy`
  - 그 외
    - `gcloud compute project-info describe`
    - `gcloud auth info`
    - `gcloud auth revoke`
    - `gcloud auth list`



Google Developer Groups On Campus

서비스 계정

cloud storage에 접근해야 하지만 personal credential을 허용하고 싶지는 않을 때 씀
- 서비스 계정은 email주로 식별되며, @ 뒤에 developer.gserviceaccount.com이 붙음
- 비밀번호가 없음. 개인키 공용키만 할당됨. 그래서 브라우저나 쿠키로 로그인이 안되고, 컴퓨터에 할당된 계정임

서비스 계정의 종류
1. default service account: 서비스가 사용될 때 자동으로 만들어짐, 기본으로 editor role을 가지기에 추천하지 않음
2. user managed: 유저가 생성함: 세밀하게 권한 제어 할 수 있어 추천됨
3. google-managed service accounts: 구글에 의해 생성 관리됨, 우리는 신경 쓸 필요 없음 안 중요함.

Google Developer Groups On Campus

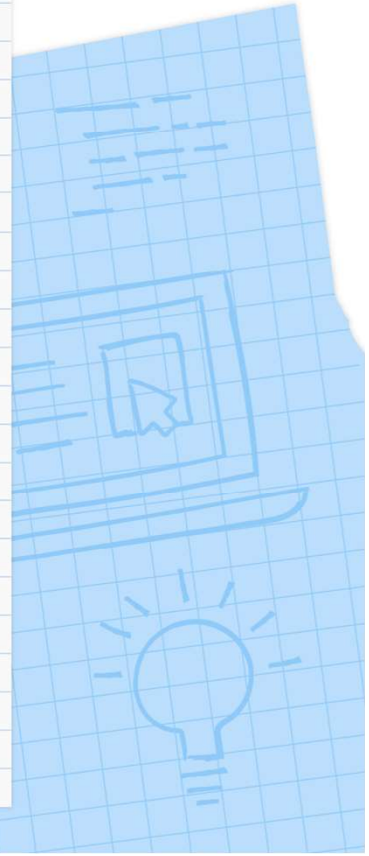
Use case

1.VM과 cloud storage 연결

1. service account role 에 올바른 권한들 넣어 계정 만들기
2. service account role을 VM에 할당하기.

•이때 구글 cloud-managed keys 사용되어 자동으로 편하게 다룰 수 있다. 즉 암호나 자격증명 파일 없이 접근할 수 있는 장점이 있다.

•그리고 인스턴스 작동중에 service account를 삭제해버리면 접근 권한을 잃을수 있으니 삭제하지 말것

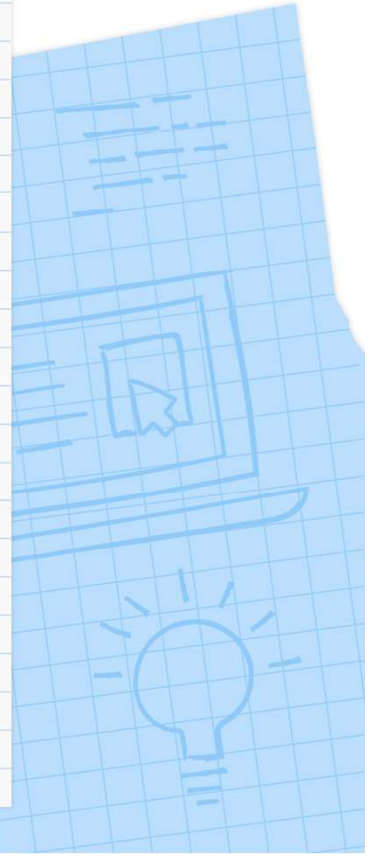


Use case

2.on prem(자체 서버)와 cloud storage 연결(long-lived)

service account를 prem에 바로 할당할 수는 없다.

1. service 계정을 올바른 권한으로 만든다.
2. service accoint user managed key를 생성한다.
3. 명령어 gcloud iam service-accounts keys create 를 통해 key를 만들고 service account key file을 다운로드 한다.
4. 환경 변수로 계정을 key file에 적용한다.
5. 구글 클라이언트 라이브러리는 ADC(application default credentials)를 기본 자격증명으로 제공하는데 이게 환경변수의 key file에 적용하는데 도움이 된다.



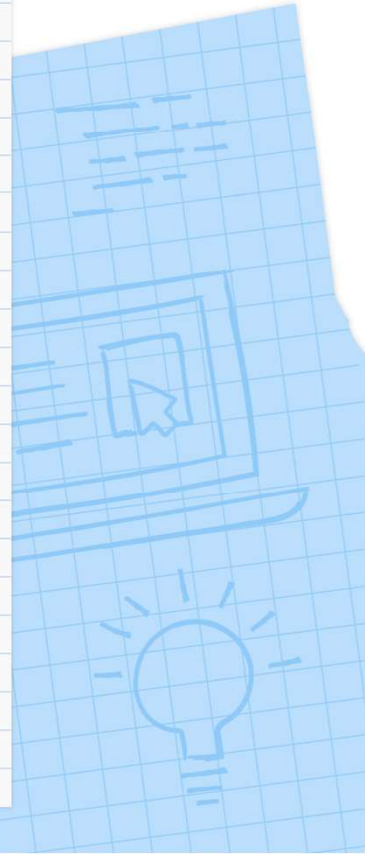
Google Developer Groups On Campus

Use case

3.on prem(자체 서버)와 google cloud APIs 연결(short lived)

잠깐만 사용할거라 오히려 장기간용 key쓰면 나중에 보안상 위협이 됨.

이런 경우 OAuth 2.0 access tokens, OpenId connect ID tokens, self-signed Json Web Tokens(JWTs) 를 사용할 수 있다.



Google Developer Groups On Campus

시나리오

Q: VM 응용 프로그램이 cloud storage bucket과 통신하길 바랍니다.
A: VM을 사용하기 위해 올바른 권한들을 넣은 service account를 구성하세요

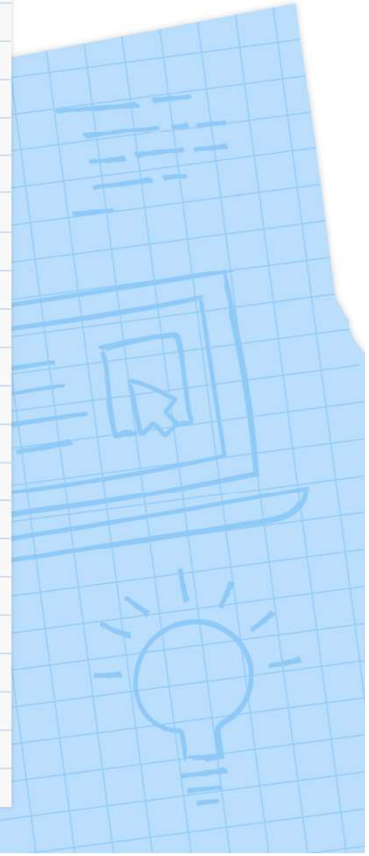
Q: 서비스 계정은 identity인가요 resource 인가요?
A: 둘다임. 서비스 계정을 규칙으로 첨부 가능(identitiy) 그리고 다른 맴버들도 서비스 계정에 역할을 부여 함으로써 접근 가능하다. (resource)

Q: 프로젝트 A의 기본 서비스 계정이 B 프로젝트의 storage bucket에 접근해야 합니다.
A: 프로젝트 B에서 A의 서비스 계정을 추가하고 버킷상의 저장소 개체를 할당하세요

ACL(access control lists)

- 누가 buckets과 object에 접근할지 level을 정함

.- IAM과 다른점. IAM은 모든 오브젝트에 동일한 권한을 주지만 ACL은 object에 따라 개별화된 접근 권한 제공 가능. 즉 좀더 세밀화됨



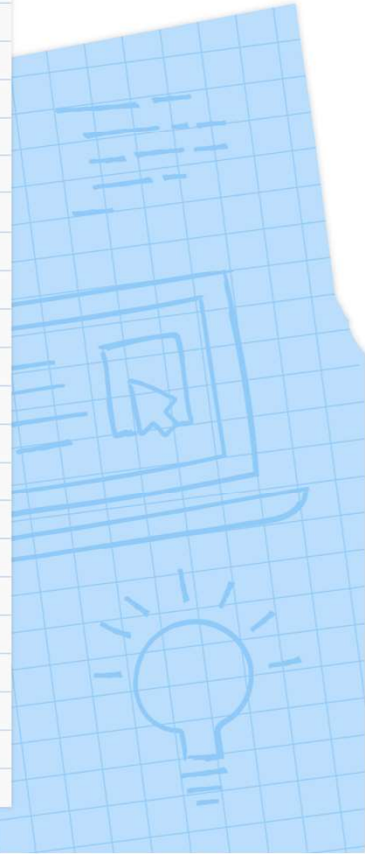
Google Developer Groups On Campus

Signed URL

URL을 통해 제한된 일정시간동안만, 특정 액션을 제공한다.
별도의 구글 계정이 필요 없다.

Signed URL은 permission에 따른 key를 만들고 아래 명령어를 통해 생성된다.

```
gsutil signurl -d 10m YOUR_KEY gs://BUCKET_NAME/OBJECT_PATH
```



Google Developer Groups On Campus

Cloud storage로 부터 정적인 website 만들기

website 이름과 동일한 bucket을 만든다.

파일들을 bucket으로 복사한다.

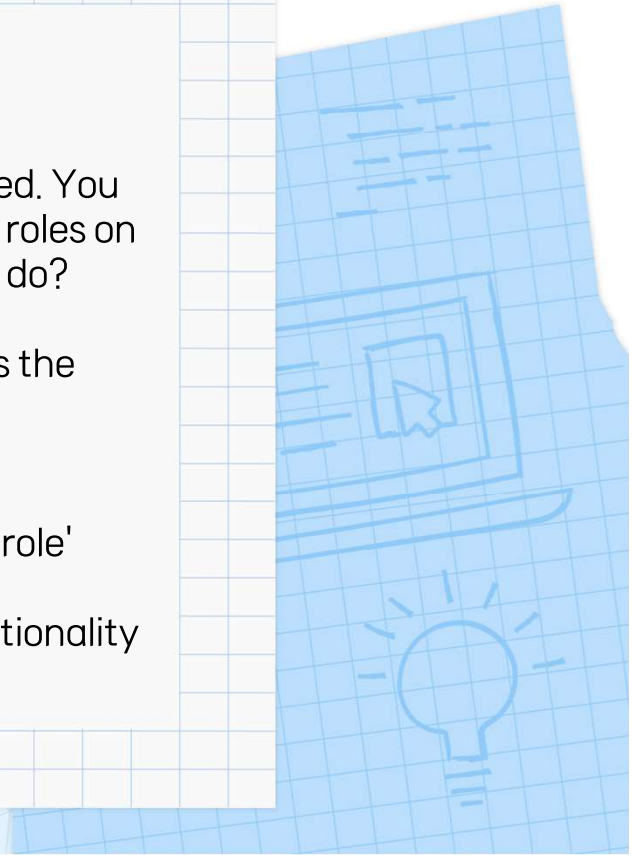
Storage Object viewer 옵션을 allusers로 허용한다. 이제 url로 모두 접근 할 수 있다.

Google Developer Groups On Campus

Quiz!

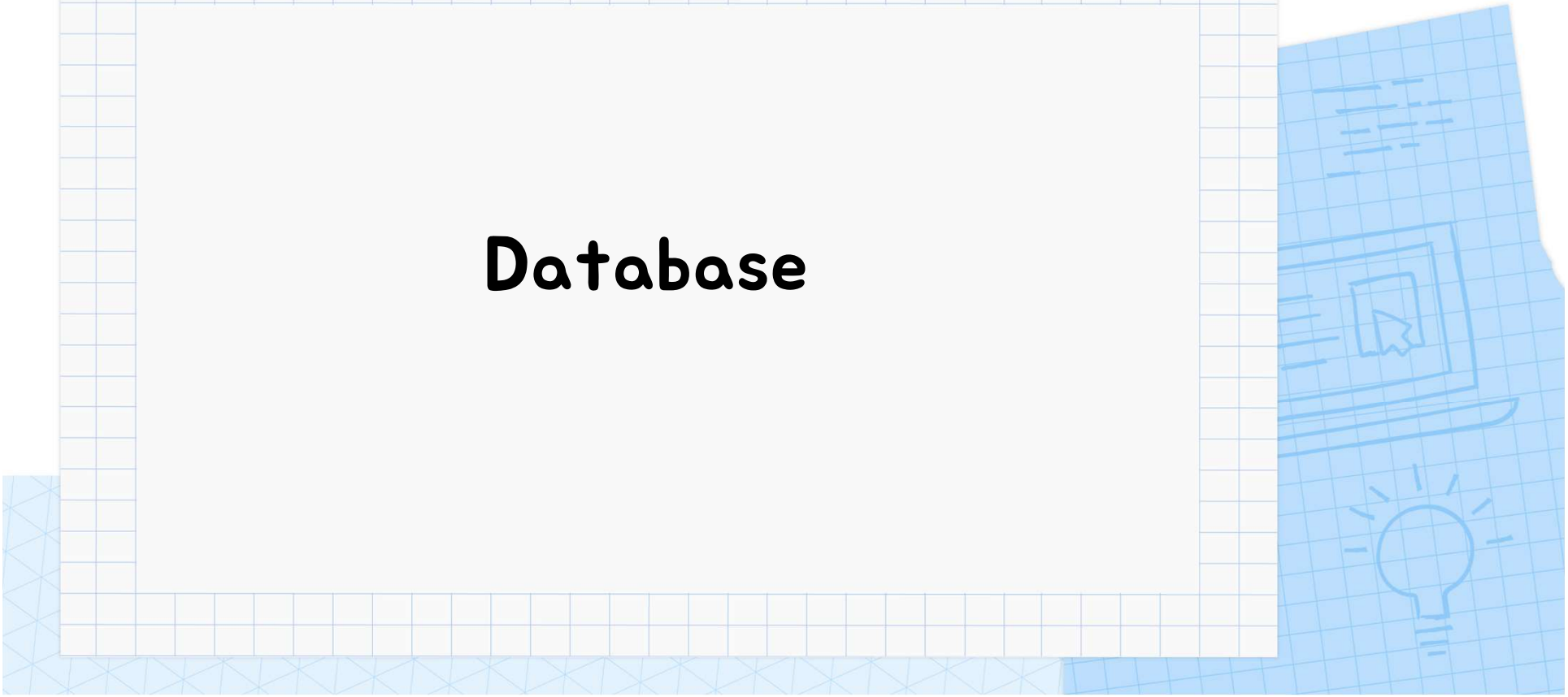
You have a development project with appropriate IAM roles defined. You are creating a production project and want to have the same IAM roles on the new project, using the fewest possible steps. What should you do?

A. Use gcloud iam roles copy and specify the production project as the destination project.
B. Use gcloud iam roles copy and specify your organization as the destination organization.
C. In the Google Cloud Platform Console, use the 'create role from role' functionality.
D. In the Google Cloud Platform Console, use the 'create role' functionality and select all applicable permissions.



Google Developer Groups On Campus

Database



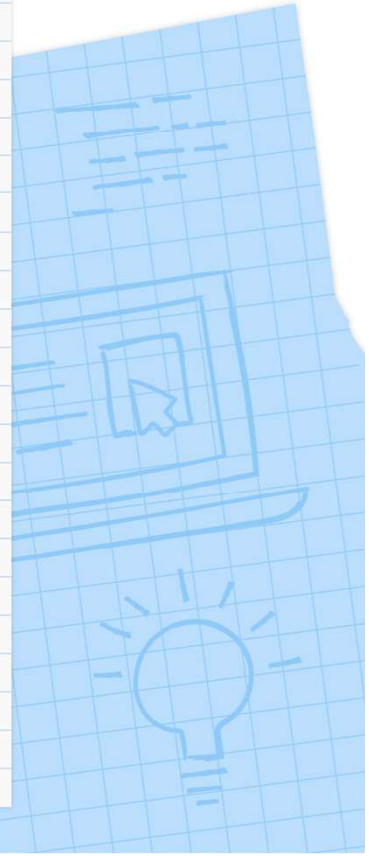
Google Developer Groups On Campus

Database 관련 문제 상황

만약 데이터베이스가 다운된다면 발생하는 문제
문제 1.접근 안되는 문제
문제 2. 데이터들 날라가는 문제

해결책1 snapshot: 매 시간마다 snapshot을 찍어 다른 데이터 센터에 저장하기
그렇다면 데이터들 날라가지 않게는 할 수 있지만 1번 접근 문제와, 스냅샷을 찍을때 느려지는 3번 문제가 추가로 발생함

해결책2 standby: 복사본인 standby database를 만들어 놓고, 장애때 돌리면
3가지 문제 다 해결됨



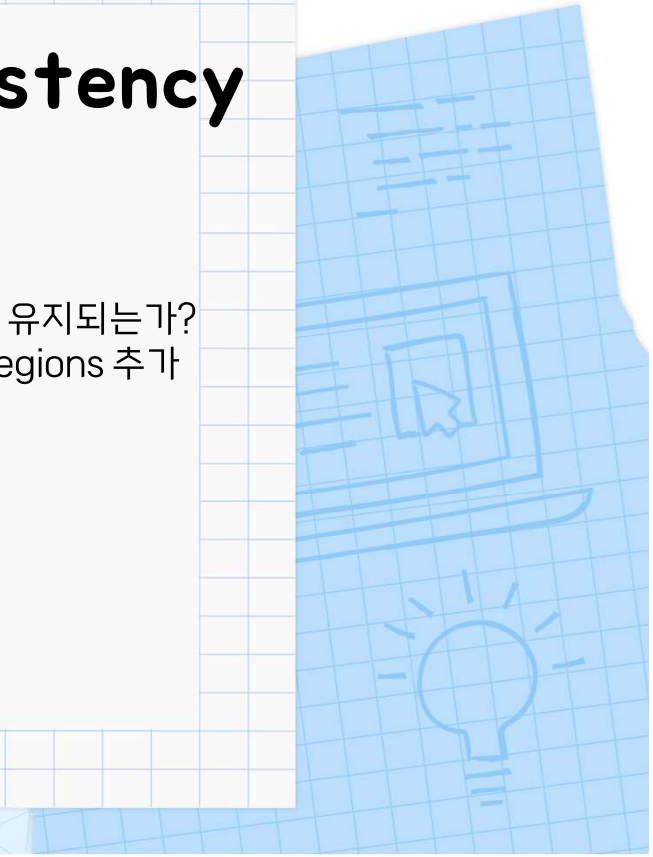
Google Developer Groups On Campus

Availability, Durablity, Consistency

availability 가용성 : 내가 원할 때 얼마나 빠르게 접근할 수 있는가
다양한 zones와 regions에 배포할 수록 가용성 높아짐

durability 내구성: 오랜 시간이 지날때 얼마정도의 파일이 손상되지 않고 유지되는가?
standby, snapshot, transaction,logs, replica 등을 다양한 zones과 regions 추가 보관

consistency 일관성: 여러 DB instance에서 동시에 업데이트 되는가?



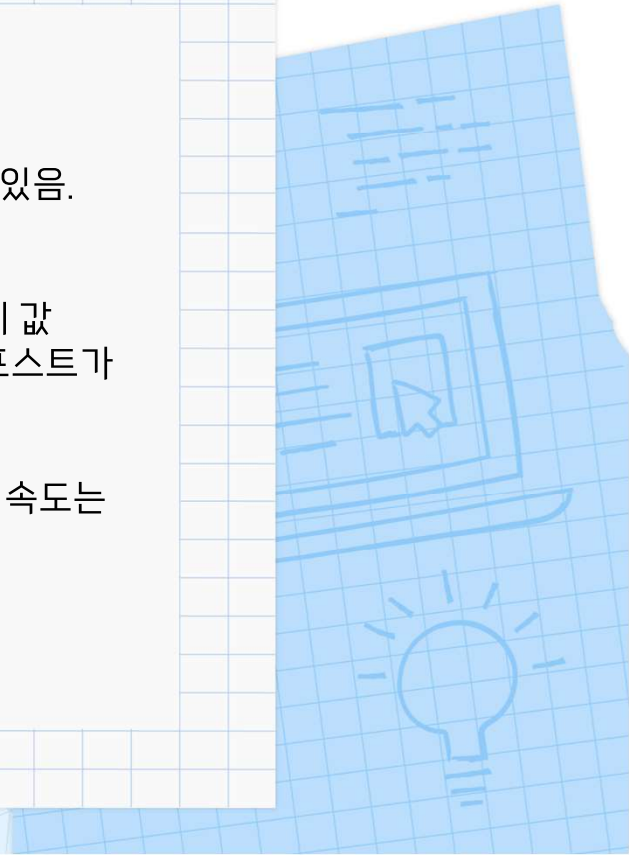
Google Developer Groups On Campus

Consistency

Strong consistency: 모든 복제로 즉시 업데이트 됨. 이로 인해 느려질 수 있음. 은행거래가 대표적

Eventual consistency: 비동기적이라 약간의 시간이 걸림. 그 시간 사이에 값 차이가 일어날 수 있고, 확장성이 무결성보다 중요할 때 쓰임. 소셜미디어 포스트가 대표적인 예임

Read-after-Write Consistency: 삽입 수정이 곧바로 일어나야 함. 복제속도는 느린만큼 접근은 빠름.



Google Developer Groups On Campus

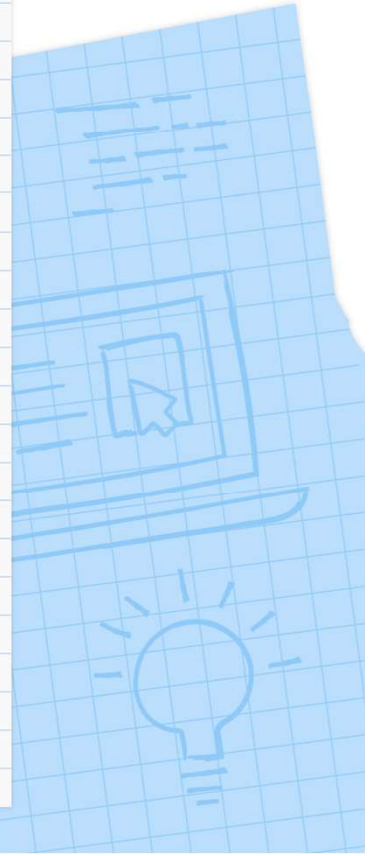
RTO와 RPO

RTO(Recovery Time Objective) : 최대 장애 시간 (다운되었을때 복구되는 시간)
RPO(Recovery Point Objective) : 최대 데이터 손실 주기 (백업주기)

- 일반적으로 RTO가 RPO보다 더 큼

- 데이터베이스를 reporting 하고 analytics 하는 응용 프로그램을 얹으면 좋음.(이런 application들은 읽기 전용임)
근데 이런거 얹으면 성능 느려질 수 있는데, CPU와 memory를 키우든 database cluster를 만들든 복제본을 따로 두어 그것만 읽게 하든 할 수 있음.

-읽기전용 복제본 따로 두어 읽게 하는 것을 추천



Google Developer Groups On Campus

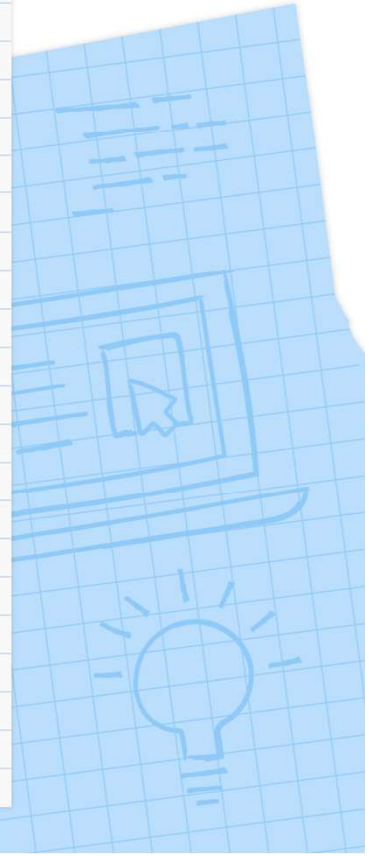
RTO와 RPO

시나리오	Solution
RTO와 RPO 둘다 엄청 작아야함	Hot standby 방식: 자동으로 데이터 동기화 하고 실패시 자동으로 standby로 연결되게 함
RTO는 좀 걸려도 괜찮지만 RPO는 작아야함	Warm standby: 자동으로 데이터는 동기화되게 하되 스탠바이 자체의 인프라는 작게하다가 문제 발생하면 scale up해서 사용
RTO는 오래걸려도 괜찮으나 RPO는 작아야함	snapshot이나 transaction logs 방식 사용
둘다 상관 X	그냥 아예 새로운 서버로 생성해 조치

Google Developer Groups On Campus

Realtional Databases

- 가장 많이 쓰였고 유명함.
- 사전에 정의된 엄격한 테이블과 스키마 가짐
- 트랜잭션이 강력하여 금융거래에 쓰임
- 그 종류로는 OLTP와 OLAP가 있음.



Google Developer Groups On Campus

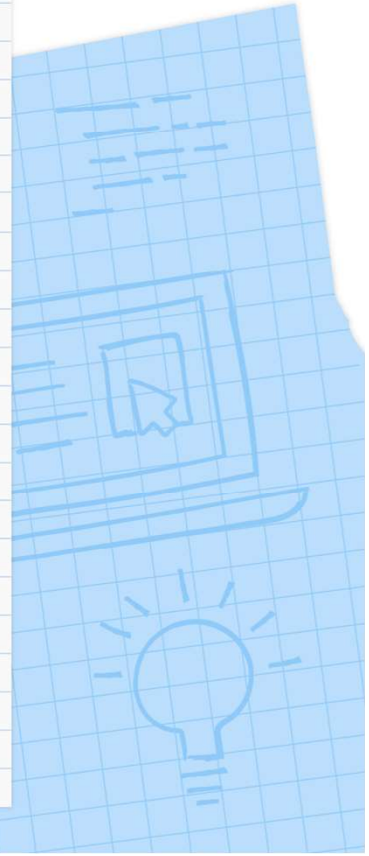
OLTP

OLTP(online transaction processing),

많은 사용자가 작은 양의 transaction 가질 때 쓰임.

ERP,CRM, E-커머스, 은행 앱 등에서 쓰이며 MySQL, Oracle, SQL 서버가 그 언어이다.

구글에서는 SQL을 기반으로 한 Cloud SQL과 대용량을 처리하는 Cloud Spanner를 제공한다.



Google Developer Groups On Campus

OLAP

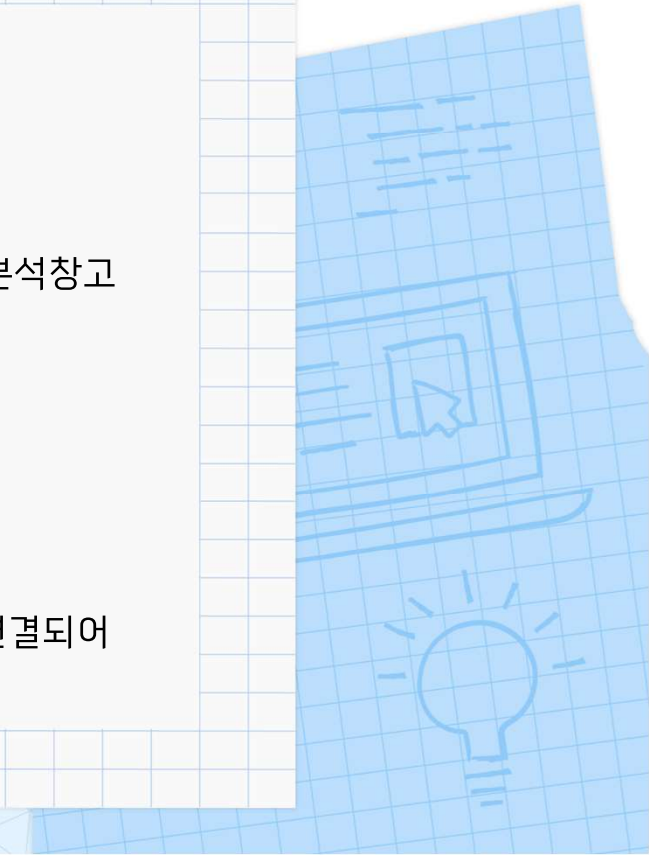
OLAP(online analytics processing)

정말 방대한 양의 데이터를 유저에게 허용하여 분석하게 해줄때. 데이터 분석창고 같은것이 그 예시이다.

GCP에서는 BigQuery가 그 예시이다.

OLTP와 OLAP는 비슷하나 데이터 저장방식이 다르다.
OLTP는 행단위로 저장되기에 transaction이 작은게 효율적이다.

OLAP는 열단위로 저장되기에 압축도가 높고, 각 속성끼리 노드 형태로 연결되어 있다. 복잡한 쿼리일 경우 실행이 효율적이다.



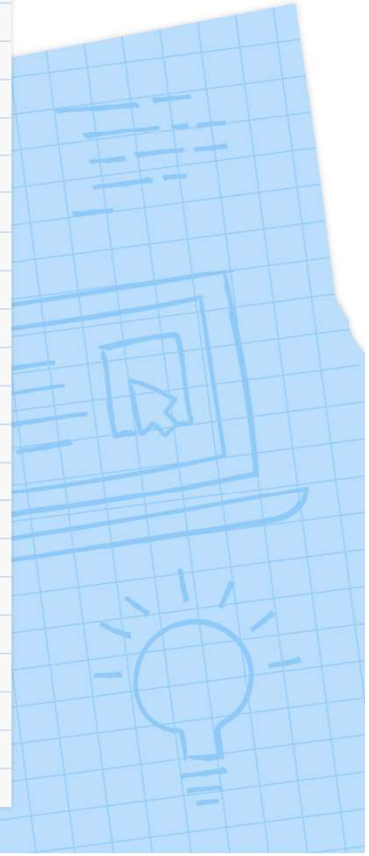
Google Developer Groups On Campus

NoSQL Databases

관계형인 SQL 뿐만 아니라 NoSQL의 유연한 스키마를 통해 수직적인 scale의 perabyte 단위를 다룰 수도 있음 .

일반적으로 확장성, 대용량과 고성능을 얻고, strong consistency 기능을 포기함

구글에서는 Cloud Firestore(Datastore)과 Bigtable이 있음.



Google Developer Groups On Campus

Datastore과 Bigtable

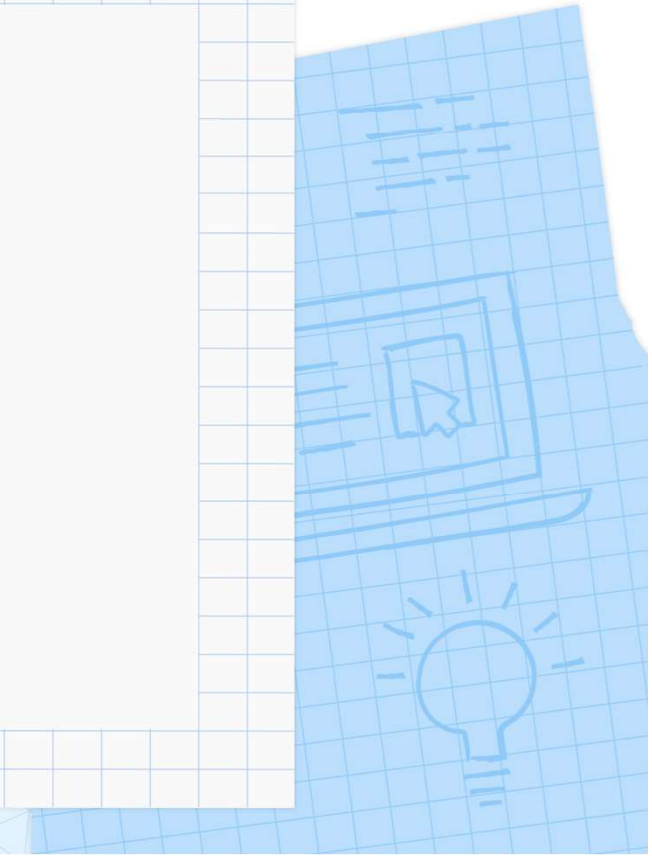
Datastore
- DataStore은 서버가 없는 NoSQL 로 ACID 트랜잭션, SQL 같은 쿼리, index를 제공함.
- 트랜잭션한 모바일 웹 어플리케이션을 위해 만들어짐
- 최신 버젼에 strong consisteny와 moblie web cilent library가 추가됨

Bigtable
- 규모의 확장성에 기반을 둔 수직 database임
- 서버 인스턴스가 있어야 함
- 10TB에서 페라바이트 이상일때 권장함
- 대용량 분석과 workload에서 사용 권장
- transactional workloads 에는 적합하지 않음. multi-row transaction 제공 안하고, singe-low transaction만 제공하기 때문

Google Developer Groups On Campus

In-Memory DB

- 디스크에서 찾는것보다 메모리에서 찾는게 더 빠름
- in-memory DB는 지속적인 데이터를 메모리에 저장해 적은 지연율
- GCP에서는 memory Store 가 있음
- 캐시, 세션, 리더보드 등을 처리할때 쓰임



Google Developer Groups On Campus

시나리오

Q: 테이블 구조로 빠르게 스키마 만들고 싶어요
A: Datastore/Firestore 쓰세요

Q: 적은 용량으로 비관계형 DB를 쓰고 싶어요
A: Datastore 쓰세요.

Q: 미리 짜둔 스키마로 트랜잭션하고, global 하여 초당 수백만개의 트랜잭션을 처리했으면 해요.
A: CloudSpanner 쓰세요.

Q: Transactional local DB를 초당 수천개씩 처리하고 싶어요.
A: Cloud SQL

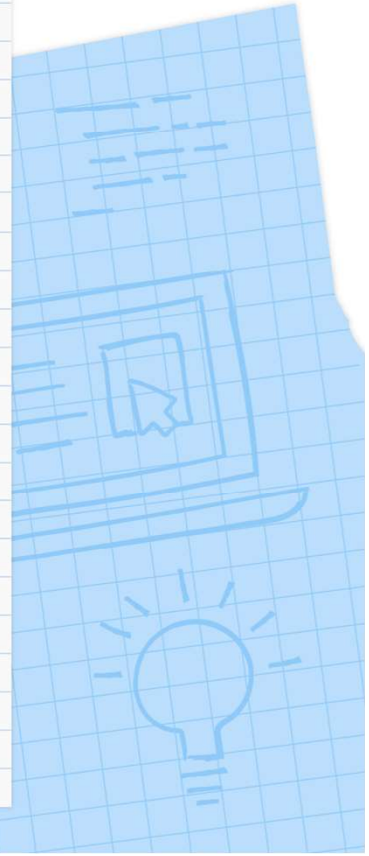
Google Developer Groups On Campus

시나리오

Q: 웹의 캐시 데이터
A: memorystore 쓰세요.

Q: 페라바이트 단위로 데이터 분석을 처리하고 싶어요.
A: BIgQuery 쓰세요.

Q: 엄청난 볼륨의 실시간 IOT 스트리밍 데이터
A: BigTable



Google Developer Groups On Campus

Quiz!

You want to select and configure a cost-effective solution for relational data on Google Cloud Platform. You are working with a small set of operational data in one geographic location. You need to support point-in-time recovery. What should you do?

A. Select Cloud SQL (MySQL). Verify that the enable binary logging option is selected.
B. Select Cloud SQL (MySQL). Select the create failover replicas option.
C. Select Cloud Spanner. Set up your instance with 2 nodes.
D. Select Cloud Spanner. Set up your instance as multi-regional.