

GCP기반의 금융데이터 분석





Day 1

INSTITUTE FOR INFORMATION



Chapter 01 GCP 소개

Chapter 02 GCE를 활용한 분석 환경 구축

Chapter 03 Cloud 환경에서의 Python 사용



Day 2

INSTITUTE FOR INFORMATION

Chapter 01 Cloud 환경에서의 금융 데이터 분석

Chapter 02 Python 기반의 BigQuery 접근과 데이터 조작



Day 3

INSTITUTE FOR INFORMATION

Chapter 01 크롤링을 통한 외부 데이터 수집과 적재

Chapter 02 Vertex AI를 활용한 머신러닝 모델 구현



- Day 2 -

클라우드 기반의 데이터 분석 시작하기



1) Cloud 환경에서의 금융데이터 분석

01 Cloud Storage 시작하기

[Day 02] 클라우드 기반의 데이터 분석 시작하기



Cloud Storage 개요

- ▶ 데이터를 저장하는 단순 저장소



Cloud Storage



버킷(Bucket)의 이해

- ▶ 데이터를 담는 기본 컨테이너. 디렉터리와 다르게 중첩 불가
- ▶ 버킷 내부에 폴더와 데이터를 위치시킴
- ▶ 내부에 폴더와 데이터가 있어도 바로 삭제할 수 있어 관리에 주의 필요



블롭(blob, Binary Large Object)의 이해

- ▶ 멀티미디어 파일 바이너리를 데이터베이스에 효과적으로 저장하기 위한 자료형
- ▶ 버킷에 저장된 파일의 경우 블롭으로 지칭하며 관련 파이썬 객체의 명칭 또는 메서드 또한 blob을 주로 사용

01 Cloud Storage 시작하기

[Day 02] 클라우드 기반의 데이터 분석 시작하기



버킷(Bucket) 생성

▶ 상단 메뉴의 **[[+] 만들기]** 버튼을 눌러 생성하며 총 5단계의 설정을 거쳐 최종 생성

Cloud Storage

버킷

모니터링

설정

버킷

필터 버킷 필터링

- 버킷 이름 지정
이름:
- 데이터 저장 위치 선택
위치: asia (아시아의 멀티 리전)
위치 유형: Multi-region
- 데이터의 스토리지 클래스 선택
기본 스토리지 클래스: Standard
- 객체 액세스를 제어하는 방식 선택
공개 액세스 방식: 사용
액세스 제어: 균일한 액세스 제어
- 객체 데이터를 보호하는 방법 선택
보호 도구: 없음
데이터 암호화: Google-managed key

알아두면 좋은 정보

위치별 가격 책정

스토리지 요금은 데이터의 스토리지 클래스와 버킷 위치에 따라 다릅니다.[가격 책정 세부정보](#)

현재 구성: Multi-region / Standard

항목	비용
asia (아시아의 멀티 리전)	GB당 월 \$0.026
기본 복제 사용	쓰기 1GB당 \$0.080

월 비용 예상

01 Cloud Storage 시작하기

[Day 02] 클라우드 기반의 데이터 분석 시작하기



버킷(Bucket) 생성 – 이름 지정

- ▶ 버킷명은 고유 해야 하며 민감한 정보를 포함하지 않는 것이 좋음
- ※ 상세 사항은 별도 지침(guideline) 참고



버킷 이름 지정

전역적으로 고유하고 영구적인 이름을 선택하세요. [이름 지정 가이드라인](#)

test

❗ 이미 사용 중인 버킷 이름입니다. 다른 이름을 입력하세요.



버킷 이름 지정

전역적으로 고유하고 영구적인 이름을 선택하세요. [이름 지정 가이드라인](#)

test-rloha-2077

팁: 민감한 정보를 포함하면 안 됩니다.

01 Cloud Storage 시작하기

[Day 02] 클라우드 기반의 데이터 분석 시작하기



버킷(Bucket) 생성 – 데이터 저장 위치

- ▶ 단일 리전(region), 다중 리전 중 선택할 수 있으며 프로젝트 성격에 맞는 선택 필요
- ▶ 단순 테스트를 위해서는 단일 리전의 asia-northeast3(서울) 선택 권장

리전	이중 리전	멀티 리전
가용성¹ <ul style="list-style-type: none"> 가용 영역 전반에 걸친 데이터 중복(동기) RTO=0: 영역에 장애 발생 시 자동 장애 조치 및 장애 복구(스토리지 경로를 변경할 필요 없음) 	<ul style="list-style-type: none"> 리전보다 높은 가용성 리전 전반에 걸친 데이터 중복(비동기) 15분 내에 복제되는 터보 복제 옵션 RTO=0: 리전에 장애 발생 시 자동 장애 조치 및 장애 복구(스토리지 경로를 변경할 필요 없음) 	<ul style="list-style-type: none"> 리전보다 높은 가용성 리전 전반에 걸친 데이터 중복(비동기) RTO=0: 리전에 장애 발생 시 자동 장애 조치 및 장애 복구(스토리지 경로를 변경할 필요 없음)
가격 책정 <ul style="list-style-type: none"> 최저 스토리지 가격 복제 요금 없음 동일 리전 내에서 데이터를 읽을 때 이그레스 요금 없음 	<ul style="list-style-type: none"> 최고 스토리지 가격 쓰기에 복제 요금 적용 한 리전 내에서 데이터를 읽을 때 이그레스 요금 없음 	<ul style="list-style-type: none"> 리전보다 높지만 이중 리전보다 낮은 스토리지 가격 쓰기에 복제 요금 적용 데이터를 읽을 때 항상 이그레스 요금 부과

01 Cloud Storage 시작하기

[Day 02] 클라우드 기반의 데이터 분석 시작하기



버킷(Bucket) 생성 – 스토리지 클래스 선택

▶ 데이터 접근(엑세스) 주기에 따른 클래스 선택

- 데이터의 스토리지 클래스 선택

스토리지 클래스는 업타임의 차이를 최소화하면서 스토리지, 가져오기, 작업 비용을 설정합니다. 객체를 자동으로 관리할지 선택하거나, 데이터와 워크로드를 저장할 기간이나 사용 사례를 기준으로 기본 스토리지 클래스를 지정하세요. [Learn more](#)

- 기본 클래스 설정

객체별로 클래스를 수동으로 수정하거나 객체 수명 주기 규칙을 설정하지 않는 한 버킷의 모든 객체에 적용됩니다. 사용량을 잘 예측할 수 있는 경우에 가장 적합합니다. 버킷을 만든 후에는 자동 클래스로 변경할 수 없습니다.

- Standard ?

단기 스토리지 및 자주 액세스하는 데이터에 적합

- Nearline

백업 및 월 1회 미만 액세스하는 데이터에 적합

- Coldline

재해 복구 및 분기당 1회 미만 액세스하는 데이터에 적합

- Archive

연 1회 미만 액세스하는 데이터의 디지털 장기 보존에 적합

01 Cloud Storage 시작하기

[Day 02] 클라우드 기반의 데이터 분석 시작하기



버킷(Bucket) 생성 – 객체 액세스 제어

▶ 별도의 암호키가 필요하지 않은 공개 접근 관련 설정 검토 필수

• 객체 액세스를 제어하는 방식 선택

공개 액세스 방지

인터넷을 통해 공개적으로 데이터에 액세스할 수 없도록 제한합니다. 이 버킷이 웹 호스팅에 사용되지 않게 합니다. [자세히 알아보기](#)

☒ 이 버킷에 공개 액세스 방지 적용

액세스 제어

☒ 균일한 액세스 제어

버킷 수준 권한(IAM)만 사용하여 버킷의 모든 객체에 대한 균일한 액세스 권한을 가지도록 합니다. 90일이 지나면 이 옵션이 영구적으로 적용됩니다. [자세히 알아보기](#)

☐ 세분화된 액세스 제어

버킷 수준 권한(IAM) 외에도 객체 수준 권한(ACL)을 사용하여 개별 객체에 대한 액세스 권한을 지정합니다. [자세히 알아보기](#)

01 Cloud Storage 시작하기

[Day 02] 클라우드 기반의 데이터 분석 시작하기



버킷 생성 확인

- ▶ 버킷의 각종 설정이 제대로 반영되었는지 확인 필요
- ▶ 최초 생성시 버킷은 비어있으며 [파일 업로드] 등 관련 버튼으로 파일 관리 가능

test-rloha-2077

위치	스토리지 클래스	공개 액세스	보호
asia-northeast3 (서울)	Standard	공개 아님	없음

객체 구성 권한 보호 수명 주기 관측 가능성 신규

버킷 > test-rloha-2077

파일 업로드 폴더 업로드 폴더 만들기 데이터 이전 보존 조치 관리 다운로드 삭제

이름 프리픽스로만 필터링 필터 객체 및 폴더 필터링

<input type="checkbox"/>	이름	크기	유형	생성 시간 ?	스토리지 클래스	최종 수정 날짜	공개 액세스 ?
--------------------------	----	----	----	---------	----------	----------	----------

표시할 행이 없습니다.

01 Cloud Storage 시작하기

[Day 02] 클라우드 기반의 데이터 분석 시작하기



파일 업로드 테스트 - 수동

▶ [파일 업로드] 버튼을 사용하여 파일을 로컬환경에서 업로드 할 수 있음

버킷 > test-rloha-2077

파일 업로드 폴더 업로드 폴더 만들기 데이터 이전 ▼ 보존 조치 관리

이름 프리픽스로만 필터링 ▼ 필터 객체 및 폴더 필터링

<input type="checkbox"/>	이름	크기	유형	생성 시간 ?
<input type="checkbox"/>	bank.csv	450.7KB	text/csv	2077. 11. 1...

01 Cloud Storage 시작하기

[Day 02] 클라우드 기반의 데이터 분석 시작하기



파일 업로드 테스트 – 파이썬

- ▶ "google-cloud-storage" 라이브러리 설치 필요
 - ▶ 라이브러리 설치 후 노트북 재부팅 권장
 - ▶ 설치 라이브러리명과 불러오는 라이브러리명이 다르기 때문에 주의
 - ▶ 원활한 접근을 위해 의 계정의 서비스키 정보가 들어있는 .json 파일 준비 권장
- ※ [IAM] → [서비스 계정] → [서비스 계정 만들기] → [키]

```
!pip install google-cloud-storage --user
```

```
from google.cloud import storage
import os
```

```
os.environ["GOOGLE_APPLICATION_CREDENTIALS"] = "json 파일 경로"
```

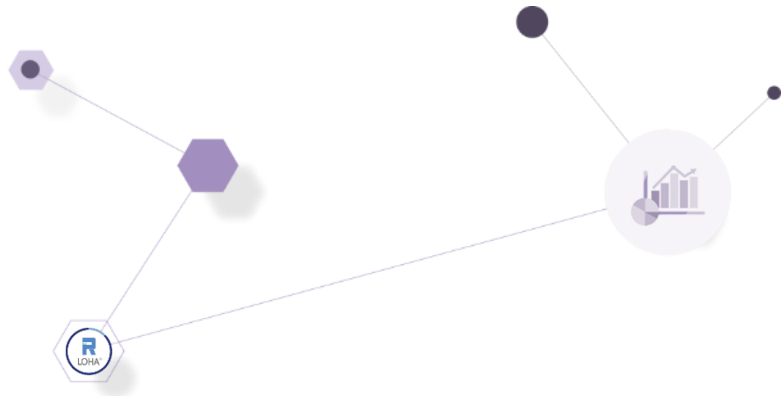
```
storage_client = storage.Client()
buckets = list(storage_client.list_buckets())
buckets
```

02 파이썬 기반 GCE 활용

[Day 02] 클라우드 기반의 데이터 분석 시작하기

</> 관련 기능([google](#) > [cloud](#) > [storage](#))

기능	설명
.Client()	접속 클라이언트 정보 반환
.list_buckets()	클라이언트의 버킷 목록 반환
.list_blobs()	특정 버킷의 파일(blob) 목록 반환
.name	특정 blob 객체에서 파일명 추출에 사용



2) Python 기반의 BigQuery 접근과 데이터 조작

01 BigQuery 시작하기

[Day 02] 클라우드 기반의 데이터 분석 시작하기



BigQuery 개요

- ▶ 쿼리 엔진이 내장된 서버리스(serverless) 서비스로, 확장성이 높은 데이터 웨어하우스
- ▶ 인프라를 관리 필요 없이 쿼리 실행 가능
- ▶ Apps Script, Looker, Data Studio 같은 다양한 도구와 연계 가능
- ▶ 배치 데이터와 스트리밍 데이터 수집 모두 지원
 - ※ REST API를 활용한 데이터 스트리밍 지원
- ▶ 컬럼 기반의 연산과 파티션, 클러스터링 지원으로 효율적 데이터 처리 가능



BigQuery



가격 정책

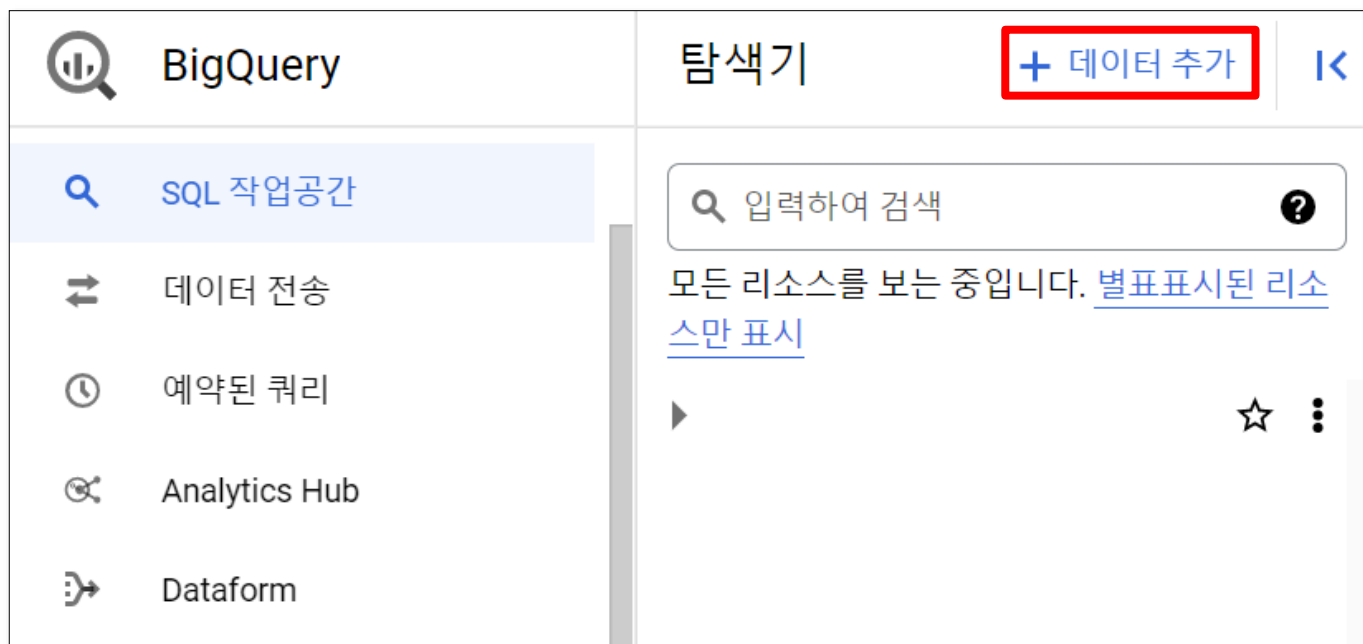
- ▶ 쿼리 실행시 발생하는 비용과 데이터 보관시 발생하는 비용으로 크게 나뉘어짐
- ▶ 매월 처리되는 데이터 1TB당 5\$가 청구되며 첫 1TB에 대한 비용은 무료
- ▶ 열(columns) 기반의 처리이기 때문에 쿼리의 "SELECT" 문에서 선택하는 열 개수에 따라 비용이 차이남

01 BigQuery 시작하기

[Day 02] 클라우드 기반의 데이터 분석 시작하기

💡 데이터 추가(데이터 업로드)

▶ [SQL 작업공간] 탭을 누른 후 화면 상단의 [+데이터 추가] 버튼을 통해 데이터 업로드



01 BigQuery 시작하기

[Day 02] 클라우드 기반의 데이터 분석 시작하기



데이터 추가(데이터 업로드) - 소스(출처) 선택


- ▶ 데이터는 로컬파일, GCE 등 다양한 출처로 부터 가져올 수 있음
- ▶ 데이터 추가를 위한 출처는 Google Drive 같은 Google 제품군을 포함하여 AWS 같은 타 서비스도 일부 지원


데이터 추가


Source

데이터 소스 검색

인기 소스


로컬 파일
로컬 파일 업로드


Google Cloud Storage
Google 객체 스토리지 서비스


외부 데이터 소스에 대한 연결
BigQuery에서 외부 데이터 소스로 연결



Google Drive
Google 스토리지 서비스



Amazon S3 - Data Transfer
Data Transfer Service를 통한 Amazon 객체 스토리지 서비스

01 BigQuery 시작하기

[Day 02] 클라우드 기반의 데이터 분석 시작하기



데이터 추가(데이터 업로드) – 데이터 설정

- ▶ 추가하는 데이터는 프로젝트, 데이터 세트명, 테이블명 지정이 필요
- ▶ 추가하는 데이터는 테이블로 지칭하며 이는 프로젝트 내의 데이터 세트 아래에 위치함

대상

프로젝트 *

찾아보기

데이터세트 *

sample_data

테이블 *

sample_data_tbl

유니코드 문자, 표시, 숫자, 커넥터, 대시, 공백이 허용됩니다.

테이블 유형

기본 테이블



데이터세트 만들기

프로젝트 ID

변경

데이터세트 ID *

sample_data

문자, 숫자, 밑줄이 허용됩니다.

데이터 위치

us (미국의 멀티 리전)



01 BigQuery 시작하기

[Day 02] 클라우드 기반의 데이터 분석 시작하기



데이터 추가(데이터 업로드) – 스키마(schema)

- ▶ 추가 데이터는 스키마(schema) 설정이 필요하며 이는 수동과 자동 설정으로 나뉘어짐
- ▶ 별도의 최적화된 설정이 필요한 것이 아니라면 스키마 **[자동 감지]** 기능을 이용하는 것이 편리

스키마

☐ 자동 감지

☒ 텍스트로 편집

필드 이름 *

유형 *
STRING ▼

모드
NULLABLE ▼

최대 길이

설명

❗ 필드 이름은 필수 항목입니다.



스키마

☒ 자동 감지

❗ 스키마가 자동으로 생성됩니다.

01 BigQuery 시작하기

[Day 02] 클라우드 기반의 데이터 분석 시작하기



데이터 추가(데이터 업로드) – 추가 데이터 확인

▶ 추가된 데이터는 좌측 [탐색기] 메뉴에서 [데이터 세트]의 [테이블명]을 클릭하여 각 필드 정보 확인 가능

탐색기

+ 데이터 추가

<

🔍 입력하여 검색

?

모든 리소스를 보는 중입니다. [별표표시된 리소스만 표시](#)

▶ 외부 연결

▶ 저장된 쿼리 (2)

▼ sample_data

☆ ⋮

sample_data_tbl

☆ ⋮

sample_data_tbl × +

sample_data_tbl

🔍 쿼리 ▼

+ 공유

스키마

세부정보

미리보기

≡ 필터

속성 이름 또는 값 입력

<input type="checkbox"/>	필드 이름	유형	모드
<input type="checkbox"/>	datetime	TIMESTAMP	NULLABLE
<input type="checkbox"/>	casual	INTEGER	NULLABLE
<input type="checkbox"/>	registered	INTEGER	NULLABLE
<input type="checkbox"/>	count	INTEGER	NULLABLE

02 데이터 조회 – Query

[Day 02] 클라우드 기반의 데이터 분석 시작하기

</> 쿼리(Query) 기반 조회

- ▶ 우측 쿼리 작성창에서 쿼리 기반으로 데이터 조회 가능
- ▶ 기본 ANSI SQL을 따르나 BigQuery 만의 별도 문법 존재
 - ※ EXCEPT(), 파티션, 클러스터링 등

▶ 실행

📁 저장 ▼

👥 공유 ▼

🕒 일정 ▼

⚙️ 더보기 ▼

1

select * from sample_data.sample_data_tbl limit 5

Alt+F1을 눌러 접근성 옵션을

쿼리 결과

📄 결과 저장 ▼

📊 데이터 탐색 ▼

작업 정보

결과

JSON

실행 세부정보

실행 그래프

미리보기

행	datetime	casual	registered	count	
1	2011-01-01 04:00:00 UTC	0	1	1	
2	2011-01-01 05:00:00 UTC	0	1	1	
3	2011-01-02 07:00:00 UTC	0	1	1	
4	2011-01-03 04:00:00 UTC	0	1	1	
5	2011-01-04 02:00:00 UTC	0	1	1	

03 데이터 조회 – Python

[Day 02] 클라우드 기반의 데이터 분석 시작하기

Python 기반 조회

- ▶ "google-cloud-bigquery", "db-dtypes" 라이브러리 설치 필요
 - ▶ 라이브러리 설치 후 노트북 재부팅 권장
 - ▶ 설치 라이브러리명과 불러오는 라이브러리명이 다르기 때문에 주의
 - ▶ 원활한 접근을 위해 의 계정의 서비스키 정보가 들어있는 .json 파일 준비 권장
- ※ [IAM] → [서비스 계정] → [서비스 계정 만들기] → [키]

```
!pip install google-cloud-bigquery --user
```

```
!pip install db-dtypes --user
```

```
from google.cloud import bigquery as bq  
import os
```

```
os.environ["GOOGLE_APPLICATION_CREDENTIALS"] = "json 파일 경로"
```

```
client = bq.Client()  
query_01 = "select * EXCEPT(casual) from sample_data.sample_data_tbl limit 5"  
result_q = client.query(query_01)  
result_q.to_dataframe()
```

03 데이터 조회 – Python

[Day 02] 클라우드 기반의 데이터 분석 시작하기

 **관련 기능(google > cloud > bigquery)**

기능	설명
.Client()	접속 클라이언트 정보 반환
.query()	클라이언트 정보 기반으로 BigQuery에 쿼리 전송
.to_dataframe()	쿼리 결과를 Pandas DataFrame 으로 변환