

# Google Cloud Platform & SQL

: GCP로 메모리 에러 탈출하기

2020.02.06.  
2기 정민지



# CONTENTS

## 01 클라우드란?

- 개념
- 장점

## 02 GCP (Google Cloud Platform)

- 소개
- 유용한 서비스

## 03 GCP에서 데이터분석 환경 구축

- 방법 1) Datalab
- 방법 2) VM 인스턴스에 anaconda 직접 설치
- 외부프로그램 접속
- 백그라운드에서 계속 실행하기

## 04 BigQuery & SQL

- Big Query vs Cloud SQL
- SQL이란?
- SQL 및 데이터베이스 용어
- SQL의 데이터 타입
- SQL 문법의 종류
- SELECT
- JOIN
- BigQuery 실습

## 05 Jupyter Notebook에서 BigQuery 사용

- 방법 1) GCP 가상머신에서
- 방법 2) 로컬에서

## 06 참고자료

# 01

클라우드란?

# 01. 클라우드란?

- 클라우드 서비스 개념

- 인터넷을 통해 가상화된 컴퓨터의 IT 리소스를 서비스로 제공하는 것
- 즉, 자신의 컴퓨터가 아닌 클라우드(인터넷)에 연결된 가상의 컴퓨터로 처리하는 기술



# 01. 클라우드란?

- 클라우드 서비스 장점

머신러닝 프로젝트를 진행할 때 **n GB**의 대용량 데이터를 분석해야 하는 경우가 종종 있음  
로컬 (본인 컴퓨터)에서 작동하지 않는 문제 발생

## 로컬로 할 때 문제점

- 너무 느림
- 메모리에러 발생
- 저장공간 부족

## 클라우드 서비스를 이용하면,

적은 돈으로 훨씬 좋은 성능의 컴퓨터 사용  
→ 비용과 시간 절약

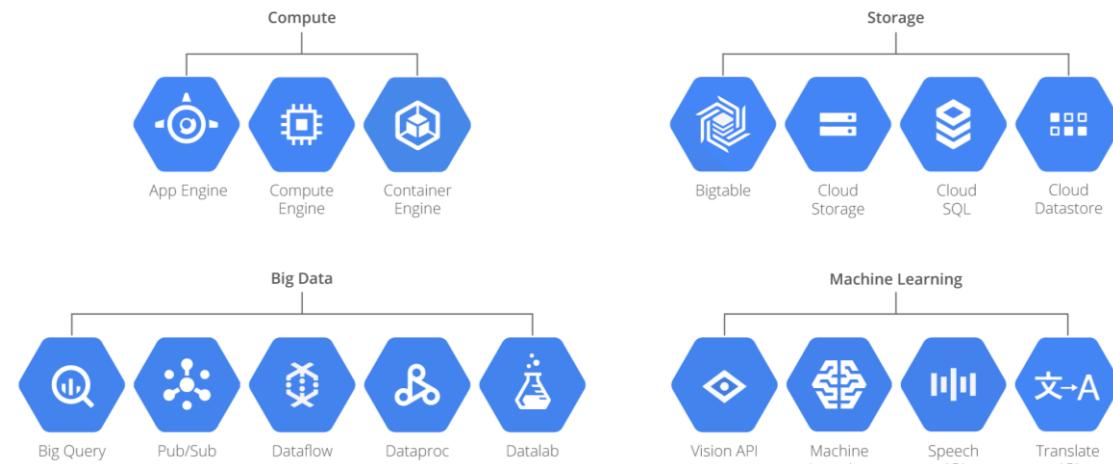
# 02

Google Cloud Platform (GCP)

# 02. Google Cloud Platform (GCP)

- Google의 데이터센터 인프라를 기반으로 Compute, Storage, Big Data, Machine Learning 등의 서비스를 제공하는 글로벌 클라우드
- Big Data, TensorFlow 프레임워크과 같은 높은 컴퓨팅 오퍼링을 전문으로 하며, 기계 학습의 혁신적인 기능은 물론 BigQuery 분석 엔진 및 Cloud Spanner 분산 데이터베이스를 제공
- **1년 동안 서비스 제한 없이 사용할 수 있는 \$300 크레딧 제공**
- **직접 유료 계정으로 전환하지 않는 한 비용 청구가 안됨** (& Free trial 기간이 끝나고도 항상 무료로 사용가능한 서비스 제공)
- 사용자 친화적이고 직관적인 UI
- 빅데이터, 머신러닝 플랫폼이 사용하기 편리하게 구성되어 있음

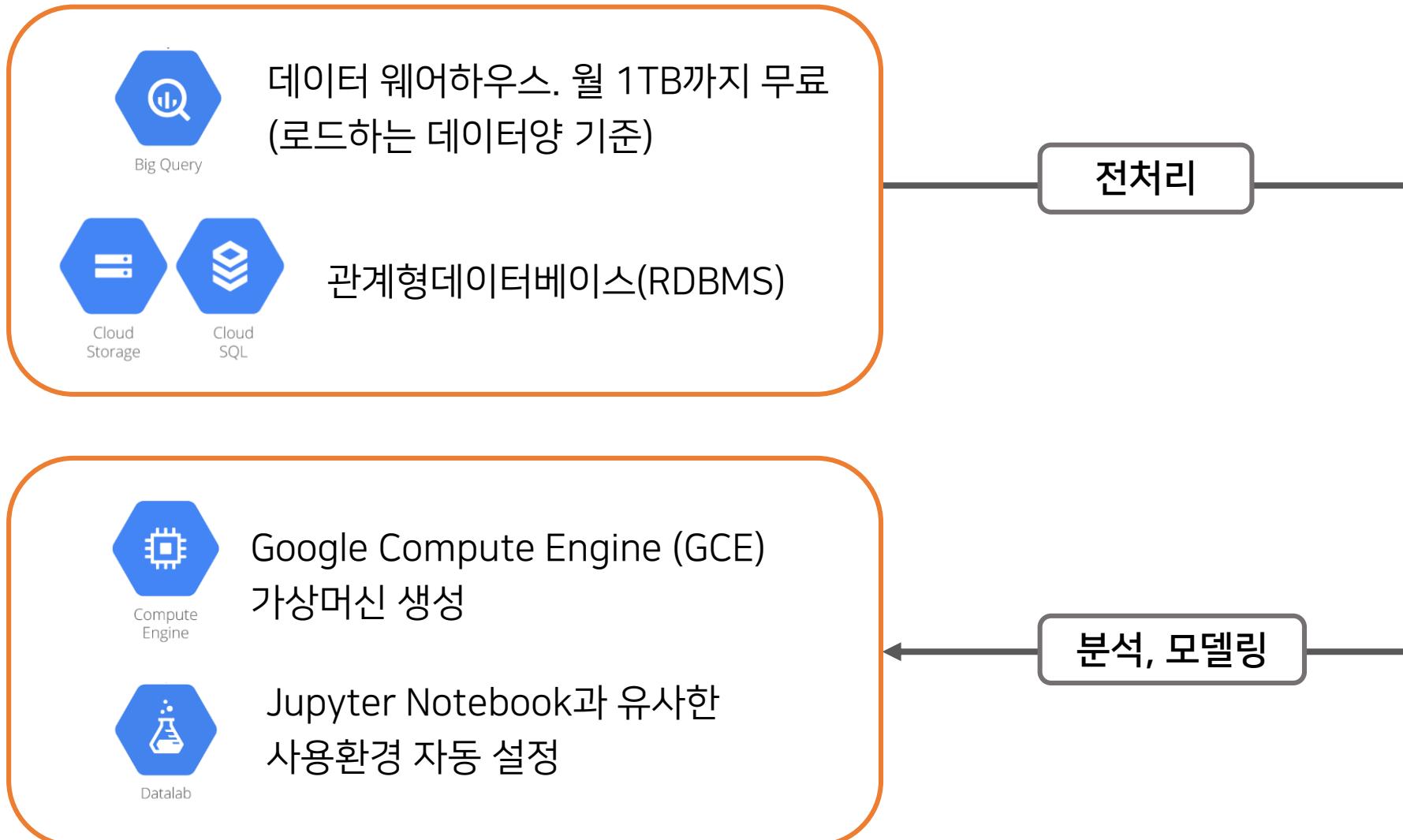
※ AWS는 세계적 규모의 컴퓨팅 센터를 보유하며 서비스 종류가 매우 다양하고 체계적인 교육 서비스를 제공함  
그러나 무료 크레딧으로는 단 한 개의 저성능 인스턴스만 사용할 수 있음 & 무료 크레딧 다 쓰면 자동 과금



[ GCP에서 제공하는 서비스 목록 ]

# 02. Google Cloud Platform (GCP)

- 머신러닝 프로젝트에서 유용하게 쓰이는 서비스들



# 03

GCP에서 데이터 분석 환경 구축하기

# 03. GCP에서 데이터 분석 환경 구축하기

- 분석환경 구축하기

## 방법 1) Google Cloud 에서 제공하는 Datalab 사용

- Jupyter Notebook 환경을 자동으로 구축해줘서 편리함
- 자동으로 끊기는 시간 있으니 모델 학습 시 끊기지 않도록 수동 설정 필요

## 방법 2) 가상머신에 직접 Anaconda 설치해서 사용

- 로컬에서와 똑같은 형태로 Jupyter Notebook 사용 가능

# 03. GCP에서 데이터 분석 환경 구축하기

- step0. 계정 생성 및 무료 크레딧 받기

1) <https://cloud.google.com/free/>

2) 구글 로그인 (구글 계정 없으면 만들기)

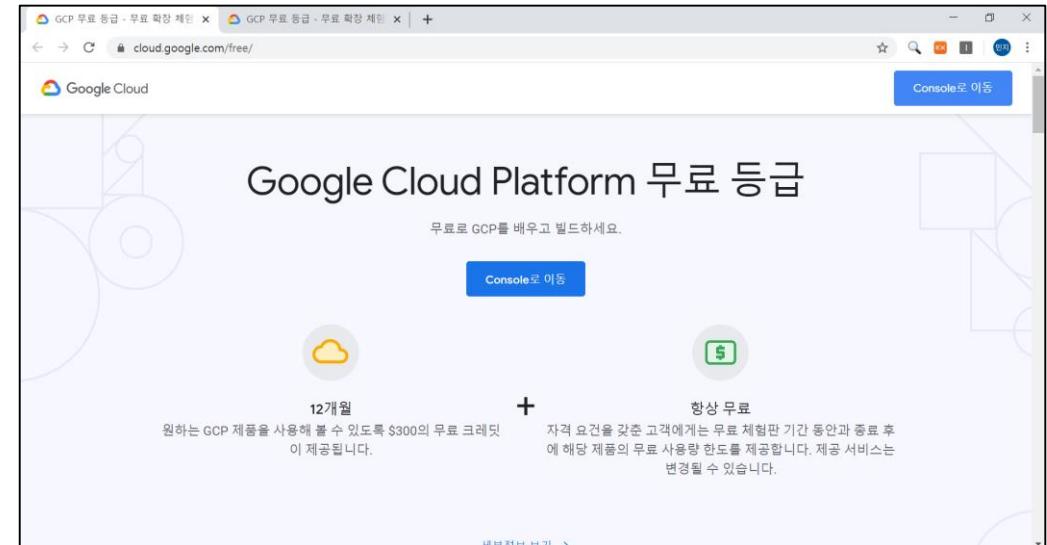
3) 절차에 따라 신용카드 등록 후

4) 1년 간 사용할 수 있는 300\$ 크레딧 받기

- 1년이 지나거나 300\$를 모두 쓰면 서비스 종료
- 유료계정 업그레이드 전에는 과금 절대 안됨
- 종료후에는 클라우드에 저장된 데이터가 손실되기 때문에  
종료전에 미리 백업해둘것

※ 참고

Free trial 기간이 끝나고도 항상 무료로 사용가능한 서비스 제공



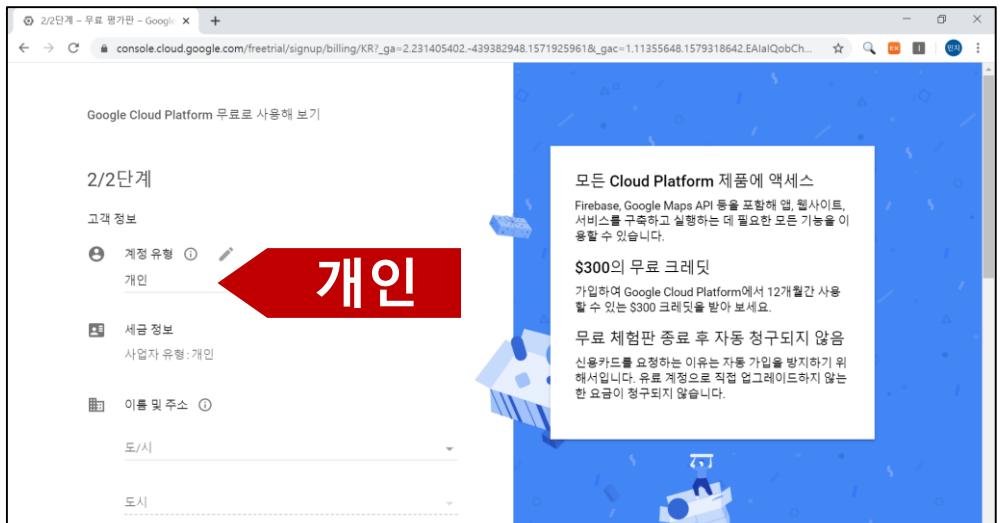
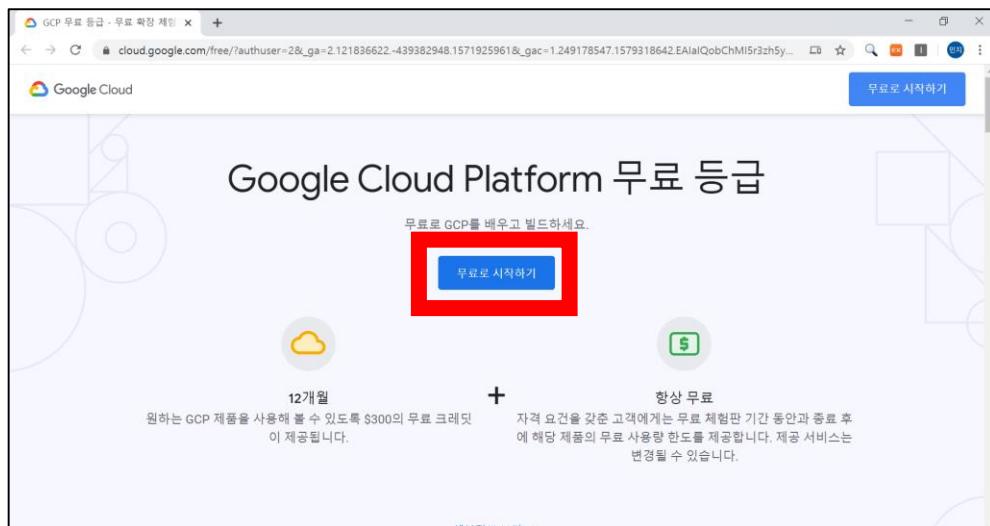
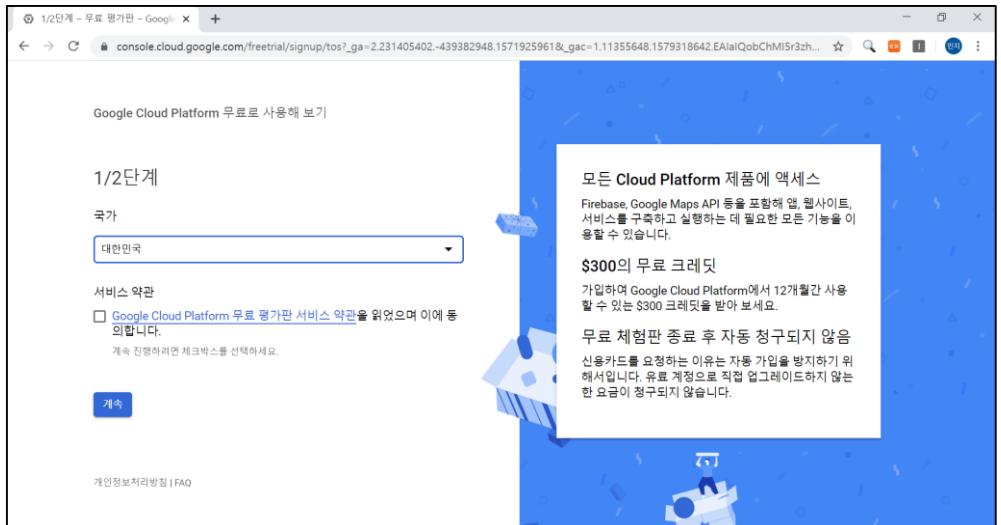
※ 기존 사용자

기존에는 해외 결제 가능한 카드만 등록이 가능하고 크레딧도 \$로 제공되었으나 얼마전부터 원화 결제 가능해짐

→ 해외 결제 안되는 카드여도 등록 가능하고 크레딧도 ₩로 제공

→ 대신 계정 이전 필요 (구글에서 오는 메일 확인 후 변경)

## 처음 Google Cloud 가입 후 카드 등록하는 과정 – 이미 했으면 생략



# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 1) Google Cloud에서 제공하는 Datalab 사용

방법 1) Google Cloud에서 제공하는 Datalab 사용

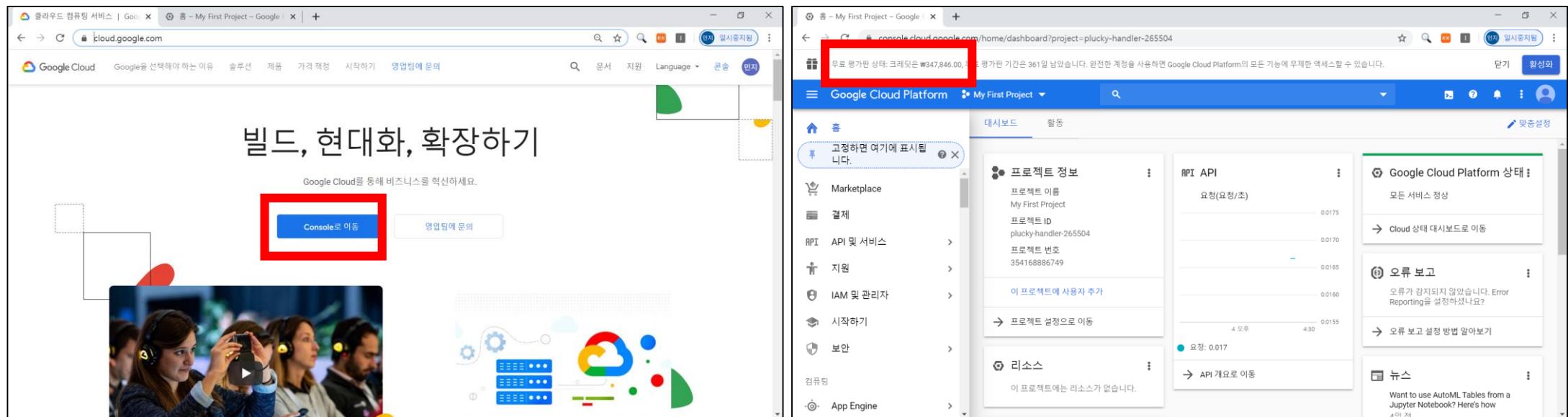
# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 1) Google Cloud에서 제공하는 Datalab 사용

## Step 1. 프로젝트 생성

- 프로젝트는 VM(virtual machine, 가상머신)이나 네트워크 자원, SQL 등 클라우드 내의 자원을 묶어서 관리하는 하나의 집합
- 각 시스템 혹은 팀별로 프로젝트 나눠서 정의하면 관리하기 쉬움

(1) <https://cloud.google.com/> 접속 후 Console로 이동 (참고 : 무료 평가판 남은 크레딧 확인 가능)



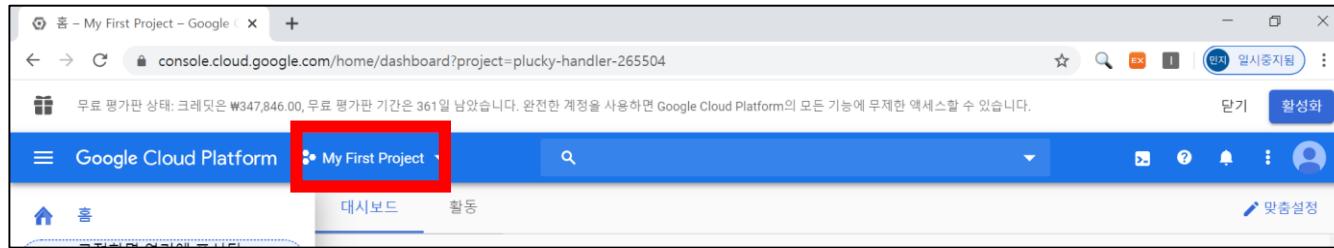
# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 1) Google Cloud에서 제공하는 Datalab 사용

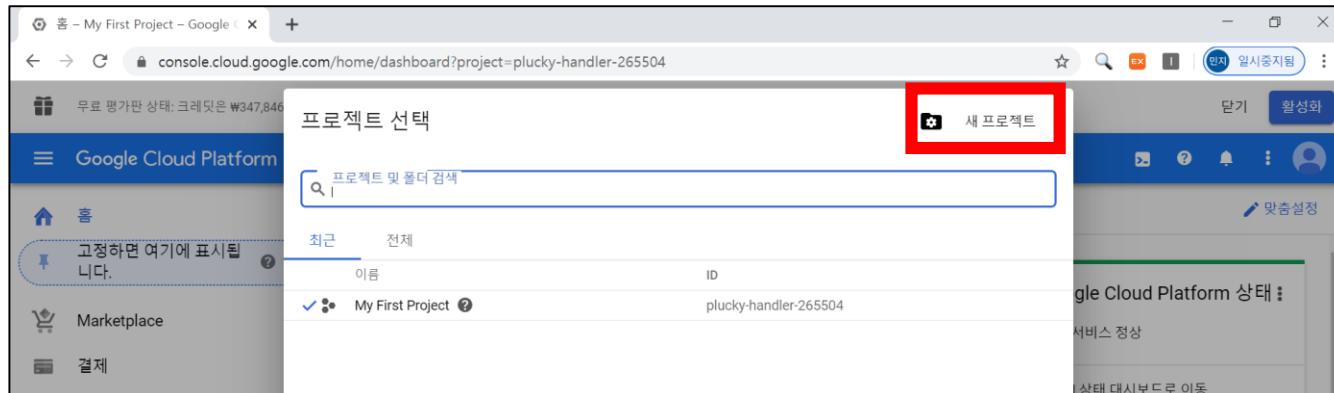
## Step 1. 프로젝트 생성

(2) 화면 좌측 상단에서 My First Project 클릭

※ My First Project는 기본적으로 생성되어 있는 프로젝트 – 이것을 사용하려면 프로젝트 생성 과정은 생략



(3) 새 프로젝트 클릭



# 03. GCP에서 데이터 분석 환경 구축하기

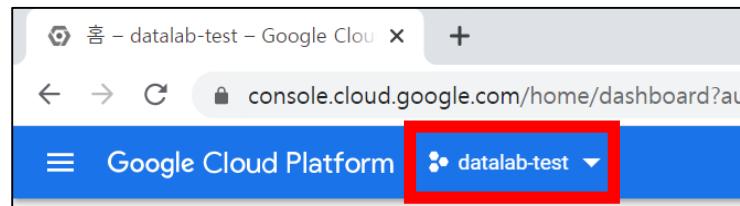
- 방법 1) Google Cloud에서 제공하는 Datalab 사용

## Step 1. 프로젝트 생성

(4) 가운데와 같이 프로젝트 이름을 입력 받는 창 나오면 프로젝트 이름 입력

The screenshot shows the 'Google Cloud Platform' interface for creating a new project. At the top, there's a blue header bar with the title 'Google Cloud Platform'. Below it, a search bar and a 'New Project' button are visible. The main area is titled '새 프로젝트'. A warning message says 'projects 할당량이 21개 남았습니다. 할당량 증가를 요청하거나 프로젝트를 삭제하세요.' with a link '자세히 알아보기'. Below this is a 'MANAGE QUOTAS' button. The 'Project Name\*' input field is highlighted with a red box and contains the text 'datalab-test'. Below the input field, the project ID 'datalab-test-266906' is shown with a note that it cannot be changed. There are fields for 'Location' (set to 'No organization') and 'Organization or folder'. At the bottom are 'Create' and 'Cancel' buttons.

(4) 새 프로젝트 생성 완료 후 새 프로젝트로 이동



# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 1) Google Cloud에서 제공하는 Datalab 사용

Step 2. Google Compute Engine 및 Cloud Source Repositories 필요한 API 사용 설정

(1) 콘솔에서 좌측탭 클릭 → API 및 서비스로 이동 → + API 및 서비스 사용 설정 클릭

The image shows two screenshots of the Google Cloud Platform API & Services dashboard. The left screenshot shows the navigation bar with 'API & Services' selected. The right screenshot shows the main dashboard with a red box highlighting the '+ API 및 서비스 사용 설정' (Add API & Service Usage Settings) button in the top right corner. Below this button, there are sections for traffic, quota, and regional usage statistics.

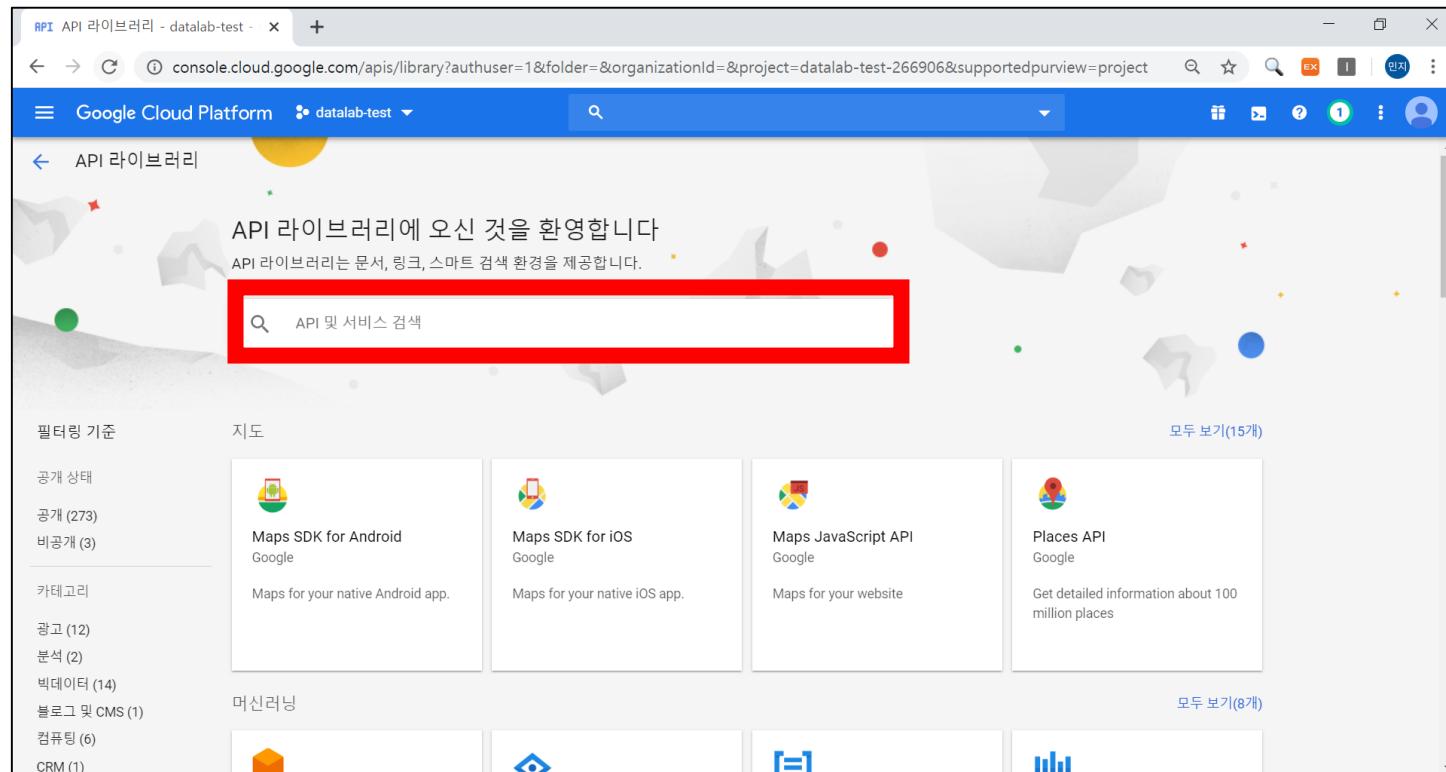
# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 1) Google Cloud에서 제공하는 Datalab 사용

Step 2. Google Compute Engine 및 Cloud Source Repositories 필요한 API 사용 설정

(2) 검색창에 Compute Engine 입력 → compute Engine API 클릭 → 사용설정

(3) 검색창에 Cloud Source Repositories 입력 → Cloud Source Repositories API 클릭 → 사용설정

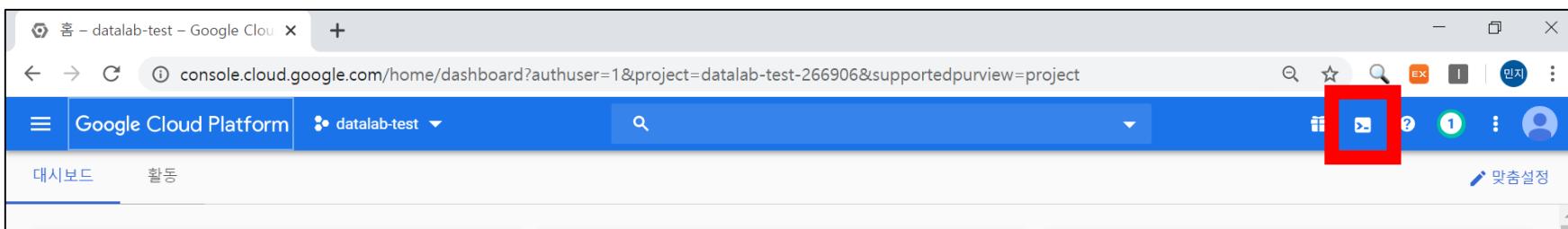


# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 1) Google Cloud에서 제공하는 Datalab 사용

## Step 3. Cloud Datalab VM 인스턴스 생성

- (1) 콘솔로 이동 → 우측 상단 왼쪽에서 두번째 버튼 클릭하여 Cloud Shell 실행



> 터미널에 아래의 코드(\$ 이후) 입력

```
$ datalab create datalab-instance
```

datalab-instance 라는 이름의 datalab VM 인스턴스 생성

- 기본적으로 제공되는 머신은 n1-standard-1 (vCPU 1개, 메모리 3.75GB)

※ 만약 위의 기본 제공 머신보다 좋은 성능의 VM 인스턴스를 만들고 싶다면

Google Cloud 머신유형 (<https://cloud.google.com/compute/docs/machine-types>)을 참고하여 아래의 코드 실행

```
$ datalab create --machine-type n1-standard-8(원하는 머신유형으로 입력하기) datalab-instance
```

머신유형이 n1-standard-8(vCPU 8개, 메모리 30GB)인 datalab-instance 라는 이름의 datalab VM 인스턴스 생성

# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 1) Google Cloud에서 제공하는 Datalab 사용

Step 3. Cloud Datalab VM 인스턴스 생성

- (2) 원하는 리전과 영역 선택

```
[50] asia-northeast3-a
[51] asia-northeast3-b
[52] asia-northeast3-c
[54] europe-north1-b
[55] europe-north1-c
[56] europe-west6-a
[57] europe-west6-b
[58] europe-west6-c
[59] northamerica-northeast1-a
[60] northamerica-northeast1-b
[61] northamerica-northeast1-c
[62] us-west2-a
[63] us-west2-b
[64] us-west2-c
Your selected zone 50 50 입력 (asia-northeast3-a 선택)
```

[ 실행 결과 ]

※ 사용 가능한 리전과 영역에 대한 설명

- <https://cloud.google.com/compute/docs/regions-zones/?hl=ko#available>
- 리전에 따라 사용 가능한 머신이 다름
- 2020년 1월부터 서울 리전 생김

(서울 리전은 생긴 지 얼마 안 돼서 아직 제공 머신이 다양하지 않음. 몇몇 리전에서는 [일반용도, 메모리 최적화, 컴퓨팅 최적화] 등 원하는 목적에 따른 더 좋은 성능의 머신 사용이 가능하므로 꼭 서울 리전을 사용할 필요는 없음.  
그러나 가까울수록 네트워크 지연 시간이 적기 때문에 서울로 선택)

# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 1) Google Cloud에서 제공하는 Datalab 사용

Step 3. Cloud Datalab VM 인스턴스 생성

(3) SSH키 생성 및 메타데이터 수정

```
Your selected zone: 50
Creating the network datalab-network
Creating the firewall rule datalab-network-allow-ssh
Creating the disk datalab-instance-pd
Creating the repository datalab-notebooks
Creating the instance datalab-instance
Created [https://www.googleapis.com/compute/v1/projects/datalab-test-266906/zones/asia-northeast3-a/instances/datalab-instance].
Connecting to datalab-instance.
This will create an SSH tunnel and may prompt you to create an rsa key pair. To manage these keys, see https://cloud.google.com/compute/docs/tutorials/managing-ssh-keys
Waiting for Datalab to be reachable at http://localhost:8081/
This tool needs to create the directory [/home/g213jmj/.ssh] before
being able to generate SSH keys.

Do you want to continue (Y/n)? y■ y 입력
```

[ 실행 결과 ]

```
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase): 비밀번호 입력 (자판을 쳐도 화면에는 나오지 않으니 당황하지 말 것)
Enter same passphrase again: 비밀번호 확인 (위에서 입력한 비밀번호 입력)
```

[ 실행 결과 ]

# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 1) Google Cloud에서 제공하는 Datalab 사용

## Step 3. Cloud Datalab VM 인스턴스 생성

### (3) SSH키 생성 및 메타데이터 수정

```
Your identification has been saved in /home/g213jmj/.ssh/google_compute_engine.  
Your public key has been saved in /home/g213jmj/.ssh/google_compute_engine.pub.  
The key fingerprint is:  
SHA256:3jRCVkPGs1Y86Dnvtb9nZGv1FSisaOhzzxt1AaygYQY g213jmj@cs-6000-devshell-vm-3d6c8f80-1bf7-45c7-b1b0-1edbf1a9d33a  
The key's randomart image is:  
+---[RSA 2048]---+  
| E. =* . |  
| + .oo=o. |  
| o o.+o. .. |  
| . o*. o ... |  
| .. So+... . |  
| . + =o.o = |  
| . . o... ++ |  
| o . oo . o+ |  
| o o... .++ |  
+---[SHA256]---+  
Updating project ssh metadata...#Updated [https://www.googleapis.com/compute/v1/projects/datalab-test-266906].  
Updating project ssh metadata...done.  
Waiting for SSH key to propagate.  
Enter passphrase for key '/home/g213jmj/.ssh/google_compute_engine':
```

SSH키에 대한 비밀번호 설정 (원하는대로 입력) → 나중에 접속 시 필요하니 기억해두기

The connection to Datalab is now open and will remain until this command is killed.  
Click on the \*Web Preview\* (square button at top-right), select \*Change port > Port 8081\*, and start using Datalab. **VM 인스턴스 생성 완료**

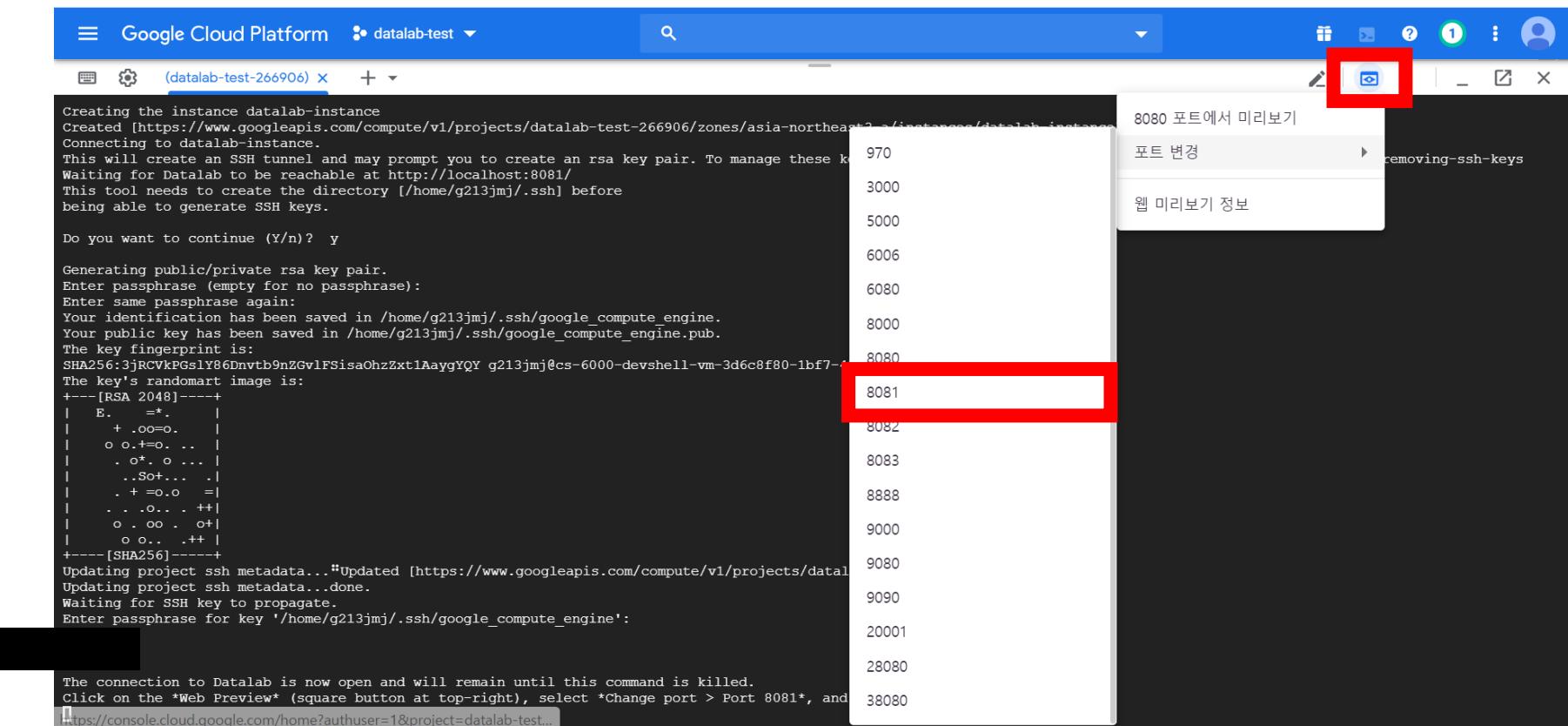
[ 실행 결과 ]

# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 1) Google Cloud에서 제공하는 Datalab 사용

Step 3. Cloud Datalab VM 인스턴스 생성

(4) 웹 미리보기 버튼을 클릭 - 포트변경 - 8081선택 → 브라우저에서 Cloud datalab 홈페이지 열기

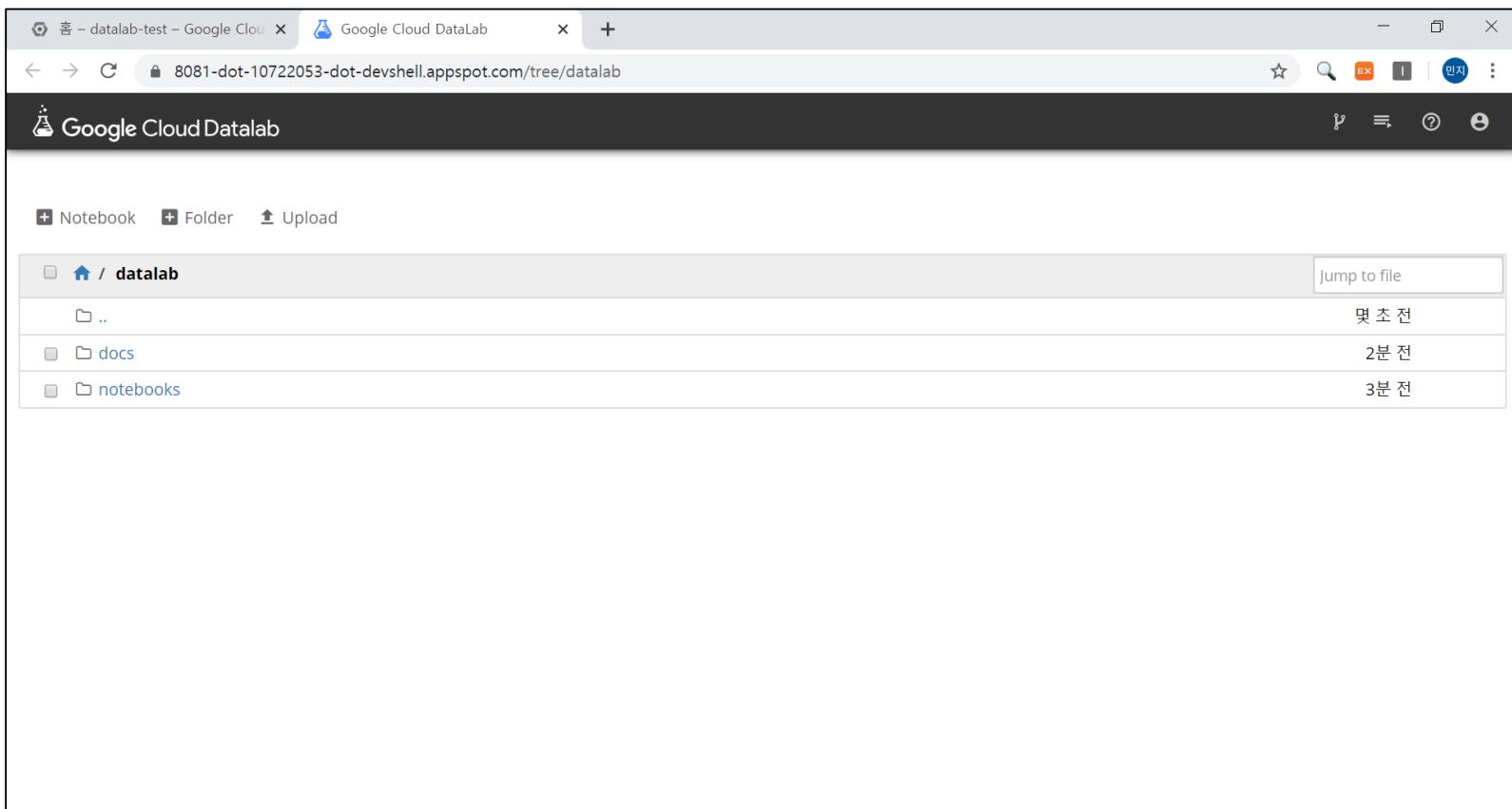


# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 1) Google Cloud에서 제공하는 Datalab 사용

Step 3. Cloud Datalab VM 인스턴스 생성

(5) 브라우저에서 Cloud datalab 홈페이지 열린 모습

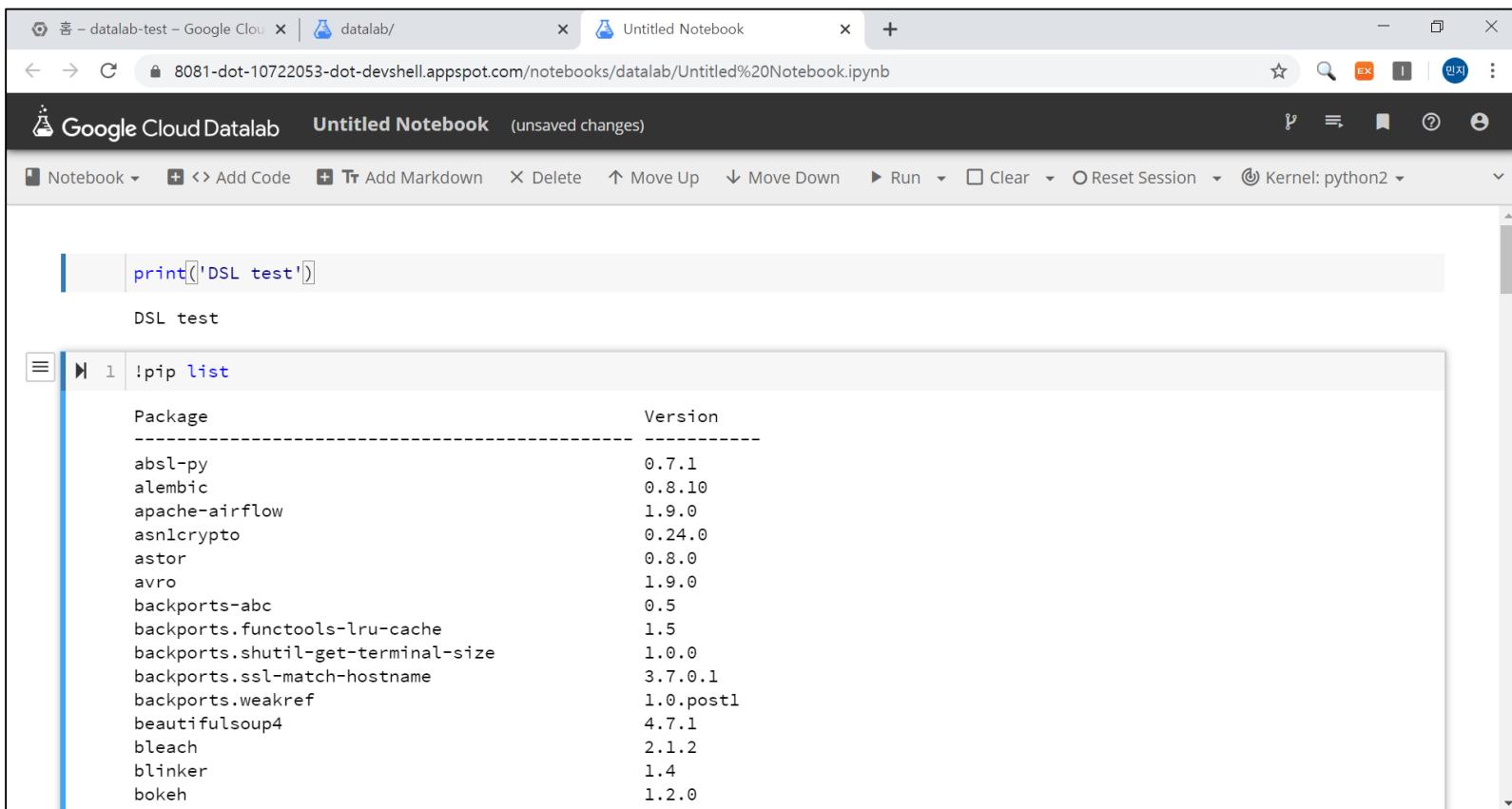


# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 1) Google Cloud에서 제공하는 Datalab 사용

Step 3. Cloud Datalab VM 인스턴스 생성

(6) 새로운 Notebook 파일 생성해서 이것저것 해보기



Package	Version
absl-py	0.7.1
alembic	0.8.10
apache-airflow	1.9.0
asn1crypto	0.24.0
astor	0.8.0
avro	1.9.0
backports-abc	0.5
backports.functools-lru-cache	1.5
backports.shutil-get-terminal-size	1.0.0
backports.ssl-match-hostname	3.7.0.1
backports.weakref	1.0.post1
beautifulsoup4	4.7.1
bleach	2.1.2
blinker	1.4
bokeh	1.2.0

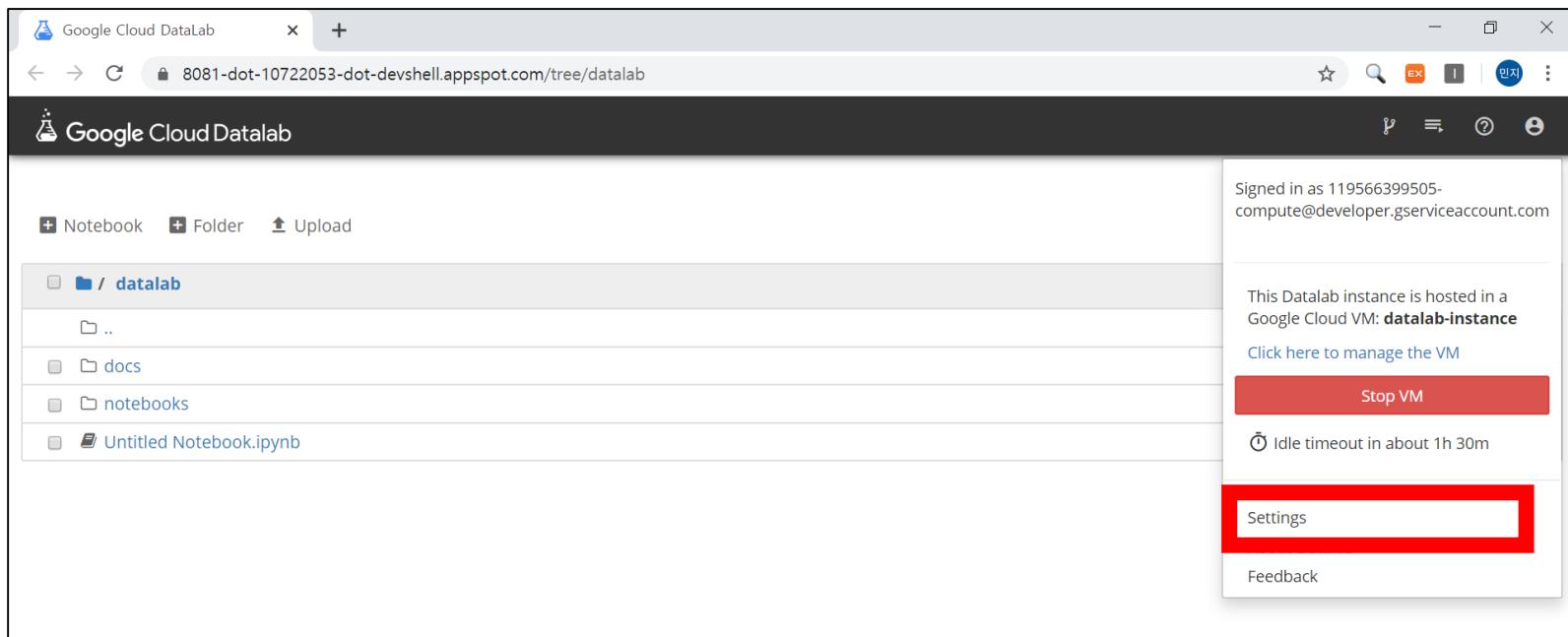
# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 1) Google Cloud에서 제공하는 Datalab 사용

## Step 3. Cloud Datalab VM 인스턴스 생성

### (7) 테마 및 Auto Shutdown 시간 조정

- 모델을 학습시키다가 꺼지는 불상사를 방지하기 위해 Auto Shutdown 시간을 조정 (default : 90m (90분) )
- 예시 : 90m, 1h 50m, 2d 12h ...



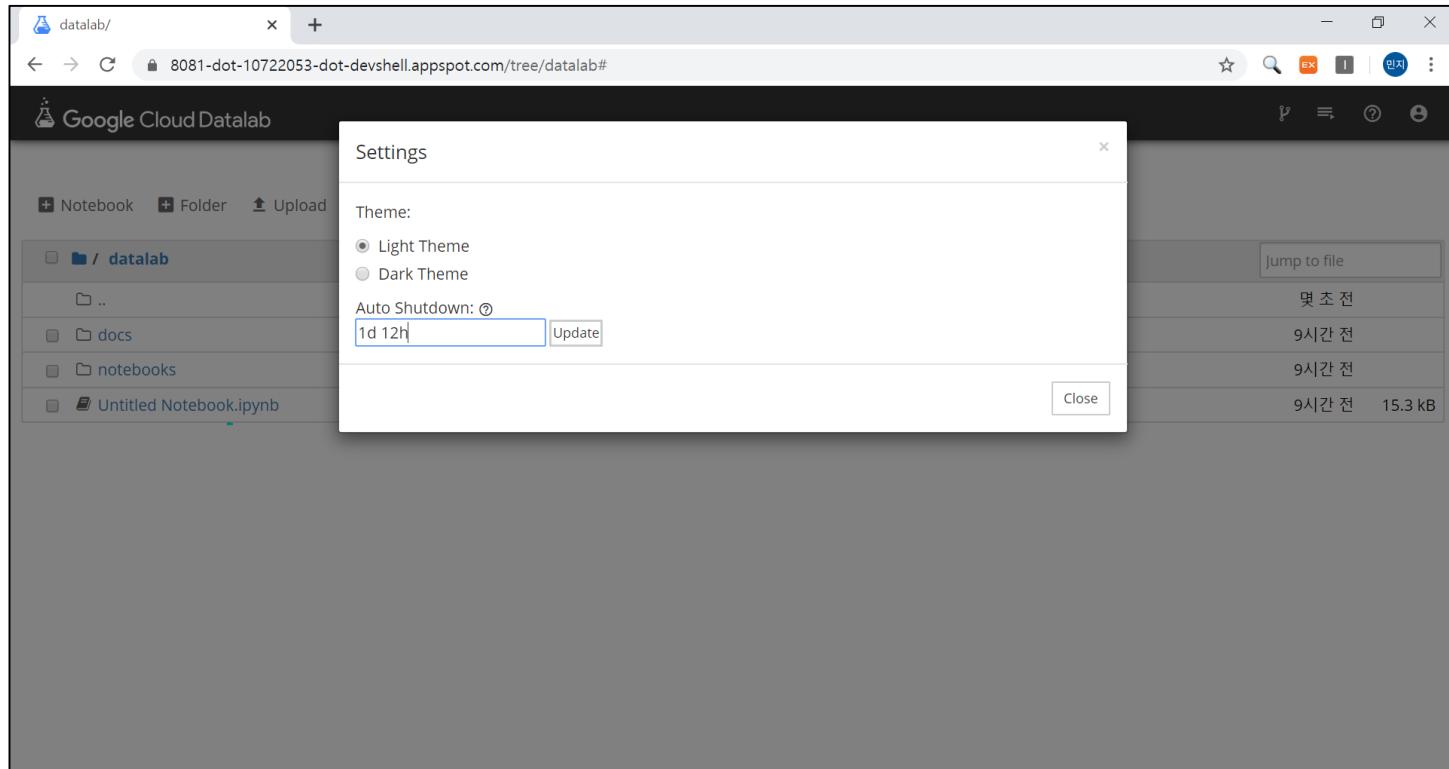
# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 1) Google Cloud에서 제공하는 Datalab 사용

## Step 3. Cloud Datalab VM 인스턴스 생성

### (7) 테마 및 Auto Shutdown 시간 조정

- 모델을 학습시키다가 꺼지는 불상사를 방지하기 위해 Auto Shutdown 시간을 조정 (default : 90m (90분) )
- 예시 : 90m, 1h 50m, 2d 12h ...



# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 1) Google Cloud에서 제공하는 Datalab 사용

## Step 3. Cloud Datalab VM 인스턴스 생성

### (8) 인스턴스 연결 종료하기

> 터미널에 아래의 코드(\$ 이후) 입력

```
$ datalab stop datalab-instance      datalab-instance라는 이름의 datalab VM 인스턴스 종료
```

```
g213jmj@cloudshell:~ (datalab-test-266906)$ datalab stop datalab-instance
Stopping datalab-instance
Stopping instance(s) datalab-instance...done.
Updated [https://compute.googleapis.com/compute/v1/projects/datalab-test-266906/zones/asia-northeast3-a/instances/datalab-instance].
g213jmj@cloudshell:~ (datalab-test-266906)$ █
```

인스턴스 종료됨

[ 실행 결과 ]

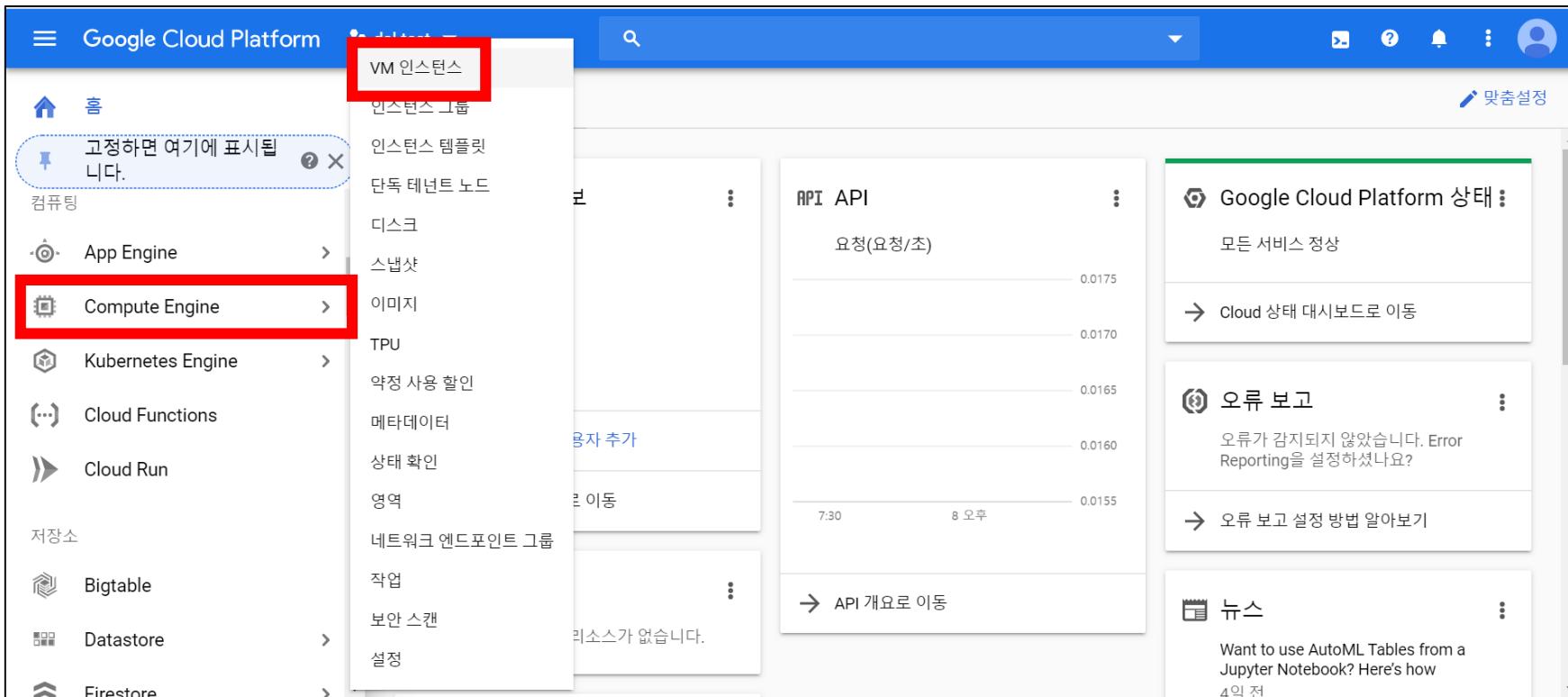
- 인스턴스를 켜놓으면 계속 비용이 발생하므로 사용하지 않을 때는 인스턴스를 반드시 중지해야함

# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 1) Google Cloud에서 제공하는 Datalab 사용

Step 4. Cloud Datalab VM 인스턴스 설정 바꾸기 (필요시)

(1) 컴퓨팅 → Compute Engine → VM 인스턴스 클릭

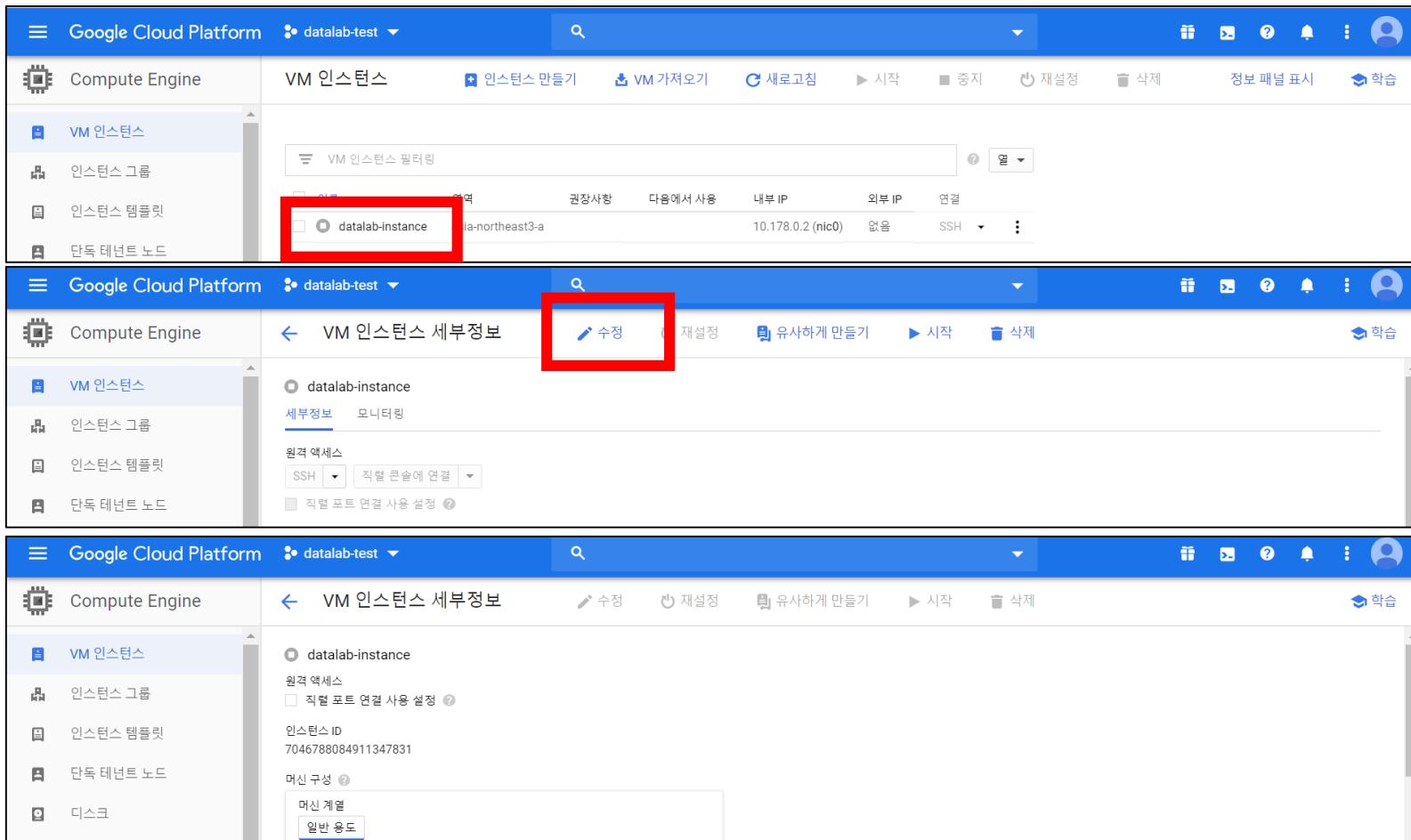


# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 1) Google Cloud에서 제공하는 Datalab 사용

Step 4. Cloud Datalab VM 인스턴스 설정 바꾸기 (필요시)

(2) 인스턴스 이름 클릭 → 수정 클릭 → 원하는 대로 설정 바꾸기(머신유형 등)



# 03. GCP에서 데이터 분석 환경 구축하기

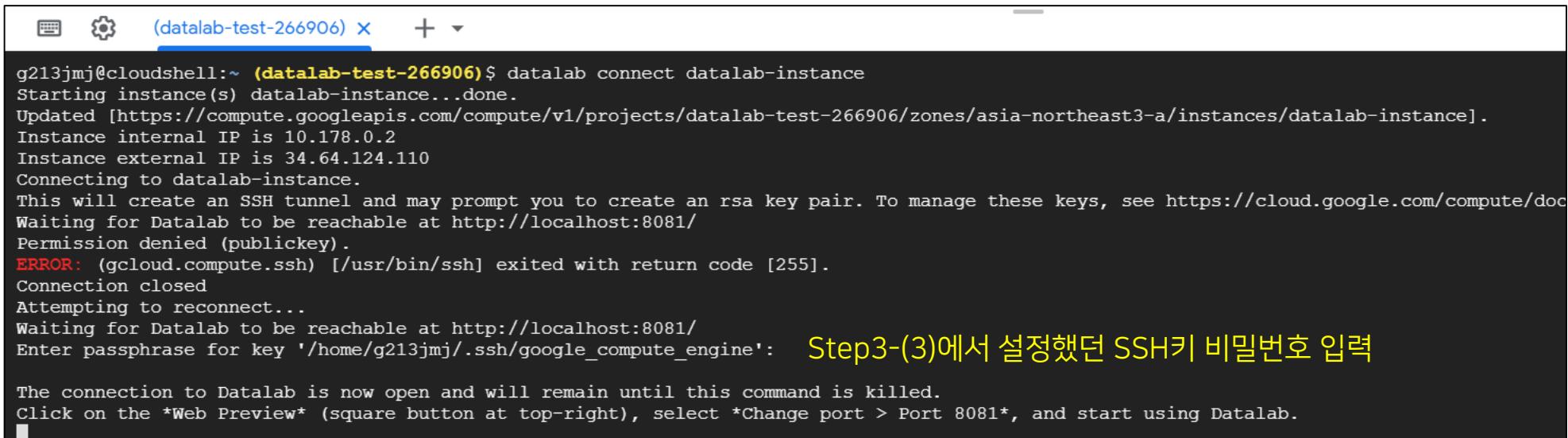
- 방법 1) Google Cloud에서 제공하는 Datalab 사용

Step 5. Cloud Datalab VM 인스턴스에 다시 연결

(1) 인스턴스에 다시 연결하기

> 터미널에 아래의 코드(\$ 이후) 입력

```
$ datalab connect datalab-instance      datalab-instance라는 이름의 datalab VM 인스턴스에 연결
```



```
g213jmq@cloudshell:~ (datalab-test-266906)$ datalab connect datalab-instance
Starting instance(s) datalab-instance...done.
Updated [https://compute.googleapis.com/compute/v1/projects/datalab-test-266906/zones/asia-northeast3-a/instances/datalab-instance].
Instance internal IP is 10.178.0.2
Instance external IP is 34.64.124.110
Connecting to datalab-instance.
This will create an SSH tunnel and may prompt you to create an rsa key pair. To manage these keys, see https://cloud.google.com/compute/doc
Waiting for Datalab to be reachable at http://localhost:8081/
Permission denied (publickey).
ERROR: (gcloud.compute.ssh) [/usr/bin/ssh] exited with return code [255].
Connection closed
Attempting to reconnect...
Waiting for Datalab to be reachable at http://localhost:8081/
Enter passphrase for key '/home/g213jmq/.ssh/google_compute_engine': Step3-(3)에서 설정했던 SSH키 비밀번호 입력

The connection to Datalab is now open and will remain until this command is killed.
Click on the *Web Preview* (square button at top-right), select *Change port > Port 8081*, and start using Datalab.
```

[ 실행 결과 ]

## 03. GCP에서 데이터 분석 환경 구축하기

- 방법 2) 가상머신에 직접 Anaconda 설치해서 사용

방법 2) 가상머신에 직접 Anaconda 설치해서 사용

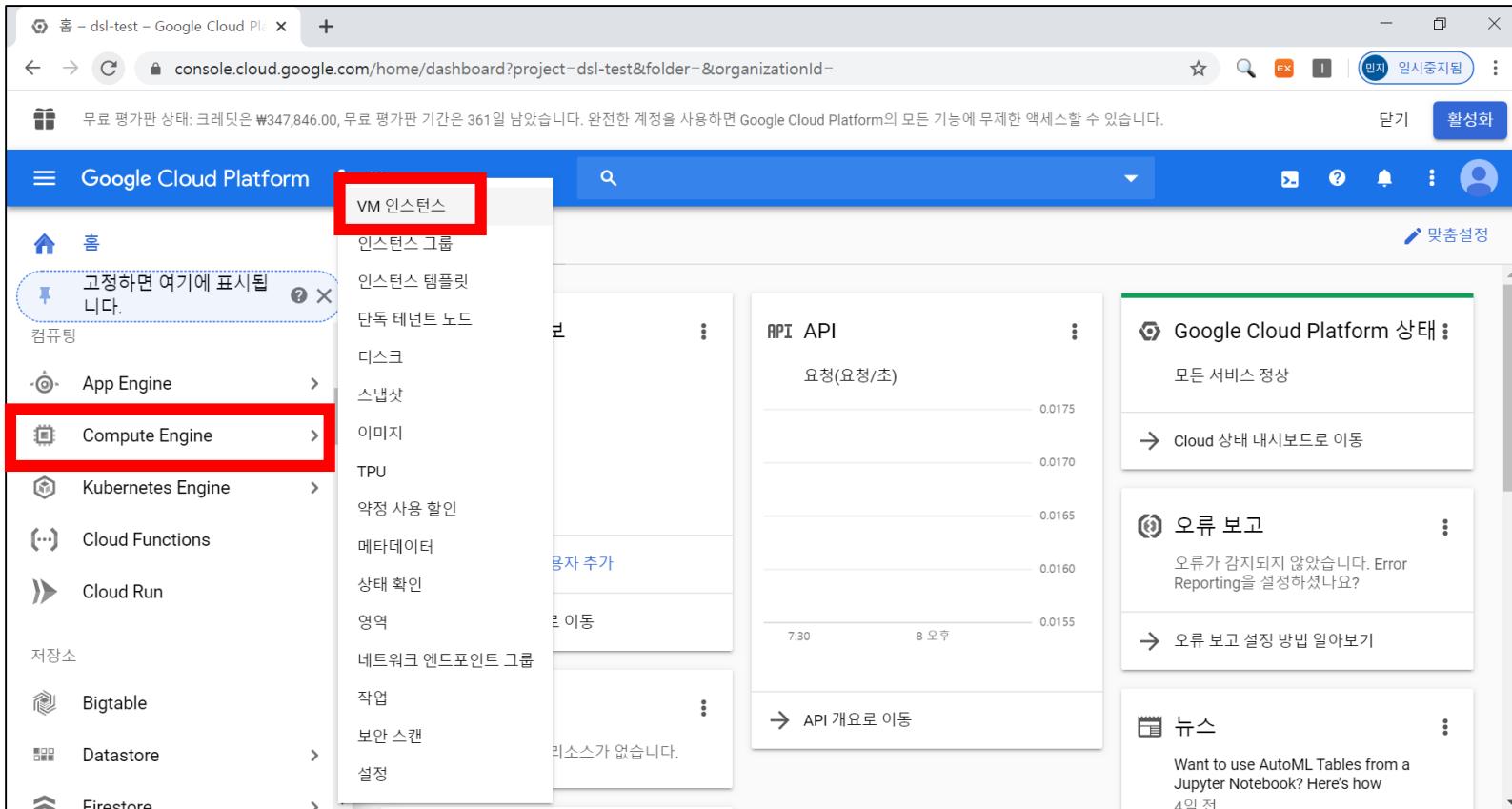
# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 2) 가상머신에 직접 Anaconda 설치해서 사용

Step 1. 프로젝트 생성 ← 방법 1)의 Step 1 참고

Step 2. VM 인스턴스 생성

(1) 컴퓨팅 → Compute Engine → VM 인스턴스 클릭

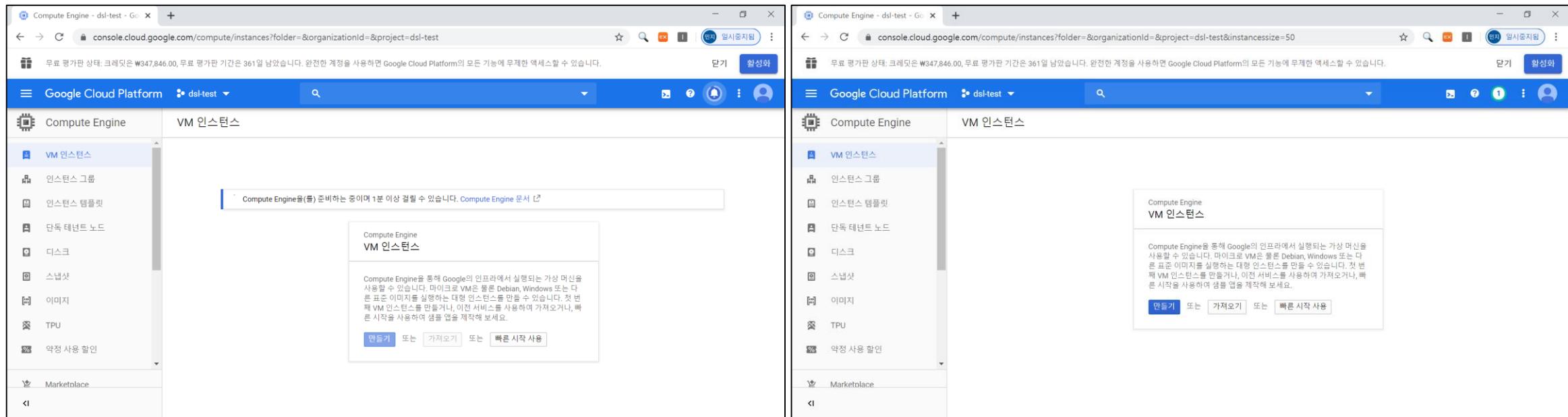


# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 2) 가상머신에 직접 Anaconda 설치해서 사용

## Step 2. VM 인스턴스 생성

### (2) VM 인스턴스 만들기

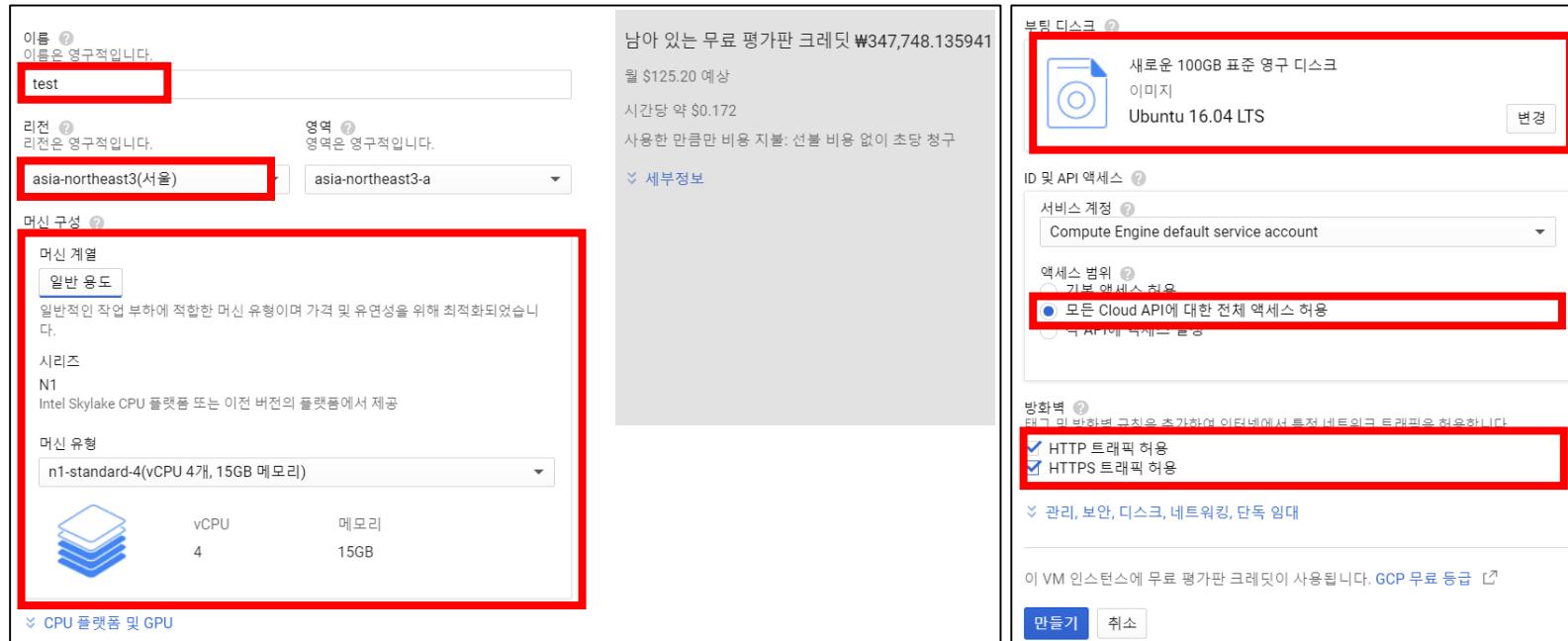


# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 2) 가상머신에 직접 Anaconda 설치해서 사용

## Step 2. VM 인스턴스 생성

### (3) VM 인스턴스 구성



※ 실습에서는

이름 : test

리전 : asia-northeast3(서울)

머신계열 : 일반용도

머신유형 : n1-standard-4

(vCPU 4개, 15GB 메모리)

부팅디스크 : Ubuntu 16.04 LTS (100GB)

IP 및 API 액세스 - 액세스 범위

: 모든 Cloud API에 대한 전체 액세스 허용

방화벽 : 둘다 허용

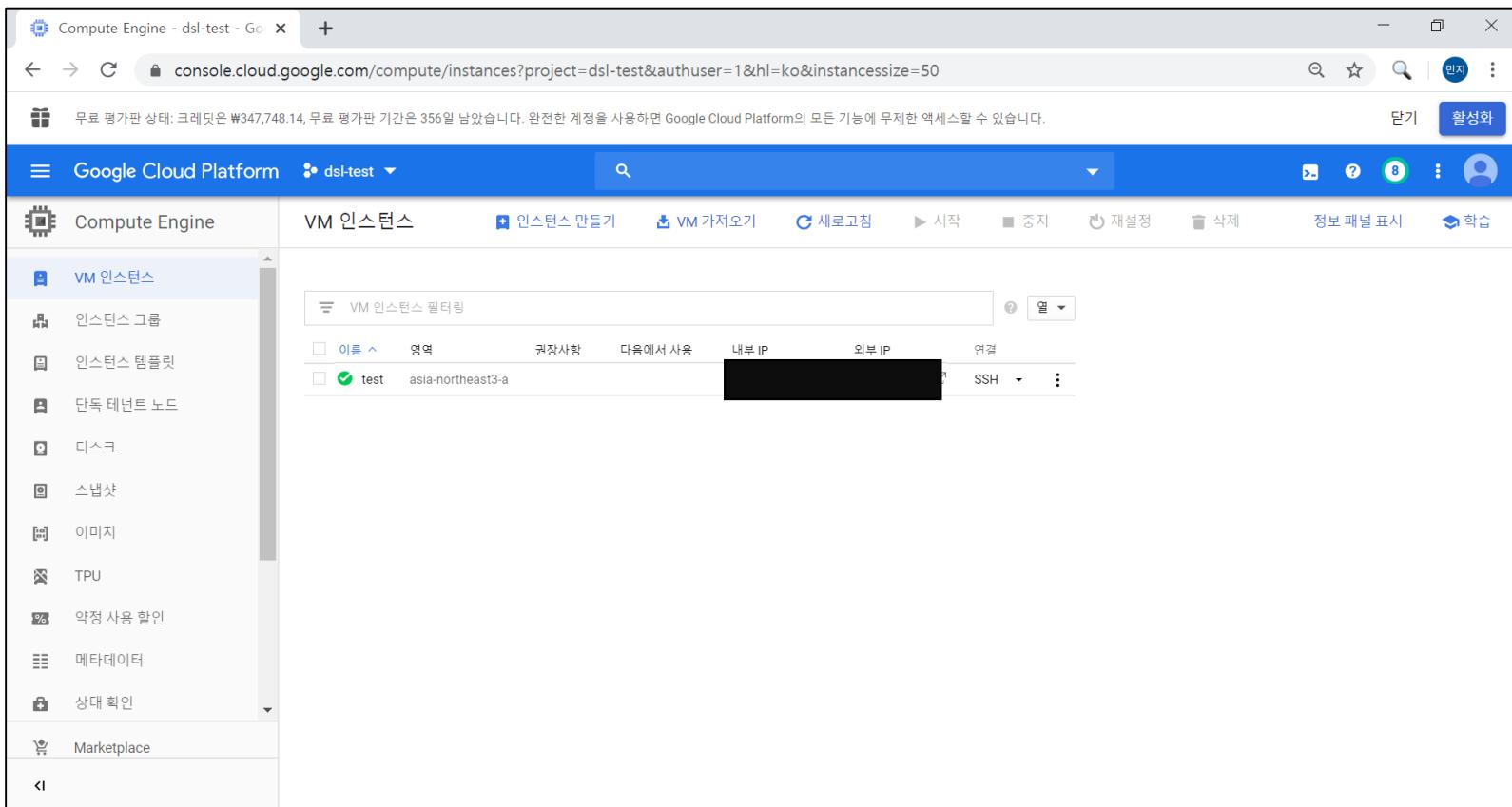
- 원하는 대로 VM 인스턴스 구성하기 (화면 우측(회색박스)에서 선택사항에 따라 달라지는 비용 확인 가능)
- 머신 구성은 추후 사용량에 따라 맞춤 머신 사양을 추천해주므로 고민이 된다면 일단 아무거나 선택해도 상관없음

# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 2) 가상머신에 직접 Anaconda 설치해서 사용

Step 2. VM 인스턴스 생성

(4) VM 인스턴스 생성 완료



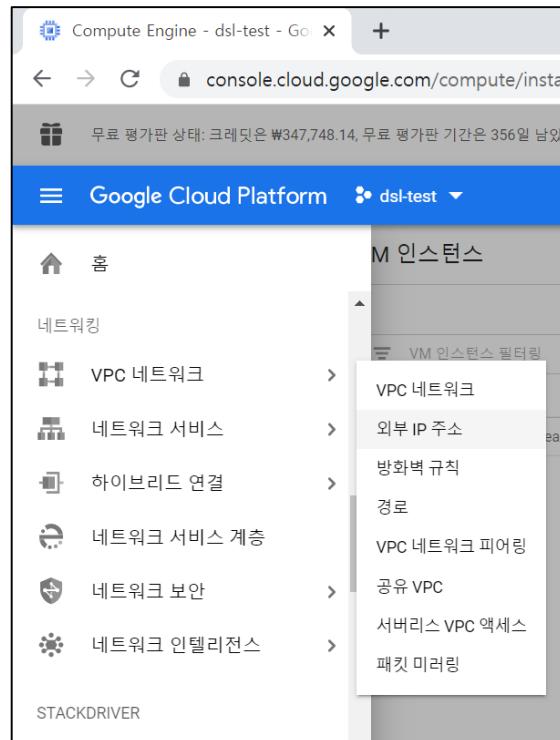
# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 2) 가상머신에 직접 Anaconda 설치해서 사용

## Step 3. 외부 IP 주소 고정

- 이 외부 IP를 고정해야 외부 접근이 가능  
( 고정된 외부 IP 주소를 가지지 않는 VM은 외부로부터의 접근을 차단하여 보안성이 높으나 매번 바뀌기 때문에 번거로우므로 외부 IP를 고정 )

(1) 좌측탭에서 네트워킹 - VPC 네트워크 - 외부 IP 주소로 이동



(2) 유형을 임시→고정으로 바꾸기

The top screenshot shows the 'External IP Address' page with a dropdown menu for 'Type' currently set to 'Temporary'. The bottom screenshot shows the same page with the 'Type' dropdown set to 'Fixed'. Both screenshots show a table with one row: 'asia-northeast3' (IP address blacked out), 'IPv4', and 'VM instance test (asia-northeast3-a)'.

이름	외부 주소	리전	유형	버전	다음에서 사용 중	네트워크 계층	라벨
		asia-northeast3	임시	IPv4	VM 인스턴스 test (asia-northeast3-a 영역)		

위의 검은색 부분이 본인 인스턴스에 접근할 수 있는 외부 IP 주소  
( 다른 사람에게 노출되지 않도록 할 것 )

# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 2) 가상머신에 직접 Anaconda 설치해서 사용

## Step 4. 방화벽 설정

- 외부에서 접근하기 위해 방화벽에서 포트를 열어줘야함

(1) 좌측탭에서 네트워킹 - VPC 네트워크 - 방화벽 규칙 으로 이동 ➔ 방화벽 규칙 만들기 클릭

The screenshot shows the Google Cloud Platform interface for managing VPC network rules. The left sidebar has 'VPC 네트워크' selected under '방화벽 규칙'. The main content area displays a table of existing firewall rules, with the top row showing a 'Create Firewall Rule' button highlighted by a red box.

이름	유형	대상	프로토콜 / 포트	작업	우선순위	네트워크
default-allow-http	수신	http-server	tcp:80	허용	1000	default
default-allow-https	수신	https-server	tcp:443	허용	1000	default
default-allow-icmp	수신	전체 적용	icmp	허용	65534	default
default-allow-internal	수신	전체 적용	tcp:0-65535 udp:0-65535 icmp	허용	65534	default
default-allow-rdp	수신	전체 적용	tcp:3389	허용	65534	default
default-allow-ssh	수신	전체 적용	tcp:22	허용	65534	default

# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 2) 가상머신에 직접 Anaconda 설치해서 사용

## Step 4. 방화벽 설정

### (2) 새로운 방화벽 규칙 설정

The image shows two side-by-side screenshots of the Google Cloud Platform Firewall Rule creation interface.

**Left Screenshot (Left Panel):**

- Name:** test-jupyter (highlighted by a red box)
- Description:** 선택사항 (선택 사항)
- Logs:** 사용 안함 (selected)
- Network:** default
- Priority:** 1000
- Direction:** 수신 (selected)
- Time-based Schedule:** 허용 (selected)

**Right Screenshot (Right Panel):**

- Target:** 네트워크의 모든 인스턴스 (highlighted by a red box)
- Source Filter:** IP 범위
- Source IP Range:** 0.0.0.0/0 (highlighted by a red box)
- Source IP Range (Advanced):** 없음
- Protocol and Port:** 모두 허용 (selected) (highlighted by a red box)
- Protocol and Port (Advanced):** 시설된 프로토콜 및 포트
- Action:** 만들기 (Create) (highlighted in blue)

**Summary (Bottom):**

- 이름 :** test-jupyter
- 대상 :** 네트워크의 모든 인스턴스
- 소스IP범위 :** 0.0.0.0/0
- 프로토콜 및 포트 :** 모두 허용

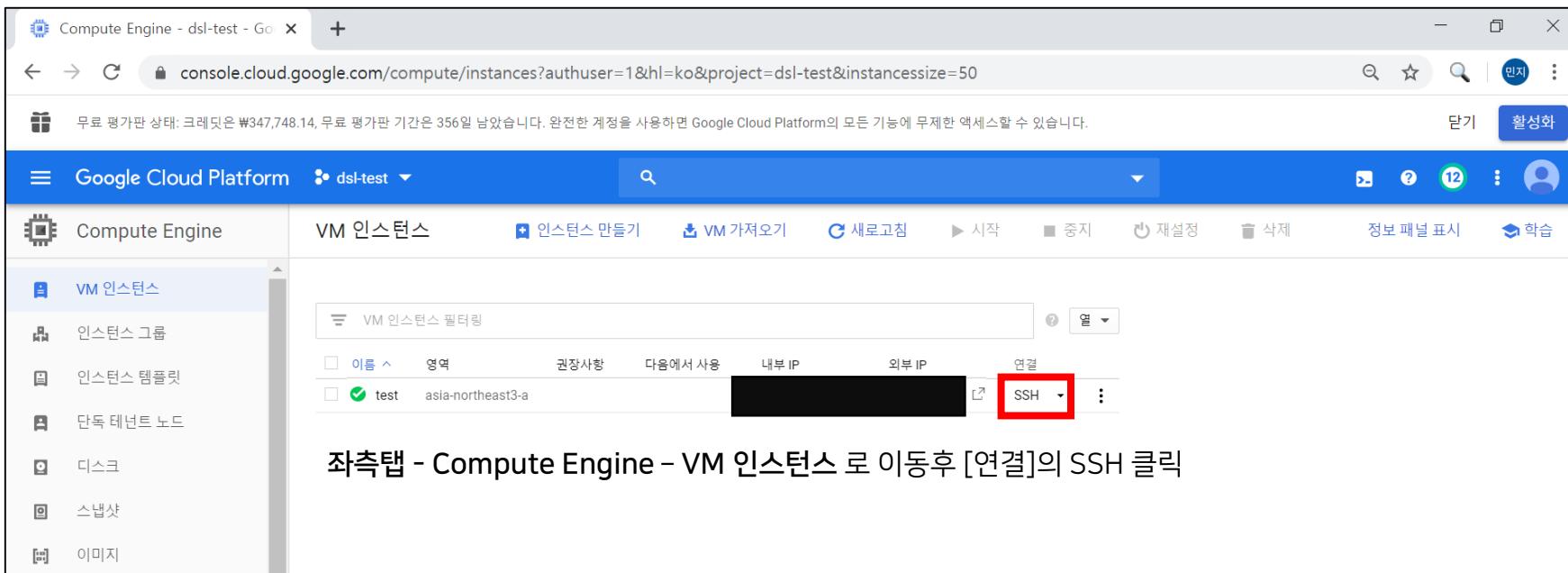
# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 2) 가상머신에 직접 Anaconda 설치해서 사용

## Step 5. Anaconda3 설치

### (1) SSH 터미널 오픈

Putty나 mobaXterm 등의 프로그램을 사용하여 가상머신에 접근할 수도 있으나 구글에서는 별도의 SSH 터미널 없이도 웹 브라우저 상에서 SSH 터미널을 오픈해줌



# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 2) 가상머신에 직접 Anaconda 설치해서 사용

Step 5. Anaconda3 설치

(2) VM 인스턴스의 터미널에 접속한 모습

The screenshot shows a terminal window titled "g213jmj@test: ~ - Chrome". The URL in the address bar is "ssh.cloud.google.com/projects/dsl-test/zones/asia-northeast3-a/instances/test?authuser=1&hl=ko&projectNumber=1020435037078". The terminal output includes:

```
Connected, host fingerprint: ssh-rsa 0 9E:6D:DB:A3:B3:A9:47:A3:2C:E9:AE:D1:44:B4  
:32:A0:62:86:A7:23:6A:13:35:80:D4:6A:56:67:20:7B:B9:C9  
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-1052-gcp x86_64)  
  
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/advantage  
  
0 packages can be updated.  
0 updates are security updates.  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/*copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
g213jmj@test:~$
```

# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 2) 가상머신에 직접 Anaconda 설치해서 사용

Step 5. Anaconda3 설치

(3) Anaconda3-5.2.0 설치

원하는 버전의 아나콘다 설치 (실습에서는 Anaconda3-5.2.0 버전 사용)

※ 다른 버전을 원한다면 <https://repo.continuum.io/archive/> 에서 원하는 버전의 파일 찾아서 사용

※ python3은 Anaconda3, python2는 Anaconda2 사용

※ 가상머신은 linux환경이므로 linux 버전 다운로드

> 터미널에 아래의 코드들(\$ 이후) 차례로 입력

```
$ mkdir tmp      설치 파일 받기 위한 임시 폴더 생성
$ cd tmp        생성한 폴더(tmp)로 이동
$ wget https://repo.anaconda.com/archive>Anaconda3-5.2.0-Linux-x86\_64.sh      Anaconda3-5.2.0 설치 파일 다운로드
```

```
q213jmj@test: ~/tmp - Chrome
ssh.cloud.google.com/projects/dsl-test/zones/asia-northeast3-a/instances/test?authuser=1&hl=ko&project...
g213jmj@test:~$ mkdir tmp
g213jmj@test:~$ cd tmp
g213jmj@test:~/tmp$ wget https://repo.anaconda.com/archive/Anaconda3-5.2.0-Linux-x86_64.sh
--2020-01-27 10:42:07-- https://repo.anaconda.com/archive/Anaconda3-5.2.0-Linux-x86_64.sh
Resolving repo.anaconda.com (repo.anaconda.com)... 104.16.131.3, 104.16.130.3, 2606:4700::6810:8303, ...
Connecting to repo.anaconda.com (repo.anaconda.com)|104.16.131.3|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 651745206 (622M) [application/x-sh]
Saving to: 'Anaconda3-5.2.0-Linux-x86_64.sh'

Anaconda3-5.2.0-Linux-x 100%[=====] 621.55M 78.2MB/s   in 8.2s

2020-01-27 10:42:16 (76.1 MB/s) - 'Anaconda3-5.2.0-Linux-x86_64.sh' saved [651745206/651745206]
```

< 실행 결과 >

# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 2) 가상머신에 직접 Anaconda 설치해서 사용

Step 5. Anaconda3 설치

> 터미널에 아래의 코드(\$ 이후) 입력

```
$ bash Anaconda3-5.2.0-Linux-x86_64.sh
```

설치 파일 실행

< 실행 결과 >

```
g213jmj@test:~/tmp$ bash Anaconda3-5.2.0-Linux-x86_64.sh
Welcome to Anaconda3 5.2.0

In order to continue the installation process, please review the license
agreement.
Please, press ENTER to continue
>>> █ Enter 키 쳐서 넘기기 → 계속 Enter키 치다보면 아래와 같은 질문 나옴

Do you accept the license terms? [yes|no]
[no] >>>
Please answer 'yes' or 'no':'
>>>
Please answer 'yes' or 'no':' yes 입력
>>> yes █

Anaconda3 will now be installed into this location:
/home/g213jmj/anaconda3

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below

[/home/g213jmj/anaconda3] >>> █ Enter키 치기 (설치 경로 바꾸고 싶으면 원하는 경로 입력)
```

# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 2) 가상머신에 직접 Anaconda 설치해서 사용

## Step 5. Anaconda3 설치

- 패키지들이 쭉 설치되다가 질문 나옴

< 실행 결과 >

```
installing: anaconda-5.2.0-py36_3 ...
installation finished.
Do you wish the installer to prepend the Anaconda3 install location
to PATH in your /home/g213jmj/.bashrc ? [yes|no]
[no] >>> yes■ yes 입력
Appending source /home/g213jmj/anaconda3/bin/activate to /home/g213jmj/.bashrc
A backup will be made to: /home/g213jmj/.bashrc-anaconda3.bak
```

For this change to become active, you have to open a new terminal.

Thank you for installing Anaconda3!

---

Anaconda is partnered with Microsoft! Microsoft VSCode is a streamlined code editor with support for development operations like debugging, task running and version control.

To install Visual Studio Code, you will need:

- Administrator Privileges
- Internet connectivity

Visual Studio Code License: <https://code.visualstudio.com/license>

Do you wish to proceed with the installation of Microsoft VSCode? [yes|no]
>>> no■ no 입력 (VSCode 설치하고 싶다면 yes 입력)

# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 2) 가상머신에 직접 Anaconda 설치해서 사용

Step 5. Anaconda3 설치

앞의 과정을 통해 설치가 완료된 후 아래의 코드 입력

```
$ source ~/.bashrc      source 명령을 사용하여 일반적으로 로그인할 때만 읽혀지는 .bashrc 파일을 Linux에서 강제로 재로드하도록 함
```

# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 2) 가상머신에 직접 Anaconda 설치해서 사용

Step 6. Jupyter Notebook 접속

(1) Jupyter Notebook 접속을 위한 비밀번호 생성

> 터미널에 차례로 코드 입력

```
$ python      파일 실행
>>> from notebook.auth import passwd    import 패키지
>>> passwd()      Jupyter notebook에 접속할 때 사용할 비밀번호 설정
>>> quit()      파일 종료
```

< 실행 결과 >

```
g213jmj@test:~/tmp$ python
Python 3.6.5 |Anaconda, Inc.| (default, Apr 29 2018, 16:14:56)
[GCC 7.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from notebook.auth import passwd
>>> passwd()
Enter password:  비밀번호 입력 (자판을 쳐도 화면에는 나오지 않으니 당황하지 말 것) ➔ 나중에 접속 시 필요하니 기억해두기
Verify password:  위에서 입력한 비밀번호 한번 더 치기
'sha1:e5936f3dafa9:48ed670210d20af5f3fba3e60657b80d231faad1'  본인이 입력한 비밀번호를 암호화한 것
>>> quit()
```

# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 2) 가상머신에 직접 Anaconda 설치해서 사용

Step 6. Jupyter Notebook 접속

(2) Jupyter Notebook 설정파일 생성

> 터미널에 코드 입력

```
$ jupyter notebook --generate-config      Jupyter Notebook 설정파일 생성
```

< 실행 결과 >

```
g213jmj@test:~/tmp$ jupyter notebook --generate-config
Writing default config to: /home/g213jmj/.jupyter/jupyter_notebook_config.py
```

# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 2) 가상머신에 직접 Anaconda 설치해서 사용

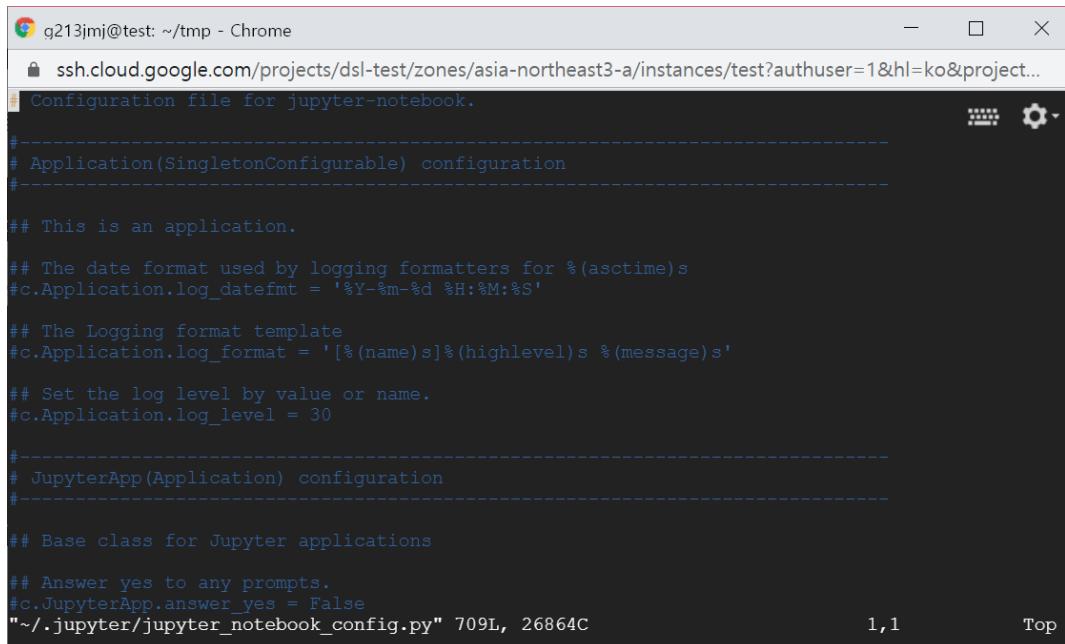
Step 6. Jupyter Notebook 접속

(3) Jupyter Notebook 설정파일 수정

> 터미널에 코드 입력

\$ vi ~/.jupyter/jupyter\_notebook\_config.py      Jupyter Notebook 설정파일 수정 (vi 에디터 사용)

< 실행 결과 >



```
q213jmj@test: ~/tmp - Chrome
ssh.cloud.google.com/projects/dsl-test/zones/asia-northeast3-a/instances/test?authuser=1&hl=ko&project...
# Configuration file for jupyter-notebook.

#-----
# Application(SingletonConfigurable) configuration
#-----

## This is an application.

## The date format used by logging formatters for %(asctime)s
#c.Application.log_datefmt = '%Y-%m-%d %H:%M:%S'

## The Logging format template
#c.Application.log_format = '[%(name)s]%(highlevel)s %(message)s'

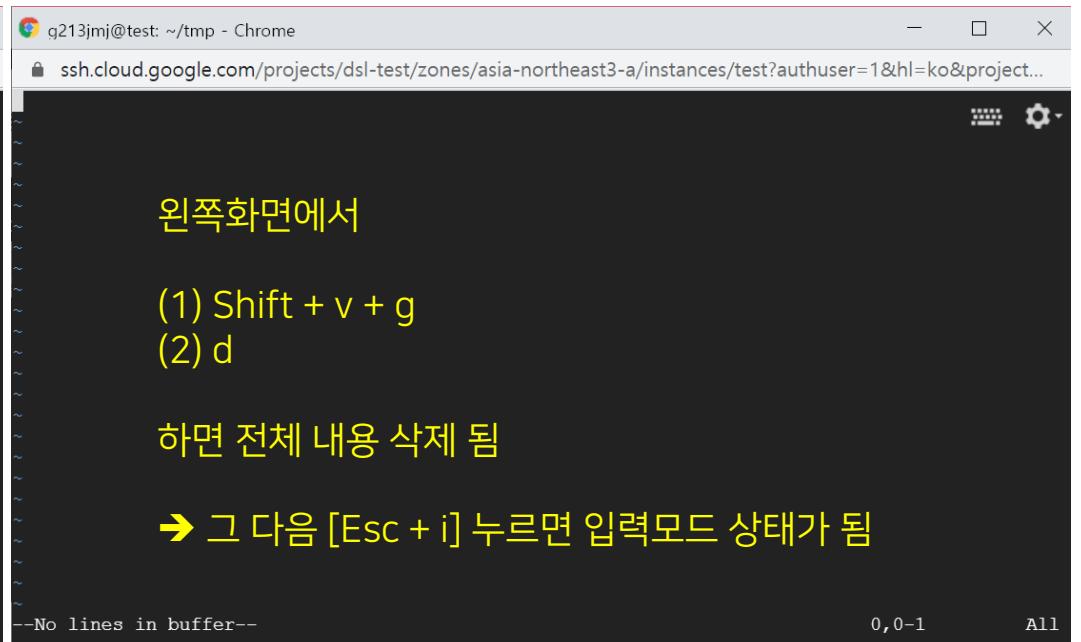
## Set the log level by value or name.
#c.Application.log_level = 30

#-----
# JupyterApp(Application) configuration
#-----


## Base class for Jupyter applications

## Answer yes to any prompts.
#c.JupyterApp.answer_yes = False
"~/jupyter/jupyter_notebook_config.py" 709L, 26864C
```

1,1                  Top



완쪽화면에서  
(1) Shift + v + g  
(2) d  
하면 전체 내용 삭제 됨  
→ 그 다음 [Esc + i] 누르면 입력모드 상태가 됨

0,0-1                  All

# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 2) 가상머신에 직접 Anaconda 설치해서 사용

Step 6. Jupyter Notebook 접속

(3) Jupyter Notebook 설정파일 수정

> 빈 설정파일에 아래 내용 입력

```
c.NotebookApp.ip = '*'          외부에서 접근가능한 ip주소를 모든 ip에서 접근가능하도록 설정
c.NotebookApp.open_browser = False 주피터 노트북을 실행할 때 자동으로 브라우저가 켜지지 않도록 설정
c.NotebookApp.port = 5000         포트 번호 설정
c.NotebookApp.password = u' Step4-1)에서 복사해두었던 암호화 된 비밀번호 '
# 예시
# c.NotebookApp.password = u'sha1:e5936f3dafa9:48ed670210d20af5f3fba3e60657b80d231faad1'
```

< 실행 결과 >

```
c.NotebookApp.ip = '*'
c.NotebookApp.open_browser = False
c.NotebookApp.port = 5000
c.NotebookApp.password = u'sha1:e5936f3dafa9:48ed670210d20af5f3fba3e60657b80d231faad1'
~
~
:wq
```

위의 내용 입력 후 [Esc + : + w + q + Enter] 눌러서 저장

# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 2) 가상머신에 직접 Anaconda 설치해서 사용

Step 6. Jupyter Notebook 접속

(4) Jupyter Notebook 접속

```
$ jupyter notebook
```

Jupyter Notebook 접속 (콘솔창 닫으면 접속 끊김)

# 참고

```
$ nohup jupyter notebook &
```

콘솔창을 닫더라도 웹에서 Jupyter Notebook 접속 가능하게 함

위의 검은색 부분 **외부 IP 주소**(#.#.#.# 형식) 복사

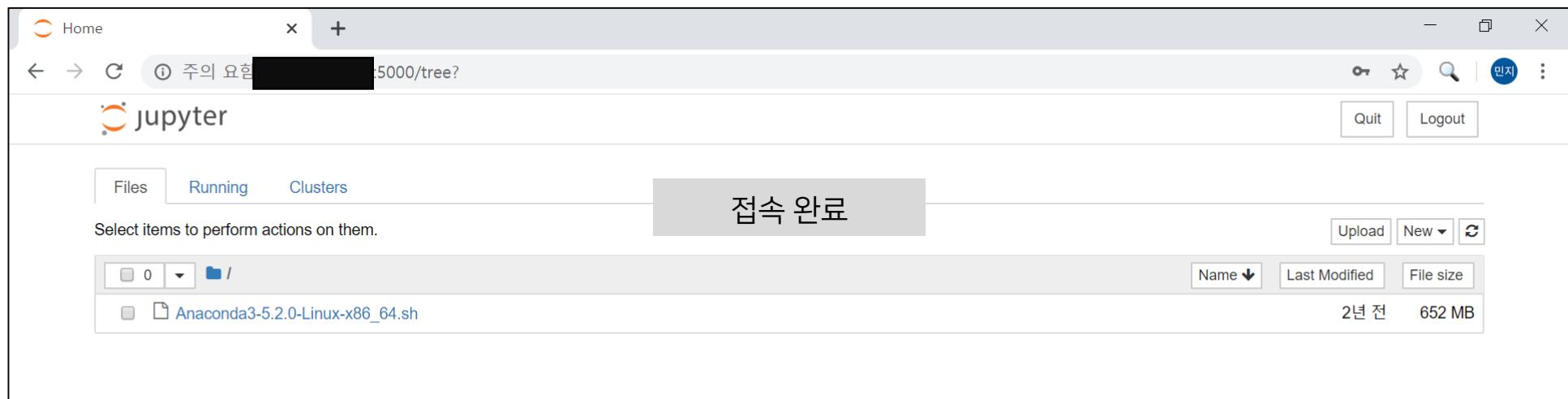
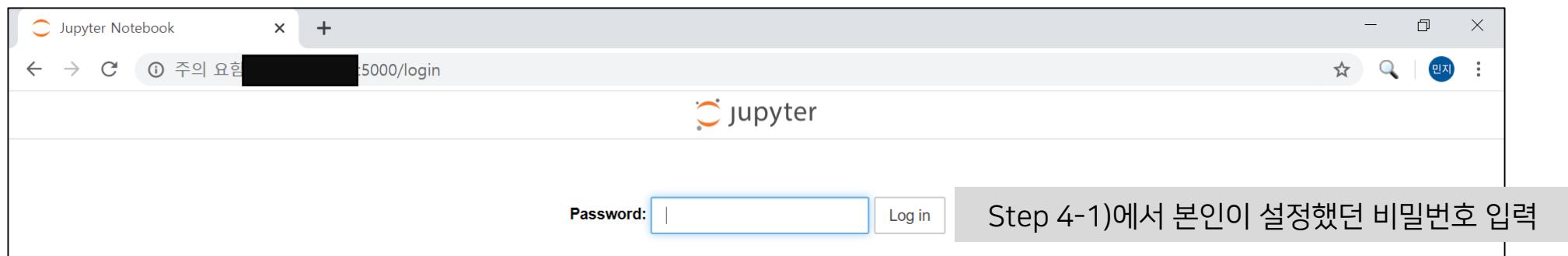
외부 IP 주소:포트번호 (Jupyter Notebook 설정파일에서 설정한 포트번호) 입력

# 03. GCP에서 데이터 분석 환경 구축하기

- 방법 2) 가상머신에 직접 Anaconda 설치해서 사용

Step 6. Jupyter Notebook 접속

(4) Jupyter Notebook 접속



# 03. GCP에서 데이터 분석 환경 구축하기

- 외부 프로그램 접속

## 외부 프로그램 접속

GCP 웹에서 SSH 접속(앞서 살펴본 것처럼)해도 되지만 외부 프로그램 접속할 때의 장점이 있다

※ 외부 프로그램으로 접속하면 좋은 점

- 연결이 빠름
- 파일 업로드/다운로드 하는 게 편함 (파일 여러개씩 올리거나 다운로드 받는 게 가능)  
→ 이게 왜 좋지?

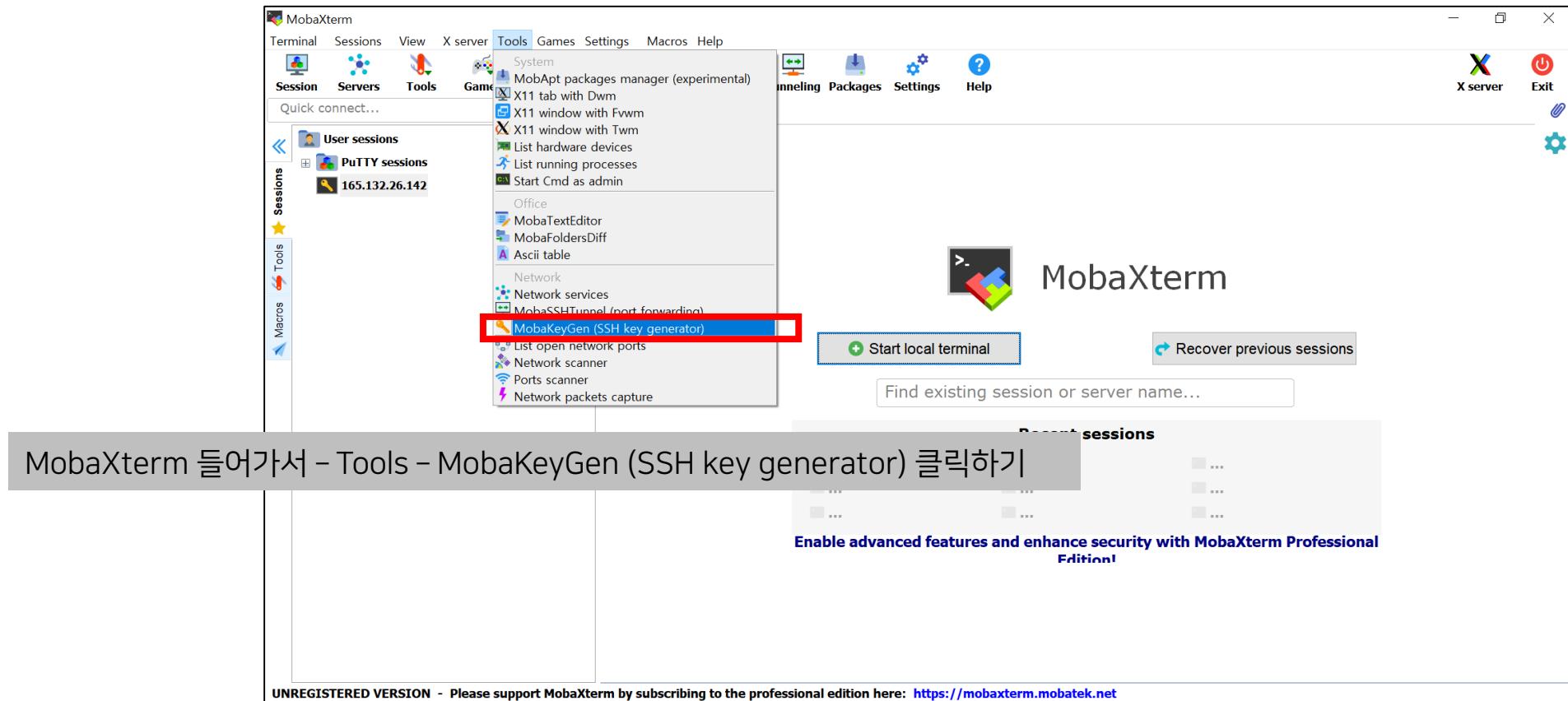
웹에서 접속하면 Jupyter notebook으로만 파일 업로드/다운로드가 가능한데,  
Jupyter notebook으로는 파일 하나씩 밖에 다운로드 못 받아서 속터짐

# 03. GCP에서 데이터 분석 환경 구축하기

- 외부 프로그램 접속

Step 0. MobaXterm 설치 (<https://mobaxterm.mobatek.net/>)

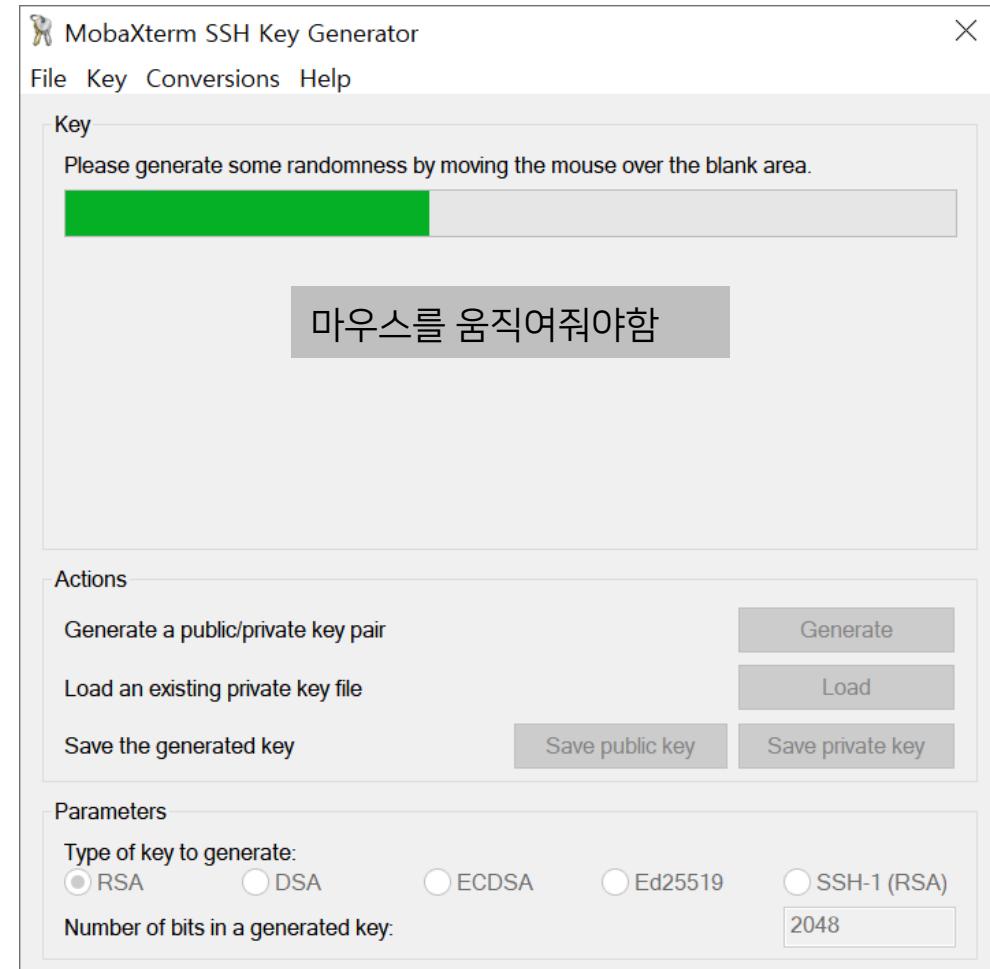
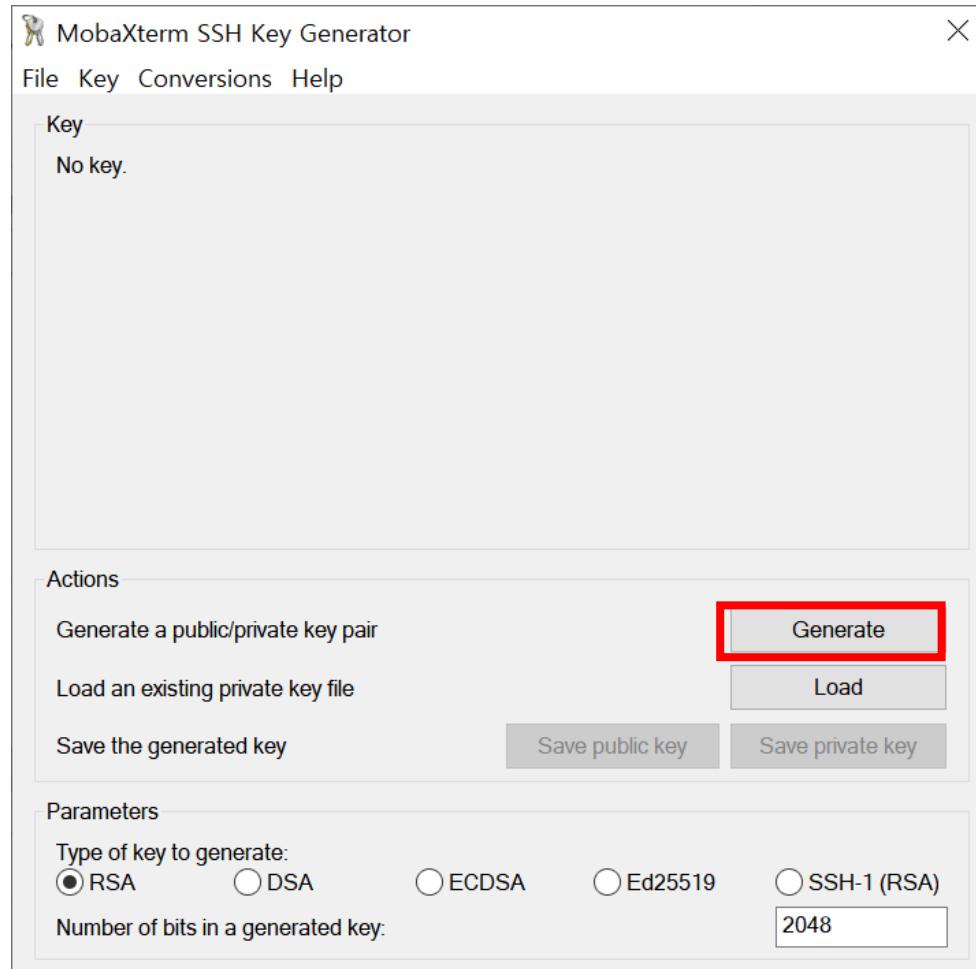
Step 1. SSH 키 생성



# 03. GCP에서 데이터 분석 환경 구축하기

- 외부 프로그램 접속

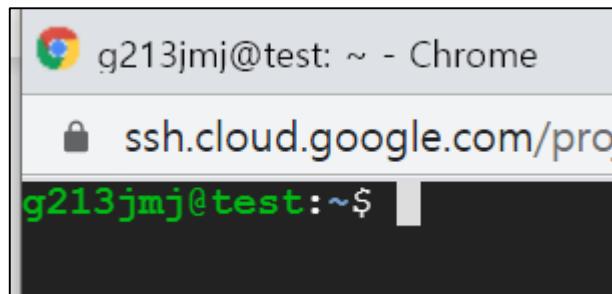
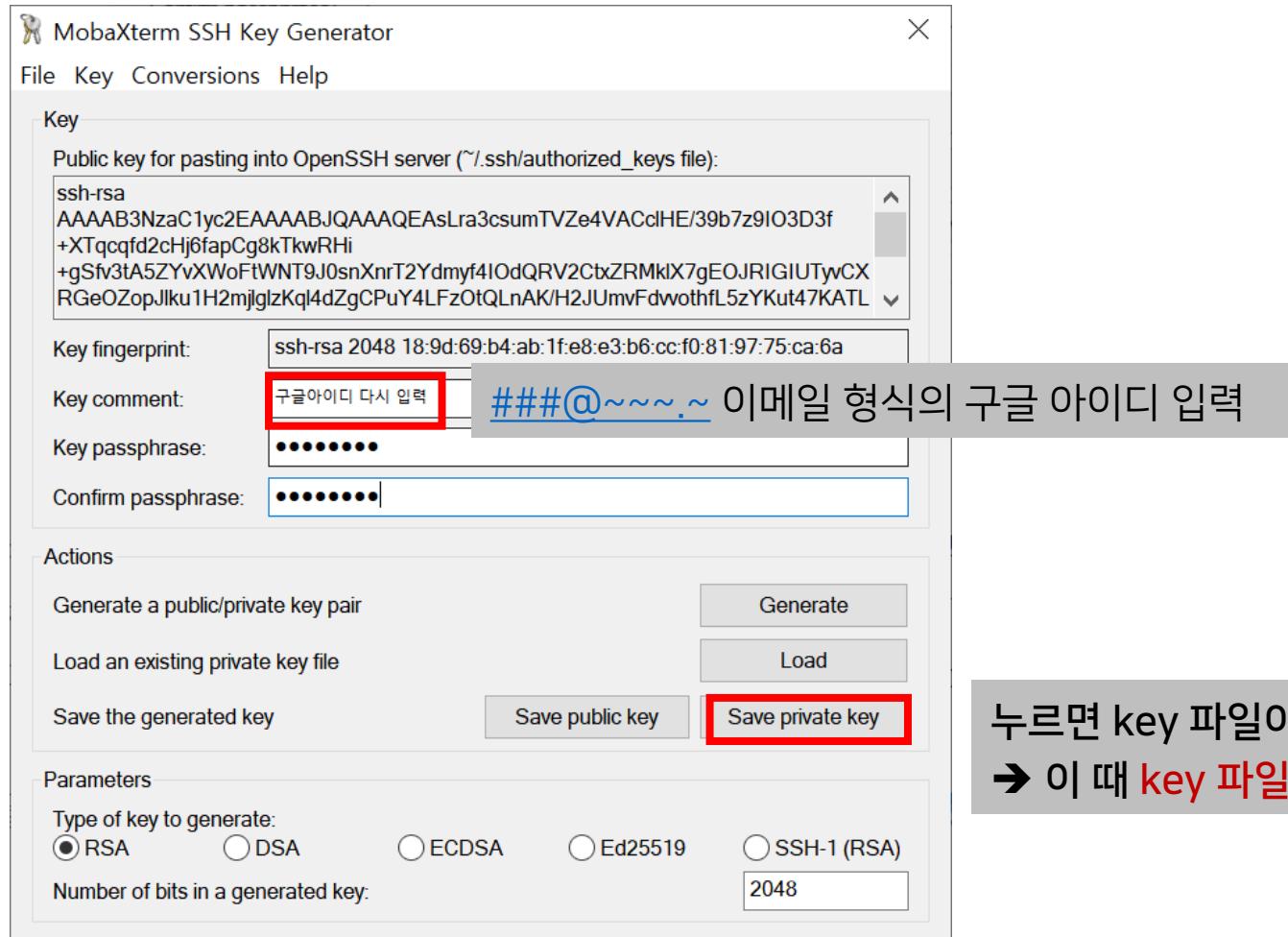
## Step 1. SSH 키 생성



# 03. GCP에서 데이터 분석 환경 구축하기

- 외부 프로그램 접속

## Step 1. SSH 키 생성



→ 이 때 주의해야할 것이 웹에서 SSH 접속했을 때  
@ 앞에 있는 부분(예:g213jmj)을 ### 부분에 써줘야함

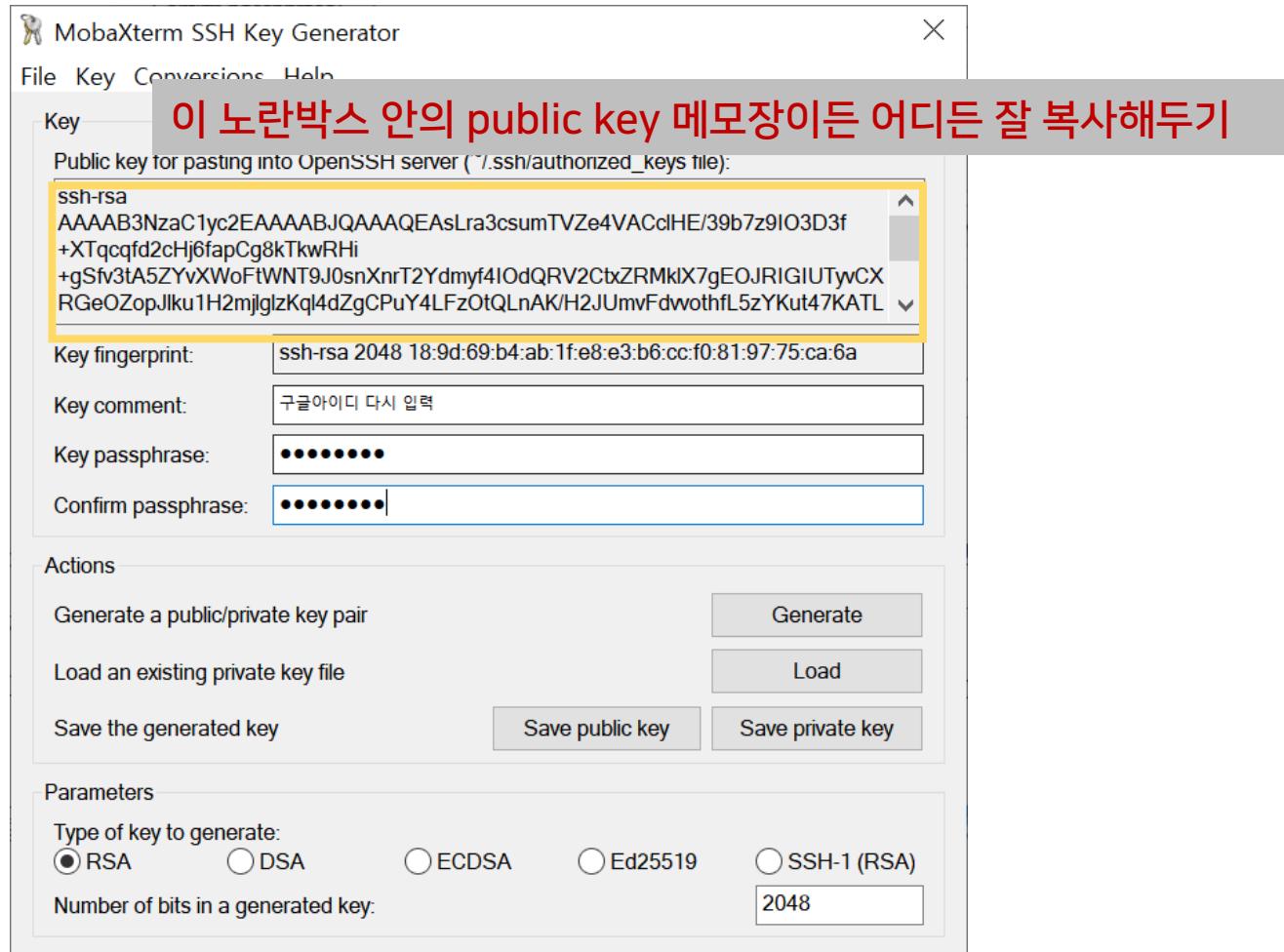
누르면 key 파일이 저장됨

→ 이 때 key 파일 저장 위치 잘 기억해두기

# 03. GCP에서 데이터 분석 환경 구축하기

- 외부 프로그램 접속

## Step 1. SSH 키 생성



# 03. GCP에서 데이터 분석 환경 구축하기

- 외부 프로그램 접속

Step 2. 메타데이터 추가하기

Compute Engine의 메타데이터로 들어가서 SSH키 클릭

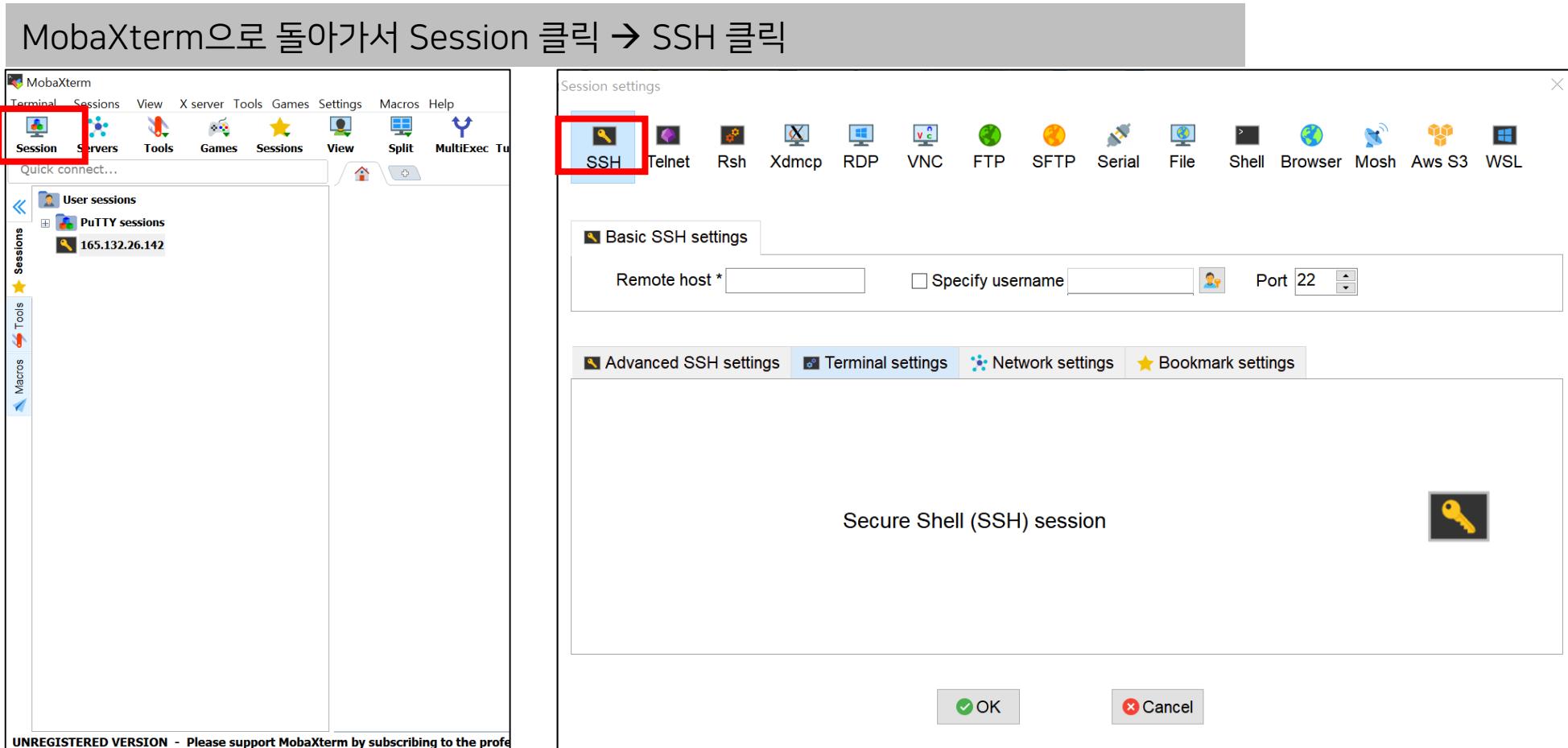
The screenshot shows the Google Cloud Platform interface for managing Compute Engine instances. On the left, a sidebar lists various resources: VM 인스턴스, 인스턴스 그룹, 인스턴스 템플릿, 단독 테넌트 노드, 디스크, 스냅샷, 이미지, TPU, 약정 사용 할인, 메타데이터, 상태 확인, and 영역. The '메타데이터' item is highlighted with a red box. The main pane shows the 'Metadatator' section for a specific instance, with tabs for 'Metadatator' and 'SSH 키'. The 'SSH 키' tab is also highlighted with a red box. A note below states: '이 프로젝트의 모든 인스턴스는 이러한 SSH 키를 상속받습니다. 자세히 알아보기'. Below this is a text input field labeled '공개 SSH 키 입력' with placeholder text '아까 메모장에 복사해 둔 publick key 입력'. To the right of the input field is a note: '아까 메모장에 복사해 둔 publick key 입력 이 때 key의 마지막 부분이 [###@gmail.com](#) 이라면 @gmail.com을 빼고 ###만 남길 것'. At the bottom are '저장' and '취소' buttons.

아까 메모장에 복사해 둔 publick key 입력  
이 때 key의 마지막 부분이 [###@gmail.com](#) 이  
라면 @gmail.com을 빼고 ###만 남길 것

# 03. GCP에서 데이터 분석 환경 구축하기

- 외부 프로그램 접속

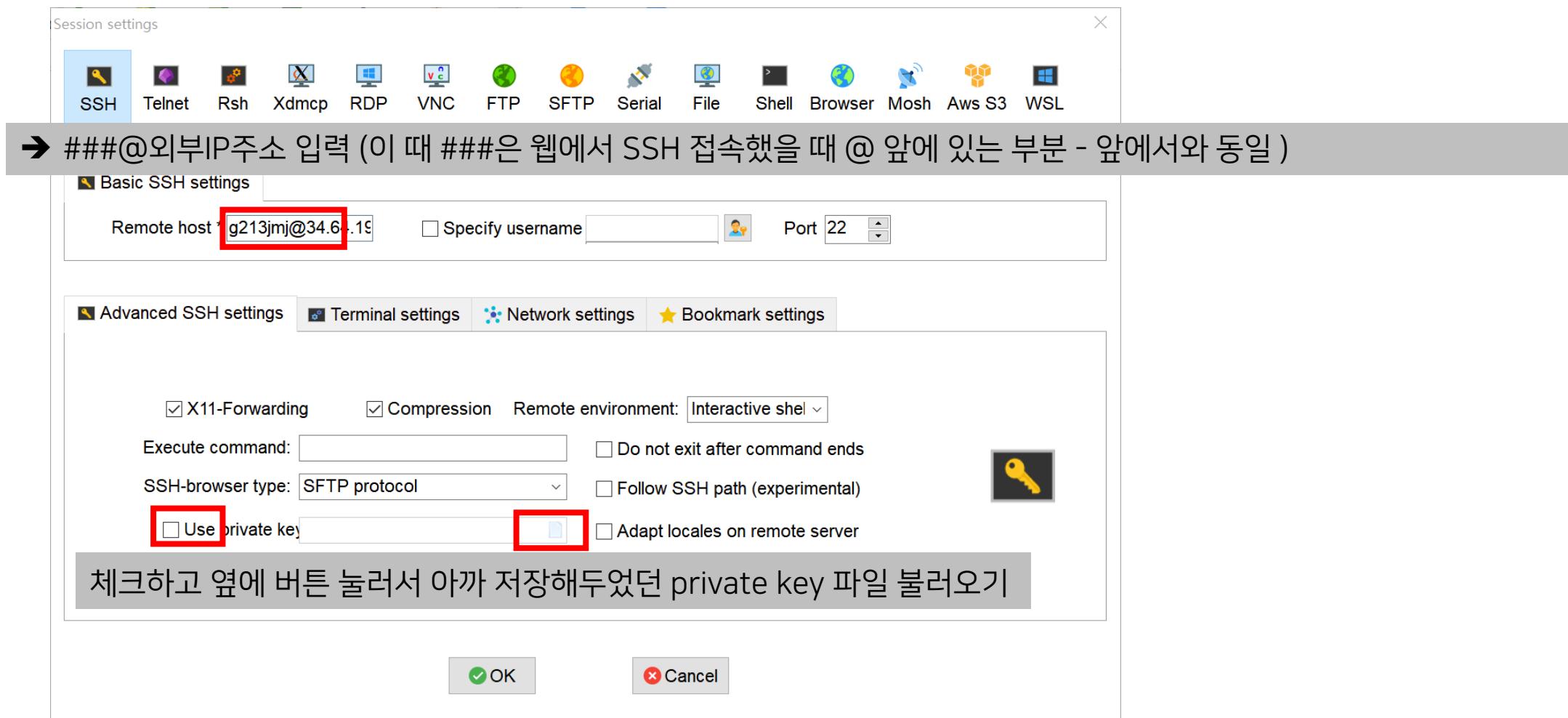
Step 3. 로컬에서 접속하기



# 03. GCP에서 데이터 분석 환경 구축하기

- 외부 프로그램 접속

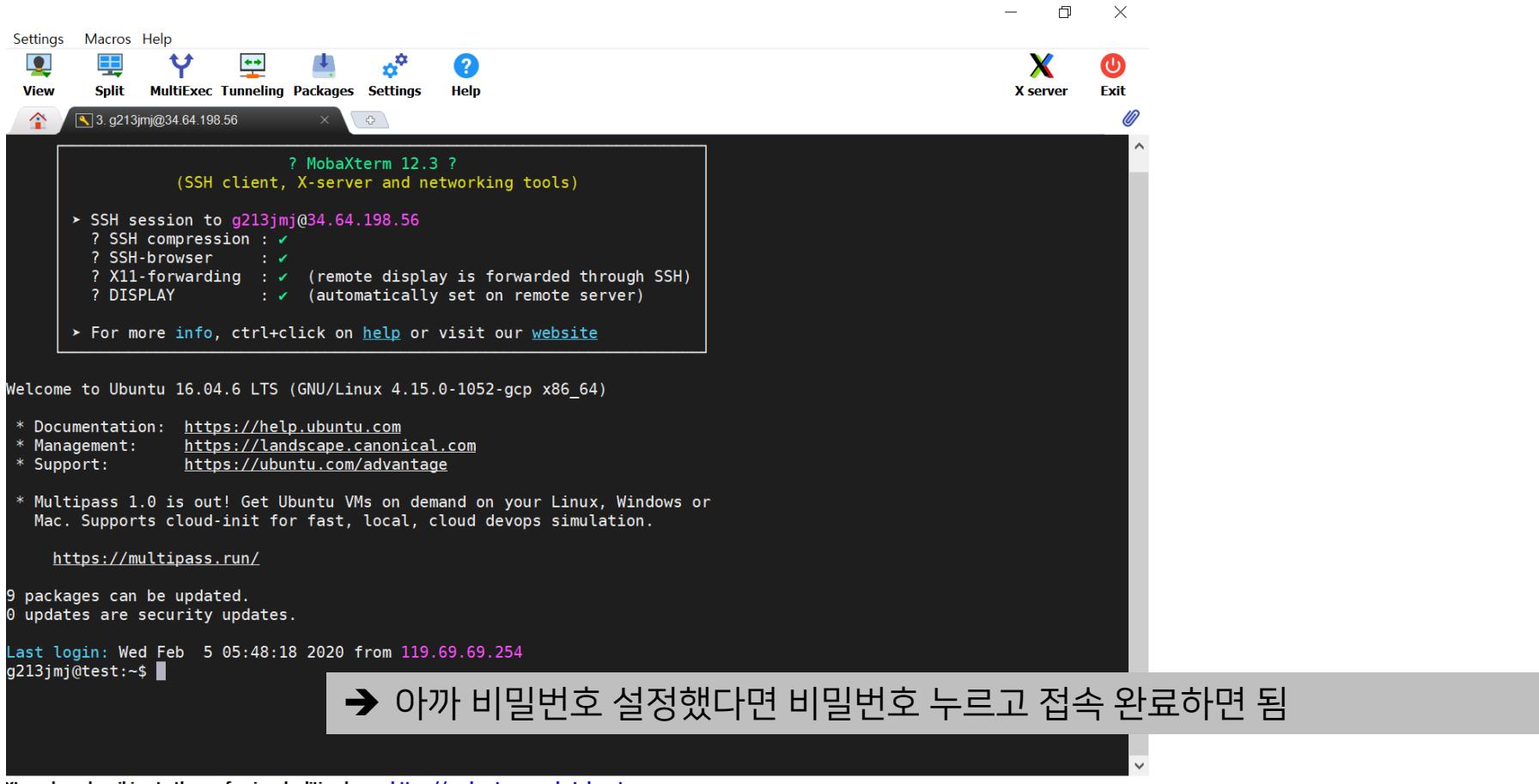
Step 3. 로컬에서 접속하기



# 03. GCP에서 데이터 분석 환경 구축하기

- 외부 프로그램 접속

Step 3. 로컬에서 접속하기



# 03. GCP에서 데이터 분석 환경 구축하기

- 백그라운드에서 계속 실행하기

★ 콘솔창을 닫더라도 접속이 계속 유지되게 하는 방법

```
$ nohup datalab connect 인스턴스이름 & datalab의 경우
```

```
$ nohup jupyter notebook & 직접 jupyter notebook 여는 경우
```

이걸 하지않고 콘솔창을 닫아버린다면(혹은 구글홈페이지에서 SSH접속했는데 인터넷 연결이 끊겨버린다면)  
계속 잘 돌아가고 있던 여러분의 소중한 모델 학습이 중단되니 꼭 참고

대신 나중에 쓰던 거 끝나면 꼭 인스턴스 중지해야 함  
안 그러면 순식간에 무료 크레딧을 다 쓰게 될 수도

# 03. GCP에서 데이터 분석 환경 구축하기

- 외부 프로그램 접속 & 백그라운드에서 계속 실행하기

## MobaXterm 팁

- 콘솔창에서 드래그만 해도 복사가 됨
- 마우스 오른쪽 클릭하면 붙여넣기 됨

## 로컬 접속

- 로컬에서 인스턴스 생성, 중지, 삭제 및 그 밖의 모든 클라우드 서비스를 로컬로 사용하고 싶다면 google cloud sdk 설치해서 사용할 것
- 참고 : <https://cloud.google.com/sdk>

# 04

BigQuery & SQL

# 04. BigQuery & SQL

- Cloud SQL vs BigQuery

## 관계형 데이터베이스란(Relational Data Base)?

- 현재 가장 많이 사용되고 있는 데이터베이스의 한 종류
- 여러 개의 테이블들이 다른 테이블들과 관계를 맺고 모여있는 집합체
- 각 테이블은 키(key)와 값(value)의 관계를 나타냄 (테이블은 우리가 흔히 접하는 데이터프레임이라고 생각하면 됨)
- 관계형 데이터베이스 관리 시스템(relational database management system, RDBMS)에는 Oracle, MySQL, PostgreSQL 등이 있음

## 데이터웨어하우스란(Datawarehouse)?

- 정보(data)+창고(warehouse)
- “**분석용 데이터 저장소**”
- 의사결정에 필요한 정보처리 기능을 효율적으로 지원하기 위한 통합된 데이터를 가진 양질의 데이터베이스
- 데이터를 축적하여 분석, 관리하는 시스템으로 이용됨
- 많은 양의 데이터를 효율적으로 저장하여 수십만 명의 사용자에게 빠른 속도로 쿼리 결과를 전달하는 것이 목적

# 04. BigQuery & SQL

- Cloud SQL vs BigQuery

관계형 데이터베이스란(Relational Data Base)?



Cloud SQL

- 현재 가장 많이 사용되고 있는 데이터베이스의 한 종류
- 여러 개의 테이블들이 다른 테이블들과 관계를 맺고 모여있는 집합체
- 각 테이블은 키(key)와 값(value)의 관계를 나타냄 (테이블은 우리가 흔히 접하는 데이터프레임이라고 생각하면 됨)
- 관계형 데이터베이스 관리 시스템(relational database management system, RDBMS)에는 Oracle, MySQL, PostgreSQL 등이 있음

데이터웨어하우스란(Datawarehouse)?



BigQuery

- 정보(data)+창고(warehouse)
- “분석용 데이터 저장소 ”
- 의사결정에 필요한 정보처리 기능을 효율적으로 지원하기 위한 통합된 데이터를 가진 양질의 데이터베이스
- 데이터를 축적하여 분석, 관리하는 시스템으로 이용됨
- 많은 양의 데이터를 효율적으로 저장하여 수십만 명의 사용자에게 빠른 속도로 쿼리 결과를 전달하는 것이 목적

# 04. BigQuery & SQL

- Cloud SQL vs BigQuery

Cloud SQL은

- 관계형 데이터베이스 관리시스템을 사용하여 **데이터를 보다 효율적으로 저장 및 관리하는 것을 목적으로 한다**

BigQuery는

- **대용량 데이터 분석을 빠르고 효율적으로 하는 것을 목적으로 한다**
- TB 단위의 데이터에 대해 초고속 SQL 쿼리를 실행하여 데이터 조회 및 분석(간단한 통계작업 등)이 가능하다
- 서버를 따로 구축하지 않기 때문에 관리가 필요 없고, 데이터만 업로드하여 사용하면 된다 (데이터 분석에만 집중 가능)
- 한 번 쓸 때마다 다루는 데이터 양에 대해서 과금 (월 1TB당 \$5)

데이터베이스 시스템에 대해 잘 알고 서버 구축에 익숙한 사람



**Cloud SQL**

빠르고 편리하게 데이터 분석하고 싶은 사람



**BigQuery**

# 04. BigQuery & SQL

데이터 분석할 때,

집계함수(sum, mean, std 등) or SQL로 가능한 전처리는 BigQuery에서  
원핫인코딩, 라벨인코딩, scaling 등 SQL로 하기 까다로운 전처리는 Python에서 하는 것 추천

# 04. BigQuery & SQL

- SQL이란?

## Query란?

- 질의어. 데이터베이스와 정보 시스템에 질의를 할 수 있게 하는 고급 컴퓨터 언어

## SQL이란?

- Structured Query Language
- RDBMS의 데이터를 관리하기 위해 설계된 특수 목적의 프로그래밍 언어
- 데이터베이스 스키마 생성 및 수정, 테이블 관리, 데이터 추가, 수정, 삭제, 조회 등 데이터베이스와 관련된 거의 모든 작업을 위해 사용되는 언어

# 04. BigQuery & SQL

- SQL 및 데이터베이스 용어

## SQL 용어

1. 열(column) = 필드(field), 속성(attribute)
  - 각각의 열은 유일한 이름을 가지고 있으며 자신만의 타입을 가짐
  
2. 행(row) = 튜플(tuple), 레코드(record)
  - 관계된 데이터의 묶음. 한 테이블의 모든 행은 같은 수의 열을 가지고 있음
  
3. 값(value)
  - 테이블은 각각의 행과 열에 대응하는 값을 가지고 있음. 이 값들은 각 열의 타입을 따라야함
  - 널값(Null) : 존재하지 않는 값을 의미하는 특별한 값
  
4. 테이블(table) = 관계(relation)
  - 같은 속성(attribute)을 지닌 튜플들의 집합



학번	이름	나이	주민번호
1	○○○값○○○	24	000000-0000000
2	△△△	22	XXXXXX-XXXXXXX
3	□□□	25	aaaaaa-aaaaaaaa

관계, 테이블

열, 필드, 속성

행, 튜플, 레코드

# 04. BigQuery & SQL

- SQL 및 데이터베이스 용어

## 5. 키(key)

- 관계(테이블)에서 조건에 만족하는 튜플(행)을 찾거나 순서대로 정렬할 때 다른 튜플들과 구별할 수 있는 유일한 기준이 되는 속성(열)
- 후보 키(candidate key) 중에서 데이터베이스 설계자가 지정

### (1) 후보키 (Candidate key)

- 관계를 구성하는 속성들 중에서 튜플을 유일하게 식별할 수 있는 속성들의 부분집합
- 모든 릴레이션은 반드시 하나 이상의 후보키를 가져야함
- 릴레이션에 있는 모든 튜플에 대해서 유일성과 최소성을 만족시켜야함

[ 학생 ] 테이블

학번	이름	나이	주민번호
1	○○○	24	0000000-0000000
2	△△△	22	XXXXXX-XXXXXXXX
3	□□□	25	aaaaaaaa-aaaaaaaa

[ 수강 ] 테이블

학번	과목명
1	Aaa
2	Bbb
2	Ccc

ex) [학생] 테이블의 후보키 : '학번', '주민번호'

# 04. BigQuery & SQL

- SQL 및 데이터베이스 용어

## (2) 기본키 (Primary key, PK)

- 후보키 중에서 데이터베이스 설계자가 선택한 메인 키
- 한 릴레이션에서 특정 튜플을 유일하게 구별할 수 있는 속성
- Null값을 가질 수 없음
- 기본키로 정의된 속성에는 동일한 값이 중복되어 저장될 수 없음

[ 학생 ] 테이블

학번	이름	나이	주민번호
1	○○○	24	0000000-00000000
2	△△△	22	xxxxxx-xxxxxxx
3	□□□	25	aaaaaa-aaaaaaaa

[ 수강 ] 테이블

학번	과목명
1	Aaa
2	Bbb
2	Aaa

ex) [학생] 테이블의 기본키 : '학번' 또는 '주민번호' 중 한 개 선택

ex) [수강] 테이블의 기본키 : '학번'+'과목명'

([수강] 테이블에서는 '학번'과 '과목명'은 다른 튜플들과 구별되지 않기 때문에 개별적으로 기본키로 사용할 수 없으나 묶어서 사용하면 구별이 가능해짐)

ex) [학생] 테이블에서 '학번'이 기본키로 정의되면 이미 입력된 '1'은 다른 튜플의 '학번' 속성 값으로 입력할 수 없음

# 04. BigQuery & SQL

- SQL 및 데이터베이스 용어

(3) 대체키 (Primary key) = 보조키

- 후보키가 둘 이상일 때 기본키를 제외한 나머지 후보키들

[ 학생 ] 테이블

학번	이름	나이	주민번호
1	○○○	24	0000000-00000000
2	△△△	22	XXXXXX-XXXXXXX
3	□□□	25	aaaaaaaa-aaaaaaaa

[ 수강 ] 테이블

학번	과목명
1	Aaa
2	Bbb
2	Ccc

ex) [학생] 테이블에서 '학번'을 기본 키로 선택했다면 '주민번호'가 대체키

# 04. BigQuery & SQL

- SQL 및 데이터베이스 용어

## (4) 외래키 (Foreign key)

- 다른 테이블의 기본키와 대응되는 키
- 참조되는 테이블의 기본키와 대응되어 테이블 간에 참조 관계를 표현할 수 있음
- 외래키로 지정되면 참조 테이블의 기본키에 없는 값은 입력할 수 없음 (참조 무결성 조건)

[ 학생 ] 테이블

학번	이름	나이	주민번호
1	○○○	24	0000000-0000000
2	△△△	22	XXXXXX-XXXXXXX
3	□□□	25	aaaaaaaa-aaaaaaaa

[ 수강 ] 테이블

학번	과목명
1	Aaa
2	Bbb
2	Ccc

ex) [수강] 테이블이 [학생] 테이블을 참조하고 있으므로 [학생] 테이블의 '학번'은 기본키이고, [수강] 테이블의 '학번'은 외래키

ex) [수강] 테이블의 '학번'에는 [학생] 테이블의 '학번'에 없는 값은 입력할 수 없다.

# 04. BigQuery & SQL

- SQL 및 데이터베이스 용어

## 6. 스키마

- 일종의 데이터명세서
- 테이블의 각 열에 대한 항목과 타입뿐만 아니라 기본키와 외래 키도 나타내야 함

# 04. BigQuery & SQL

- SQL 의 데이터 타입

가장 대중적인 RDBMS인 MySQL에서 지원하는 데이터타입

## MySQL DATA TYPES

DATE TYPE	SPEC	DATA TYPE	SPEC
CHAR	String (0 - 255)	INT	Integer (-2147483648 to 2147483647)
VARCHAR	String (0 - 255)	BIGINT	Integer (-9223372036854775808 to 9223372036854775807)
TINYTEXT	String (0 - 255)	FLOAT	Decimal (precise to 23 digits)
TEXT	String (0 - 65535)	DOUBLE	Decimal (24 to 53 digits)
BLOB	String (0 - 65535)	DECIMAL	"DOUBLE" stored as string
MEDIUMTEXT	String (0 - 16777215)	DATE	YYYY-MM-DD
MEDIUMBLOB	String (0 - 16777215)	DATETIME	YYYY-MM-DD HH:MM:SS
LONGTEXT	String (0 - 4294967295)	TIMESTAMP	YYYYMMDDHHMMSS
LONGBLOB	String (0 - 4294967295)	TIME	HH:MM:SS
TINYINT	Integer (-128 to 127)	ENUM	One of preset options
SMALLINT	Integer (-32768 to 32767)	SET	Selection of preset options
MEDIUMINT	Integer (-8388608 to 8388607)	BOOLEAN	TINYINT(1)

Copyright © mysqltutorial.org. All rights reserved.

테이블끼리 Join할 때 컬럼명이 같음에도 불구하고  
데이터 타입이 다른 경우 에러가 날 수 있으므로 주의

구글 BigQuery에서 지원하는 데이터 타입

<https://cloud.google.com/bigquery/docs/reference/standard-sql/data-types?hl=ko>

# 04. BigQuery & SQL

- SQL 문법의 종류

## SQL 문법의 종류

1. 데이터 정의 언어 (DDL : Data Definition Language)
2. 데이터 조작 언어 (DML : Data Manipulation Language)
3. 데이터 제어 언어 (DCL : Data Control Language)

# 04. BigQuery & SQL

- SQL 문법의 종류

## 1. 데이터 정의 언어 (DDL : Data Definition Language)

- 객체의 생성, 변경, 삭제 명령어 (**CREATE, ALTER, DROP** 등)
- 스키마(SCHEMA), 테이블(TABLE), 인덱스(INDEX) 등을 정의하거나 변경 혹은 삭제할 때 사용하는 언어
- 데이터베이스 관리자나 데이터베이스 설계자가 사용

## 2. 데이터 조작 언어 (DML : Data Manipulation Language)

- **SELECT, INSERT, UPDATE, DELETE**
- 데이터 CRUD [Create(생성), Read(읽기), Update(갱신), Delete(삭제)]
- 데이터베이스 사용자가 응용프로그램이나 질의어를 통하여 저장된 데이터를 실질적으로 처리하는 데 사용하는 언어
- 데이터베이스 사용자와 데이터베이스 관리시스템 간의 인터페이스 제공

# 04. BigQuery & SQL

- SQL 문법의 종류

## 3. 데이터 제어 언어 (DCL : Data Control Language)

- **GRANT** : 객체 사용 권한 설정
- 데이터의 보안, 무결성, 데이터 회복 등을 정의하는 데 사용하는 언어
- 데이터베이스 관리자가 데이터 관리를 목적으로 사용

BigQuery에서는,

데이터 제어 언어(DCL)는 쓸 수 없음

& Standard SQL 사용

# 04. BigQuery

- SELECT

- SELECT는 테이블 내용을 보기 위한 쿼리

SELECT 컬럼명1, … (if 모든 컬럼 선택 시 \*)

FROM 테이블명

WHERE 조건

GROUP BY 그룹화할 컬럼

HAVING 그룹화한 뒤 조건

ORDER BY 정렬기준컬럼 [DESC|ASC]

LIMIT 제한할 개수

[테이블명]에서

[조건]을 만족하는 row들 중

[그룹화할 컬럼]으로 그룹화해서

→ [그룹화한 뒤 조건] 을 가지는

[컬럼명] 을

[정렬기준컬럼]을 기준으로 정렬해서

[제한할 개수]만 선택한다

※ BigQuery에서 주의할 점

테이블명 → `프로젝트이름.데이터세트이름.테이블이름` 으로 써야함 (프로젝트 이름은 생략 가능)

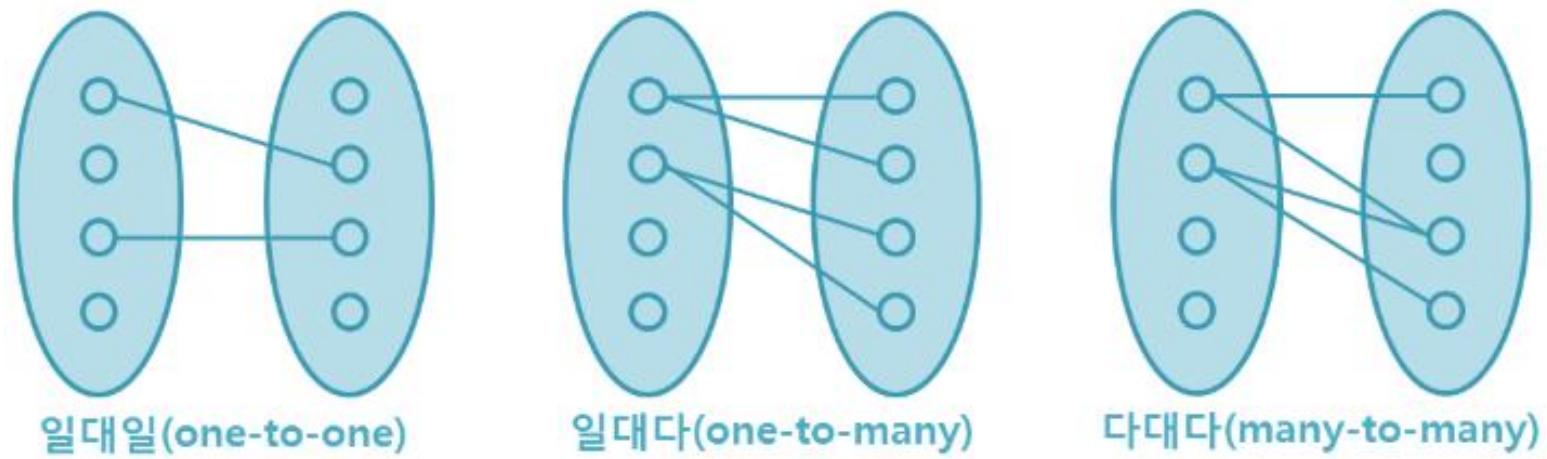
# 04. BigQuery & SQL

## - JOIN

테이블 간의 관계는 다음과 같이 표현된다

- 1) 일대일(one-to-one) 관계
- 2) 일대다(one-to-many) 관계
- 3) 다대다(many-to-many) 관계

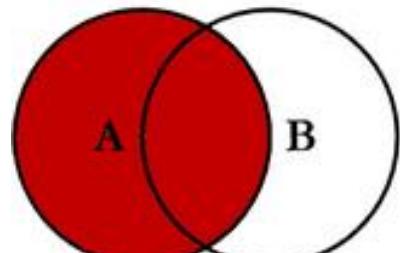
앞서 살펴본 [학생]테이블과 [수강]테이블은 일대다 관계



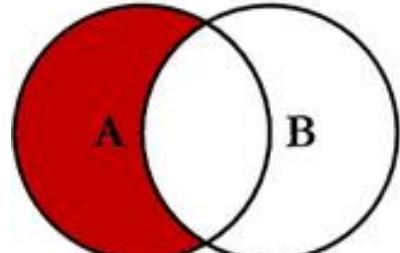
# 04. BigQuery

- JOIN

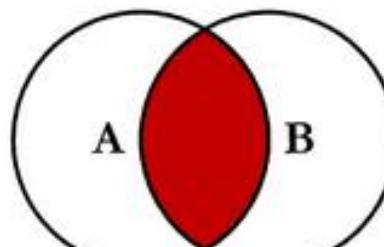
## SQL JOINS



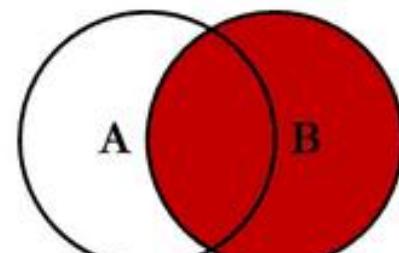
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



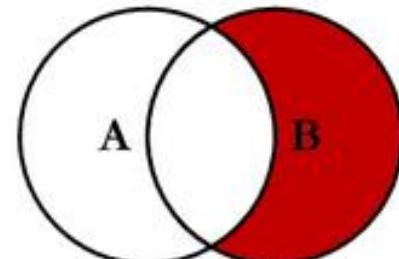
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL.
```



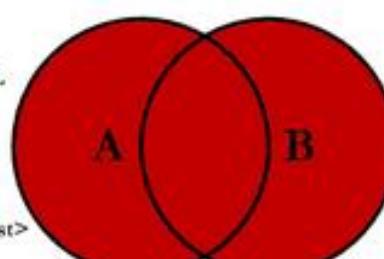
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



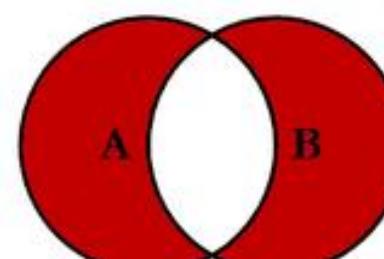
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL.
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL.
OR B.Key IS NULL
```

# 04. BigQuery

- 그 밖의 명령어들

첨부파일 [SQL-cheat-sheet.pdf](#), [SQL\\_cheatsheet.png](#) 및 [BigQuery\\_in\\_jupyter.ipynb](#) 참고

# 04. BigQuery & SQL

- BigQuery 실습 (테이블 생성하기)

[순서] BigQuery에 테이블(데이터)을 저장하고 SQL 쿼리를 통해 데이터를 조회 및 분석한다

※ 테이블 생성하는 방법

- (1) 빈 테이블
- (2) Google Cloud Storage (추천) 실습
- (3) 직접 업로드 (로컬 업로드는 10MB이하만 가능하기 때문에 추천하지 않음)
- (4) 구글 드라이브
- (5) Google Cloud Bigtable

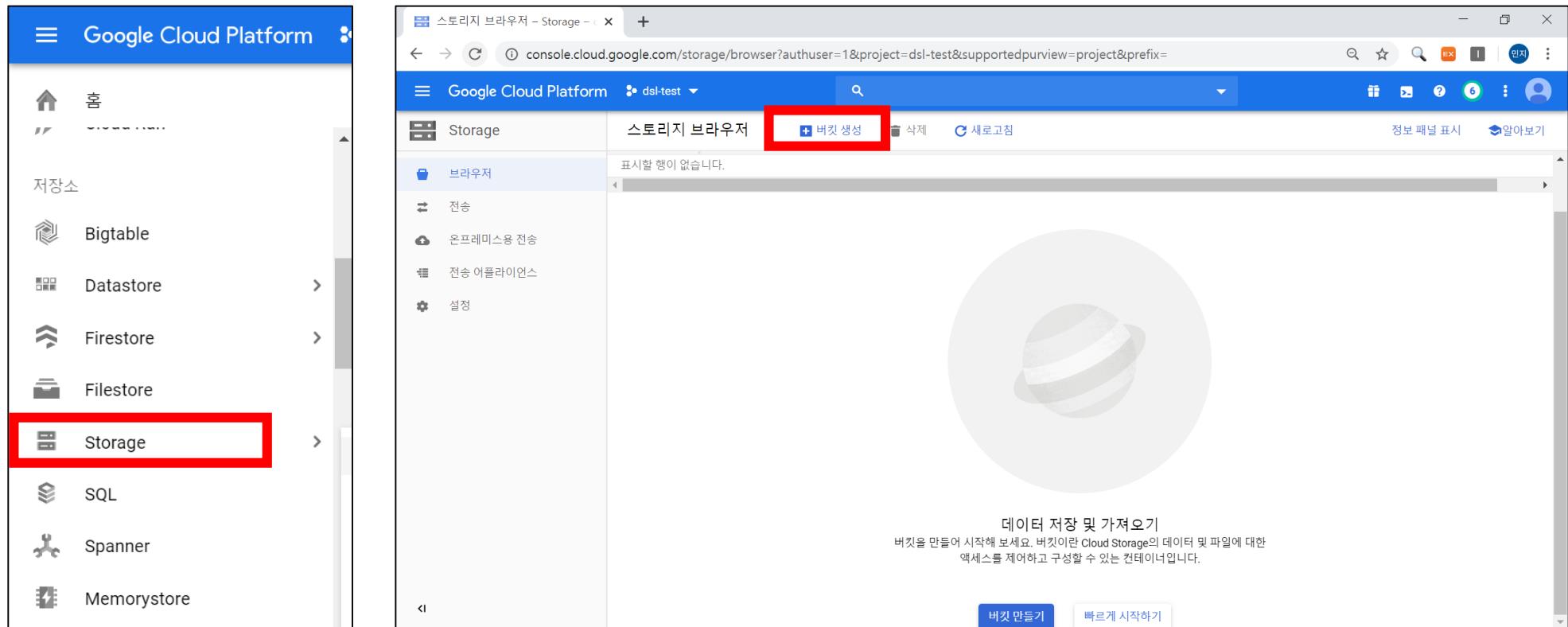
# 04. BigQuery

- Google Cloud Storage 사용하여 BigQuery에 테이블 만들기

Step 1. Google Cloud Storage 에 버킷 생성 및 파일 업로드

(1) 콘솔에서 좌측탭 클릭 → 저장소 → Storage 클릭 → + 버킷 생성 클릭

※ 버킷이란  
데이터를 담는 기본 컨테이너.  
Cloud Storage에 저장되는 모든 데이터는 버킷  
안에 포함된다는 의미



# 04. BigQuery

- Google Cloud Storage 사용하여 BigQuery에 테이블 만들기

Step 1. Google Cloud Storage 에 버킷 생성 및 파일 업로드

(2) 버킷 만들기

The screenshot shows the 'Bucket Name' step of the Google Cloud Platform Storage bucket creation process. The left sidebar shows 'Storage' selected. The main area has a title 'Bucket Name' with a back arrow. It contains a checked checkbox for 'Bucket name' and a text input field containing 'bigquery-test-upload'. Below it is a note: 'Tip: Minimize the number of bytes in the name to reduce costs.' A 'Next Step' button is at the bottom.

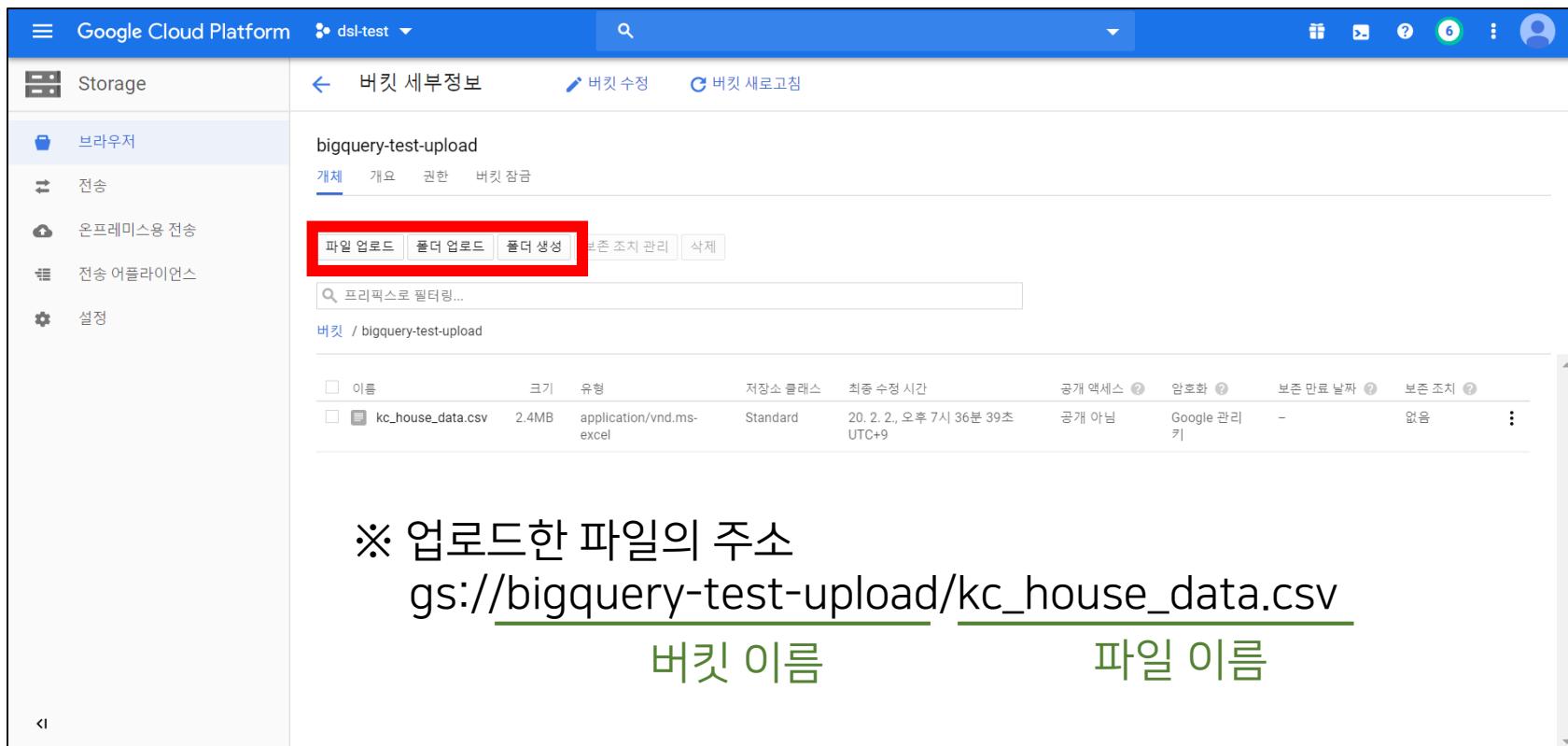
- 이름 : bigquery-upload-test
  - 위치유형 : Region
  - 위치 : asia-northeast3  
**(VM 인스턴스의 리전과 동일해야함)**
- 나머지는 default값으로  
→ 만들기

# 04. BigQuery

- Google Cloud Storage 사용하여 BigQuery에 테이블 만들기

## Step 1. Google Cloud Storage 에 버킷 생성 및 파일 업로드

### (3) 파일 업로드

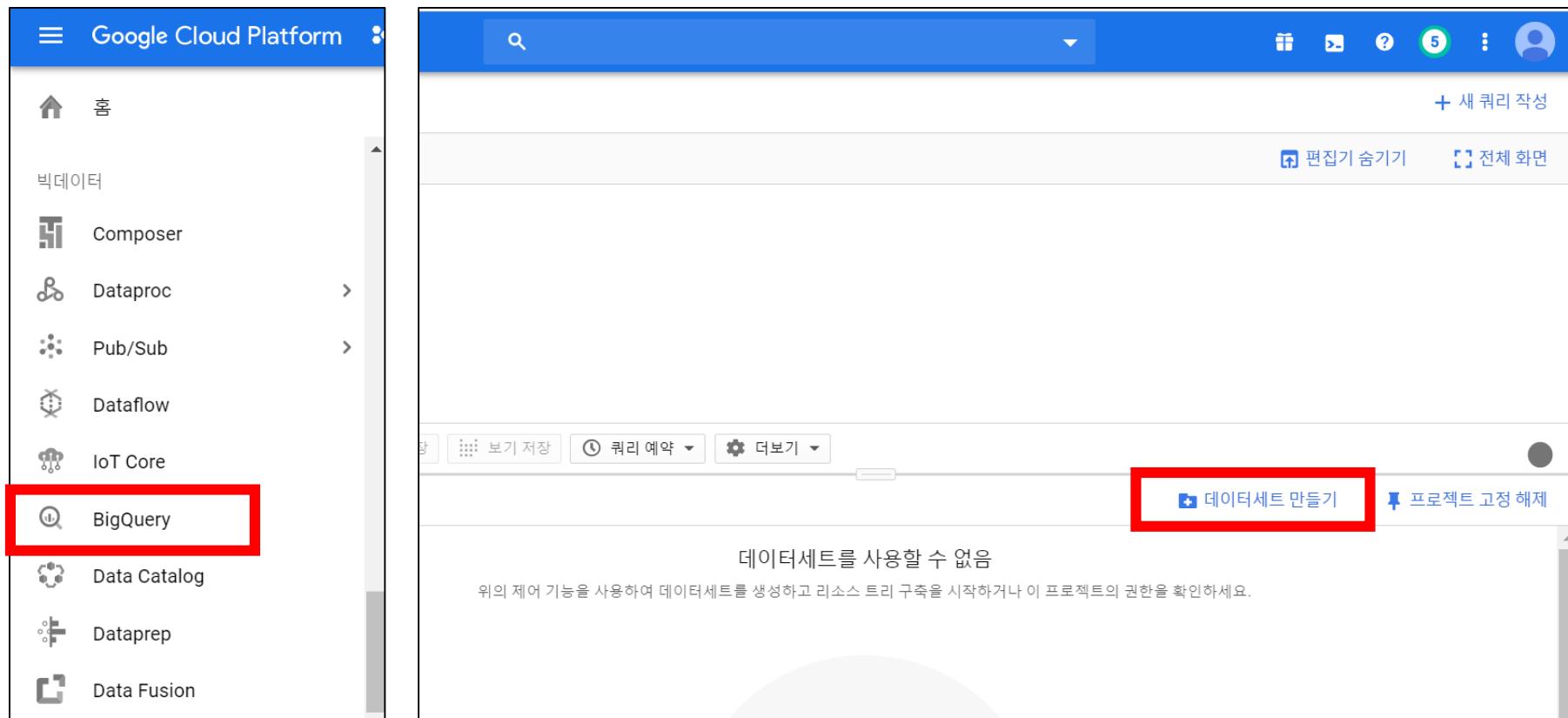


# 04. BigQuery

- Google Cloud Storage 사용하여 BigQuery에 테이블 만들기

Step 2. BigQuery에 테이블 만들기

(1) 콘솔에서 좌측탭 클릭 → 빅데이터 → BigQuery 클릭 → 데이터세트 만들기 클릭

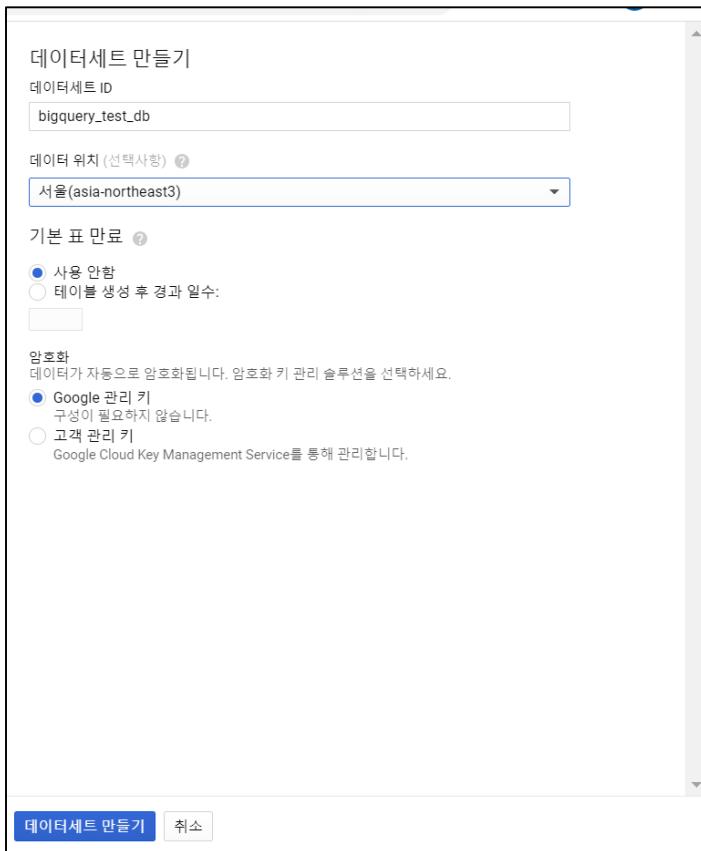


# 04. BigQuery

- Google Cloud Storage 사용하여 BigQuery에 테이블 만들기

Step 2. BigQuery에 테이블 만들기

(2) 데이터세트 만들기



- 이름 : bigquery\_test\_db
- 데이터위치 : 서울(asia-northeast3 )  
**(VM 인스턴스의 리전과 동일해야함)**

나머지는 default값으로

→ 데이터세트 만들기

※ 데이터세트 안에 여러 개의 테이블 만들 수 있음  
데이터세트는 폴더, 테이블은 데이터 파일이라고 생각하면 쉬움  
따라서 이름 지을 때 데이터 세트의 용도를 고려해서 지으면 편함

# 04. BigQuery

- Google Cloud Storage 사용하여 BigQuery에 테이블 만들기

Step 2. BigQuery에 테이블 만들기

(3) 테이블 만들기

The screenshot shows the Google Cloud Platform BigQuery interface. On the left, there is a sidebar with navigation links: 쿼리 기록, 저장된 쿼리, 작업 기록, 전송, 예약된 쿼리, 예약, BI Engine, 리소스 (+ 데이터 추가), and a dropdown for the project 'dsl-test'. Below the dropdown, the dataset 'bigquery\_test\_db' is selected. In the main area, there is a search bar and a toolbar with buttons for 실행 (Run), 쿼리 저장 (Save Query), 보기 저장 (Save View), 쿼리 예약 (Schedule Query), and 더보기 (More). A red box highlights the '+ 테이블 만들기' (Create Table) button. To the right of the toolbar, there are buttons for 데이터세트 공유 (Share Dataset), 데이터세트 복사 (Copy Dataset), and 데이터세트 삭제 (Delete Dataset). Below the toolbar, there are sections for 설명 (Description) and 라벨 (Label), both of which have an '없음' (None) status. At the bottom, there is a '데이터세트 정보' (Dataset Info) section with the following details:

데이터세트 ID	dsl-test:bigquery_test_db
생성 시간	2020. 2. 2. 오후 7:30:55
기본 표 만료	사용 안함
최종 수정 시간	2020. 2. 2. 오후 7:30:55
데이터 위치	asia-northeast3

# 04. BigQuery

- Google Cloud Storage 사용하여 BigQuery에 테이블 만들기

Step 2. BigQuery에 테이블 만들기

(3) 테이블 만들기

The image consists of three side-by-side screenshots illustrating the steps to create a BigQuery table from a CSV file stored in Google Cloud Storage.

**Screenshot 1: BigQuery Table Creation Step 1**

This screenshot shows the "Create Table" dialog. In the "Source" section, "Google Cloud Storage" is selected (highlighted with a red box). A large gray callout box with the text "클릭하고 그 다음 파일 찾아보기 클릭" (Click and then click 'Select File') points to the "Select File" button. Other fields include "Project ID" (dsl-test), "Table ID" (bigquery\_test\_db), and "Table Name" (left empty).

**Screenshot 2: BigQuery Table Creation Step 2**

This screenshot shows the "Select File" dialog. It displays a list of buckets under "Location": "bigquery-test-upload". The file "kc\_house\_data.csv" is selected (highlighted with a red box).

**Screenshot 3: BigQuery Table Creation Step 3**

This screenshot shows the "Select File" dialog again, confirming the selection of "kc\_house\_data.csv" from the "bigquery-test-upload" bucket (highlighted with a red box).

**Bottom Buttons**

At the bottom of each dialog are two buttons: "선택" (Select) and "취소" (Cancel).

# 04. BigQuery

- Google Cloud Storage 사용하여 BigQuery에 테이블 만들기

Step 2. BigQuery에 테이블 만들기

(3) 테이블 만들기

테이블 만들기

소스  
다음 항목으로 테이블 만들기: GCS 버킷에서 파일 선택:  bigquery-test-upload/kc\_house\_data.csv

파일 형식: CSV

대상  
프로젝트 이름: dsl-test  
데이터셋 이름: bigquery\_test\_db  
테이블 이름: kc\_house\_data

**스키마**  
자동 감지  스키마 및 입력 매개변수

스키마를 직접 정의하고 싶다면  
체크하지 말고 직접 작성하기

파티션 및 클러스터 설정  
분할:  파티션 없음

클러스터링 순서(선택사항): Clustering can only be used on a partitioned table, and works with tables partitioned either by column or ingestion time.

테이블 만들기 취소

Google Cloud Platform :: dsl-test

BigQuery

쿼리 기록

작업 기록

전송

예약된 쿼리

예약

BI Engine

리소스 + 데이터 추가

모드 데이터세트 검색

dc-test

bigquery\_test\_db

kc\_house\_data

스키마 세부정보 미리보기

필드 이름	유형	모드	설명
id	INTEGER	NULLABLE	
date	STRING	NULLABLE	
price	FLOAT	NULLABLE	
bedrooms	INTEGER	NULLABLE	
sqft_living	INTEGER	NULLABLE	
sqft_lot	INTEGER	NULLABLE	
floors	FLOAT	NULLABLE	
waterfront	INTEGER	NULLABLE	

테이블 생성된 모습

# 04. BigQuery

- Google Cloud Storage 사용하여 BigQuery에 테이블 만들기

Step 3. BigQuery에서 쿼리 실행해보기

The screenshot shows the Google Cloud Platform BigQuery interface. On the left, there's a sidebar with navigation links like 'BigQuery', '기능 및 정보', '단축키', '+ 새 쿼리 작성', '편집기 승기기', and '전체 화면'. Below that is a tree view of datasets and tables under 'dls-test': 'bigquery\_test\_db' and 'kc\_house\_data'. The main area is titled '쿼리 편집기' (Query Editor) and contains the following code:

```
1 SELECT *
2 FROM bigquery_test_db.kc_house_data;
```

Below the code are buttons for '실행' (Run), '쿼리 저장' (Save Query), '보기 저장' (Save View), '쿼리 예약' (Schedule Query), and '더보기' (More). A note says '실행 시 이 쿼리가 3.6MB를 처리합니다.' (This query processes 3.6MB when run). The results section is titled '쿼리 결과' (Query Results) and shows a table with 14 columns: id, date, price, bedrooms, bathrooms, sqft\_living, sqft\_lot, floors, waterfront, view, condition, grade, sqft\_above, sqft\_basement, yr\_built. The table has 5 rows of data. At the bottom, a message says "'anon85de46e948eba00bbeb2f4f3a6465099fe47f733'을 (를) 만들었습니다. 테이블로 이동" (Table created successfully. Go to table). The footer shows pagination information: '페이지당 행 수: 100', '1 - 100 / 21613', '첫 페이지 <', '>', and '>> 마지막 페이지'.

# 05

Jupyter Notebook에서 BigQuery 사용하기

# 05. Jupyter Notebook에서 BigQuery 사용하기

방법 1) GCP 가상머신에서

방법 2) 로컬에서

# 05. Jupyter Notebook에서 BigQuery 사용하기

- 방법 1) GCP 가상머신에서

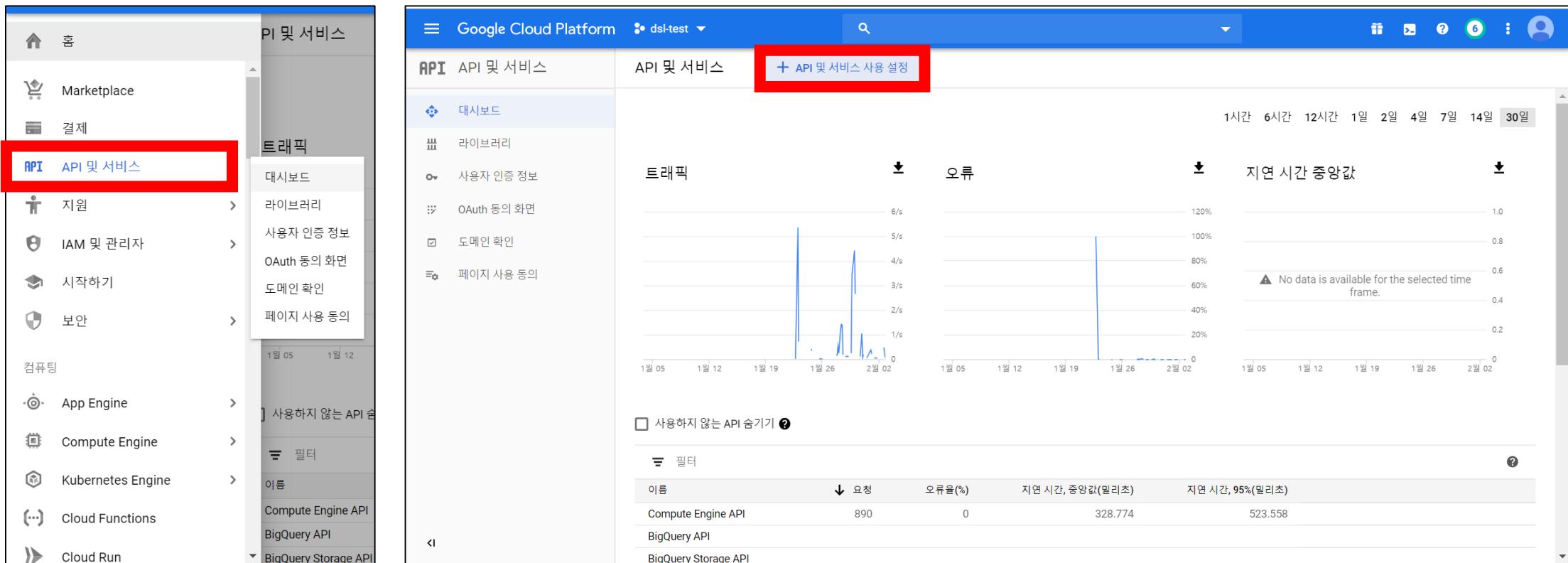
방법 1) GCP 가상머신에서

# 05. Jupyter Notebook에서 BigQuery 사용하기

- 방법 1) GCP 가상머신에서

## Step 1. 필요한 API 사용 설정

(1) 콘솔에서 좌측탭 클릭 → API 및 서비스로 이동 → + API 및 서비스 사용 설정 클릭



The screenshot shows the Google Cloud Platform API & Services page. On the left, there is a sidebar with various services like Marketplace, API & Services, IAM, etc. The 'API & Services' section is highlighted with a red box. In the main content area, there are two tabs: 'API 대시보드' and 'API 및 서비스'. The 'API 및 서비스' tab is selected and has a red box around its header. Below the tabs, there are sections for '트래픽' (Traffic) and '오류' (Errors). A prominent red box highlights the '+ API 및 서비스 사용 설정' (Add API and service usage configuration) button. At the bottom, there is a table showing API usage statistics:

이름	요청	오류율(%)	지연 시간, 중앙값(밀리초)	지연 시간, 95%(밀리초)
Compute Engine API	890	0	328.774	523.558
BigQuery API				
BigQuery Storage API				

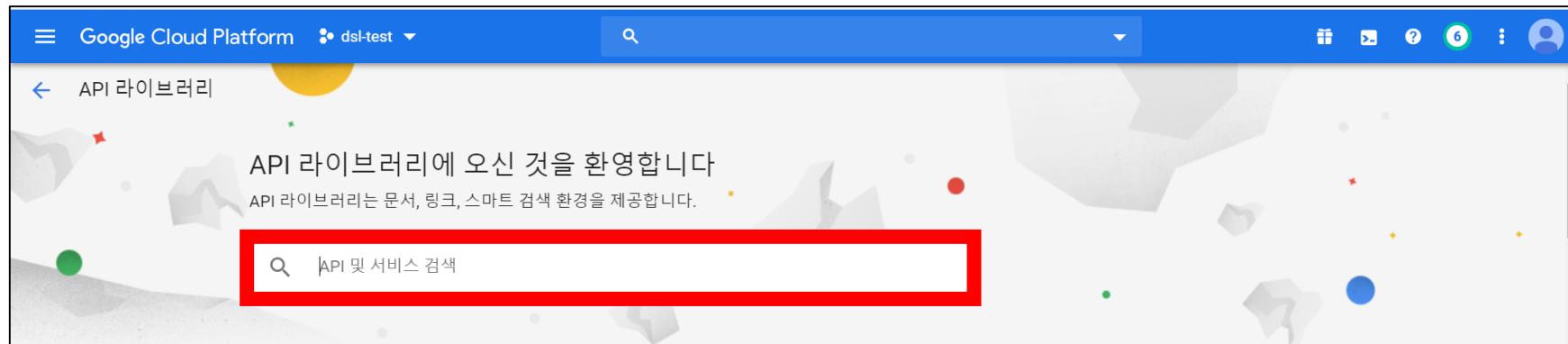
# 05. Jupyter Notebook에서 BigQuery 사용하기

- 방법 1) GCP 가상머신에서

Step 1. 필요한 API 사용 설정

(2) 검색창에 BigQuery 입력 → BigQuery API 클릭 → 사용설정

(3) 검색창에 BigQuery Storage 입력 → BigQuery Storage API 클릭 → 사용설정



# 05. Jupyter Notebook에서 BigQuery 사용하기

- 방법 1) GCP 가상머신에서

Step 2. 개발환경 인증

(1) 좌측탭에서 API 및 서비스 클릭 → 사용자 인증정보 클릭 → + 사용자 인증 정보 만들기 클릭

The screenshot shows the Google Cloud Platform (GCP) interface. On the left, there is a sidebar with various service links. The 'API & Services' link is highlighted with a red box. The main content area is titled 'API & Services' and shows a sub-section 'User Authentication'. A red box highlights the '+ Create User Authentication Information' button. Below this, there are sections for 'OAuth 2.0 Client ID', 'API Keys', and 'Service Accounts'. The 'OAuth 2.0 Client ID' section is currently empty, showing a message 'No OAuth 2.0 Client ID is present.' The 'Service Accounts' section lists one account: '1020435037078-compute@developer.gserviceaccount.com'.

# 05. Jupyter Notebook에서 BigQuery 사용하기

- 방법 1) GCP 가상머신에서

Step 2. 개발환경 인증

(2) 서비스 계정 클릭

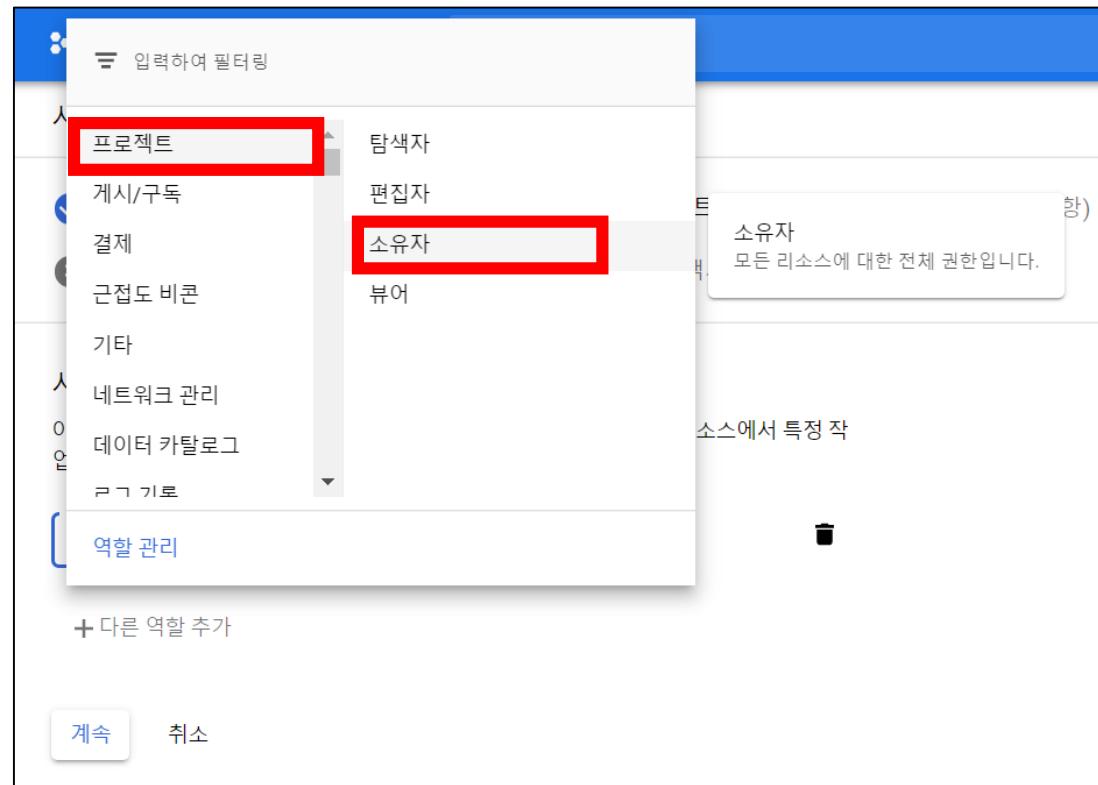
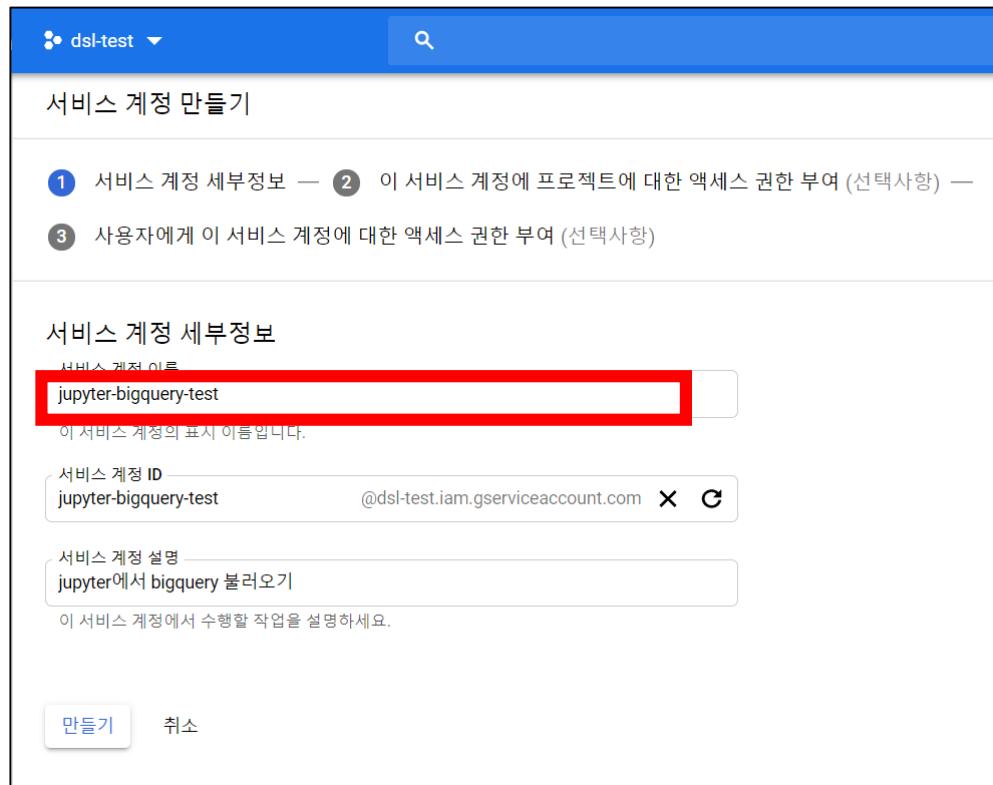
The screenshot shows the Google Cloud Platform interface for managing APIs and services. On the left, there's a sidebar with options like 대시보드, 라이브러리, 사용자 인증 정보 (which is selected), OAuth 동의 화면, 도메인 확인, and 페이지 사용 동의. The main area is titled 'API 및 서비스' and shows '사용자 인증 정보'. A red box highlights the '+ 사용자 인증 정보 만들기' button. Below it, there are sections for 'API 키' and 'OAuth 클라이언트 ID', both of which have red boxes around them. Under 'API 키', there's a note about using service accounts for server-to-server authentication. Under 'OAuth 2.0 클라이언트 ID', it says '표시할 OAuth 클라이언트가 없습니다.' At the bottom right of the main area, there's a button for '동의 화면 구성'.

# 05. Jupyter Notebook에서 BigQuery 사용하기

- 방법 1) GCP 가상머신에서

Step 2. 개발환경 인증

(3) 서비스 계정 만들기



# 05. Jupyter Notebook에서 BigQuery 사용하기

- 방법 1) GCP 가상머신에서

Step 2. 개발환경 인증

(3) 서비스 계정 만들기 – 키 만들기

The screenshot shows the 'Service Account' creation interface. At the top, there are two checked checkboxes: 'Service account information' and 'This service account has access to your project'. Below these, there is another checkbox for 'Users or groups that need to perform actions as this service account'. A red box highlights the 'Create key' button at the bottom left.

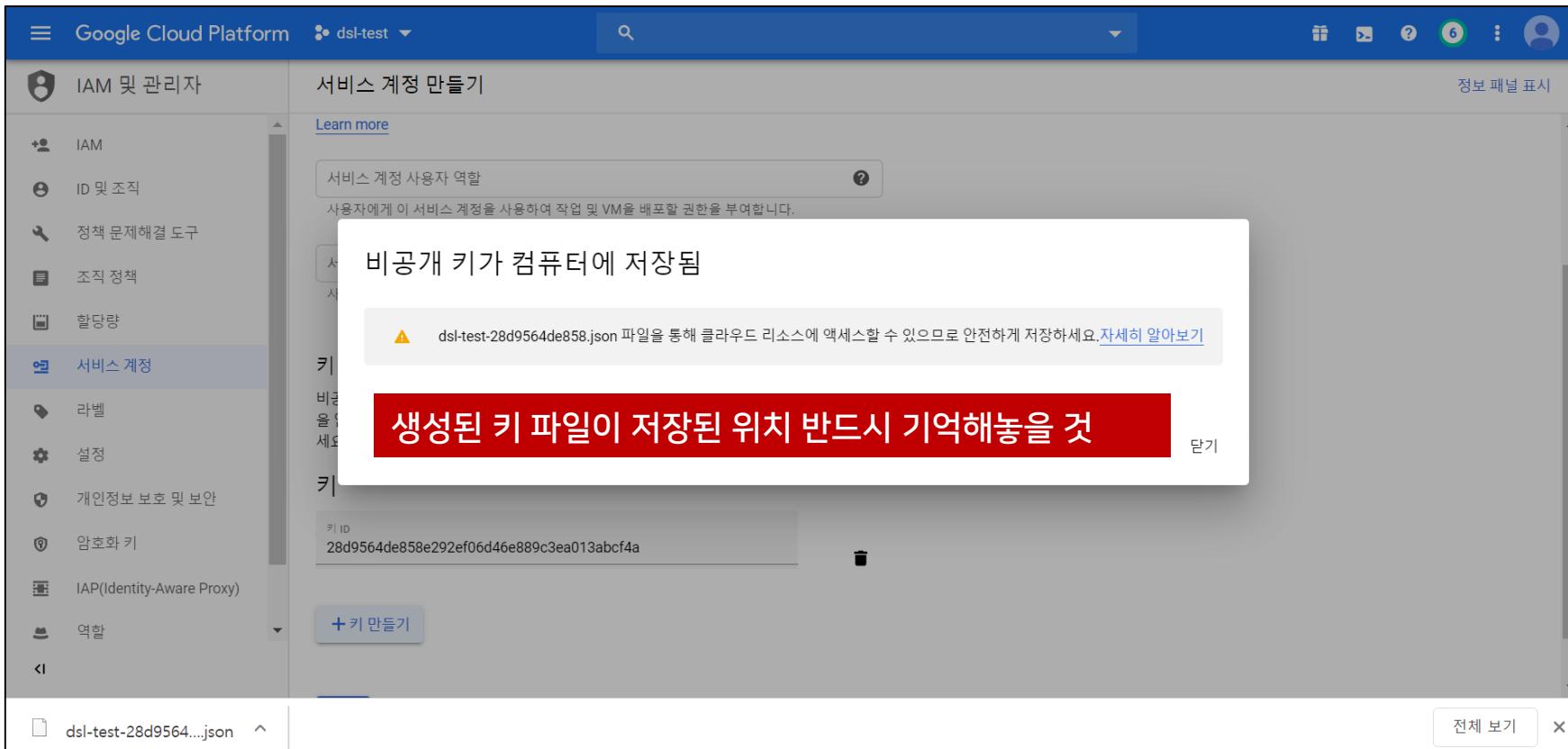
The screenshot shows the 'Create Key' configuration page. It includes a note about downloading a JSON key file for security. It offers two options: 'JSON' (selected) and 'P12'. Below the options are two buttons: a large red-bordered 'Create' button and a smaller 'Cancel' button.

# 05. Jupyter Notebook에서 BigQuery 사용하기

- 방법 1) GCP 가상머신에서

Step 2. 개발환경 인증

(3) 서비스 계정 만들기 – 키 만들기



# 05. Jupyter Notebook에서 BigQuery 사용하기

- 방법 1) GCP 가상머신에서

Step 2. 개발환경 인증

(4) 서비스 계정 생성 완료

The screenshot shows the Google Cloud Platform API & Services page under the 'API 및 서비스' tab. The left sidebar lists several options: 대시보드, 라이브러리, 사용자 인증 정보 (which is selected), OAuth 동의 화면, 도메인 확인, and 페이지 사용 동의. The main content area has three sections: '사용자 인증 정보' (User Authentication Information), 'API 키' (API Key), and 'OAuth 2.0 클라이언트 ID' (OAuth 2.0 Client ID). In the 'API 키' section, it says '표시할 API 키가 없습니다.' (No API keys displayed). In the 'OAuth 2.0 클라이언트 ID' section, it says '표시할 OAuth 클라이언트가 없습니다.' (No OAuth clients displayed). In the '서비스 계정' (Service Account) section, there is a table with three rows:

이메일	이름 ↑	모든 서비스의 사용률(지난 30일)
1020435037078-compute@developer.gserviceaccount.com	Compute Engine default service account	0
jupyter-bigquery-test@dsl-test.iam.gserviceaccount.com	jupyter-bigquery-test	0

A large gray box at the bottom right of the table contains the text '서비스 계정 생성된 모습' (Service account created successfully).

# 05. Jupyter Notebook에서 BigQuery 사용하기

- 방법 1) GCP 가상머신에서

Step 2. 개발환경 인증

(5) 키 파일을 가상머신 안에 저장 – jupyter notebook에서 upload 버튼을 눌러서 키 파일(3에서 로컬에 저장해놓은 키 파일) 업로드하기

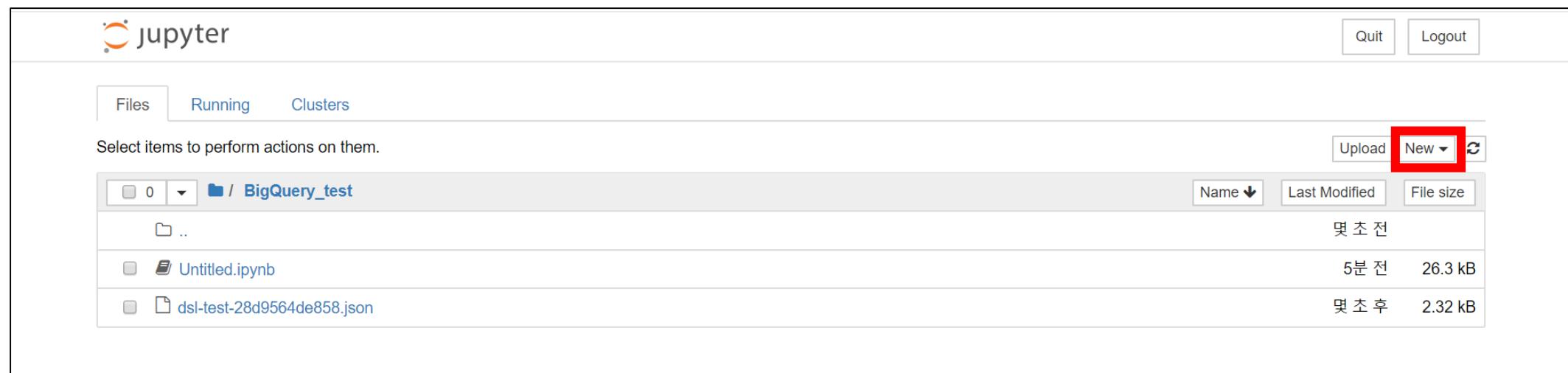


# 05. Jupyter Notebook에서 BigQuery 사용하기

- 방법 1) GCP 가상머신에서

Step 2. 개발환경 인증

(6) 키 파일 저장된 위치에서 New 버튼 - Python3 눌러서 Notebook 파일 한 개 생성



# 05. Jupyter Notebook에서 BigQuery 사용하기

- 방법 1) GCP 가상머신에서

Step 2. 개발환경 인증

(7) 환경 변수 설정 - 환경 변수 *GOOGLE\_APPLICATION\_CREDENTIALS*를 설정하여 애플리케이션 코드에 사용자 인증 정보를 제공

```
jupyter bigquery_in_jupyter_notebook Last Checkpoint: 몇 초 전 (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
In [1]: # 현재 경로에 있는 파일들 보여줌
!ls
ds1-test-28d9564de858.json Untitled.ipynb
In [2]: # 현재 파일이 있는 경로를 보여줌
!pwd
/home/g213jmj/BigQuery_test
In [4]: # 환경변수 설정
!export GOOGLE_APPLICATION_CREDENTIALS="/home/g213jmj/BigQuery_test/ds1-test-28d9564de858.json"
```

※ 주의 : = 에서 띄어쓰기 하면 안됨

※ jupyter notebook에서 코드 앞에 쓰는 ! 는 ! 이후에 나오는 내용들을 터미널에서 입력하는 것과 동일하게 처리 한다는 의미  
따라서 위의 과정을 터미널창에서 해도 상관은 없음

# 05. Jupyter Notebook에서 BigQuery 사용하기

- 방법 1) GCP 가상머신에서

Step 2. 개발환경 인증

(8) 개발환경 인증이 제대로 되었는지 확인

```
In [5]: def implicit():
    from google.cloud import storage

    # If you don't specify credentials when constructing the client, the
    # client library will look for credentials in the environment.
    storage_client = storage.Client()

    # Make an authenticated API request
    buckets = list(storage_client.list_buckets())
    print(buckets)

In [6]: implicit()
```

[<Bucket: bigquery-test-upload>]

```
def implicit():
    from google.cloud import storage

    # If you don't specify credentials when constructing the client, the
    # client library will look for credentials in the environment.
    storage_client = storage.Client()

    # Make an authenticated API request
    buckets = list(storage_client.list_buckets())
    print(buckets)
```

붙여넣기

이렇게 아까 생성했던 Bucket 이름이 나오면 성공적으로 개발환경 인증이 된 것

# 05. Jupyter Notebook에서 BigQuery 사용하기

- 방법 1) GCP 가상머신에서

Step 3. Jupyter Notebook에서 BigQuery 사용

- BigQuery\_in\_jupyter.ipynb 실습파일 확인

# 05. Jupyter Notebook에서 BigQuery 사용하기

- 방법 2) 로컬에서

방법 2) 로컬에서

# 05. Jupyter Notebook에서 BigQuery 사용하기

- 방법 2) 로컬에서

Step 1. 필요한 API 사용 설정

방법 1) - Step 1 참고

Step 2. 개발환경 인증

방법 1) - Step 2 - (5)~(8)의 과정을 로컬 터미널창에서 하면 됨

<https://cloud.google.com/docs/authentication/getting-started?authuser=1> 의 [환경변수 설정]을 참고하여  
본인의 OS(Linux, mac / window)에 따라 로컬 환경에서 개발환경 인증할 것

Step 3. Jupyter에서 BigQuery 사용해보기

# 06. 참고자료

- <https://bcho.tistory.com/1107>
- <https://su2cstar.github.io/gcp/2019/03/28/Jupyter-Notebook-and-RStudio-on-GCP.html>
- <https://zzsza.github.io/gcp/2018/08/02/bigquery-and-datalab/#datalab%EC%97%90%EC%84%9C-bigquery-%EC%97%B0%EB%8F%99>
- <https://cloud.google.com/bigquery-ml/docs/bigqueryml-intro?hl=ko>
- <https://cloud.google.com/bigquery/docs/bigquery-storage-python-pandas?hl=ko>
- <https://bcho.tistory.com/1116>
- <https://cloud.google.com/docs/authentication/getting-started?authuser=1>
- [http://tcpschool.com/mysql/mysql\\_intro\\_relationalDB](http://tcpschool.com/mysql/mysql_intro_relationalDB)
- [https://www.fun-coding.org/mysql\\_basic1.html](https://www.fun-coding.org/mysql_basic1.html)
- [https://ko.wikipedia.org/wiki/%EB%8D%B0%EC%9D%B4%ED%84%B0\\_%EC%9B%A8%EC%96%B4%ED%95%98%EC%9A%B0%EC%8A%A4](https://ko.wikipedia.org/wiki/%EB%8D%B0%EC%9D%B4%ED%84%B0_%EC%9B%A8%EC%96%B4%ED%95%98%EC%9A%B0%EC%8A%A4)
- <https://limkydev.tistory.com/108>
- <https://medium.com/dream-youngs/google-cloud-console-ssh%EB%A1%9C-%EC%A0%91%EC%86%8D%ED%95%98%EA%B8%B0-for-windows-bf56104a9728>
- <https://koreanfoodie.me/34>

# Q & A

궁금한 거 있으면 연락주세요