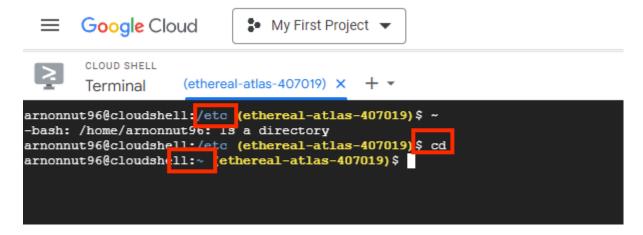
Workshop 3: Basic Cloud

Link: https://console.cloud.google.com/welcome/new?project=ethereal-atlas-407019&authuser=1&cloudshell=true

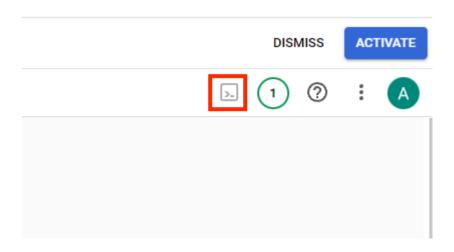
Trick

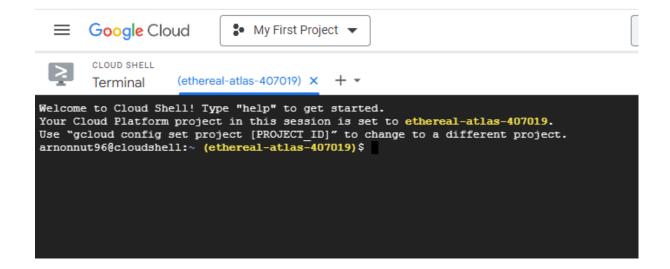
- สามารถพิมพ์ cd เพื่อกลับไปที่หน้า home ได้
- หรือจะพิมพ์เต็มๆคือ cd/home/arnnonnu96



- CTRL + L เพื่อ clear terminal
- CTRL + R reverse search

สร้าง Terminal





ls : list ดูว่าไฟล์มีอะไรบ้าง

ls : list ดูว่ามีไฟล์อะไรบ้าง

คำสั่ง ls สำหรับ list file ใน directory ปัจจุบัน จากในตัวอย่างจะเห็นว่ามี 1 ไฟล์



pwd - print working directory

วิธีการพิมพ์ตัวหนอน (shift + ~) กดค้าง

pwd: print working directory

คำสั่ง pwd สำหรับ print full path ว่าตอนนี้อยู่ที่ directory อะไร



โดยปกติแล้ว เมื่อเริ่มต้น จะเริ่มที่ HOME (/home/username) มักแทนด้วยสัญลักษณ์ ~ (tilde)

Full path ของระบบ Unix จะเริ่มที่ / (root) เสมอ โฟลเดอร์ที่อยู่ภายใต้ root มักเป็น folder ที่เกี่ยวข้องกับระบบ และไม่สามารถแก้ไขด้วยด้วย สิทธิ์ของ user ทั่วไป

Relative path คือ path ที่อ้างอิงจาก path ปัจจุบัน หรือ สามารถอ้างอิงจาก HOME (~) ได้เช่นกัน



DATATH

cd : change directory

คำสั่งสำหรับเปลี่ยน directory ไปยัง path อื่นๆ

cd: change directory

คำสั่ง cd สำหรับการเปลี่ยน directory ไปยัง path อื่นๆ คำสั่งนี้ มักใช้คู่กับ ls และ pwd

```
$ cd [PATH]
$ cd ~
$ pwd
/home/username
```

สัญลักษณ์พิเศษที่ใช้บ่อย



Home (~) คือ directory หลักของ user ใช้ในการอ้างอิง path Root (/) คือ จุดเริ่มต้นของ path โดย full path จะเริ่มจาก / เสมอ

- . แทน directory ปัจจุบันที่ใช้งานอยู่
- .. แทน directory ก่อนหน้าขึ้นไปหนึ่งชั้น (เหมือนถอยออกจากโฟลเดอร์)



Echo - print

echo : print ออกมาบน Terminal

คำสั่ง echo สำหรับการแสดงข้อความหรือตัวแปรออกมาบน Terminal

\$ echo "Hello World!" Hello World! \$ echo \$USER username เกร็ดเสริม: \$Variable หรือ \${var} \$ใช้นำหน้าชื่อตัวแปร

คำสั่ง echo ยังใช้สำหรับการ print ข้อความ ลงไปต่อท้ายไฟล์ โดยใช้เครื่องหมาย >> ได้อีกด้วย

```
$ echo "Hello World! in a file" >> hello.txt
$ ls
README-cloudshell.txt hello.txt
```

cat : concatenate อ่านไฟล์ออกมาบน Terminal

คำสั่ง cat สำหรับ print เนื้อหาในไฟล์ออกมา

```
$ cat [FILE]...
$ cat hello.txt
Hello World! in a file
```

ที่เรียกว่า concatenate เพราะ ถ้าใส่ชื่อไฟล์หลายไฟล์ใน command เดียวกัน จะนำชื่อไฟล์ทั้งหมดมาต่อ กัน และยังใช้ร่วมกับเครื่องหมาย >> เพื่อนำ output ไปใส่ไว้ในไฟล์

```
$ cat hello.txt hello.txt >> new_file.txt
$ cat new_file.txt
Hello World! in a file
Hello World! in a file
```

more / less : เปิดไฟล์เป็นหน้าๆ

more / less : เปิดไฟล์ เป็นหน้า ๆ

บางครั้ง ถ้าไฟล์มีขนาดใหญ่ การใช้คำสั่ง cat สำหรับ print เนื้อหาไฟล์ จะอ่านยาก เพราะจะแสดงผล เนื้อหาของทั้งไฟล์ออกมา ในกรณีนี้ สามารถใช้คำสั่ง more หรือ less แทนได้

```
$ more [FILENAME]
$ less [FILENAME]
```

ทั้งสองคำสั่ง มีความคล้ายคลึงกัน คือ จะแสดงผลเนื้อหาทีละหน้า

- กด space bar เพื่อที่จะดูหน้าถัดไป
- กด q เพื่อออก

ข้อแตกต่างคือ more จะ print เนื้อหาลงใน terminal ทันที เมื่อจบ command จะเห็นเนื้อหาไฟล์นั้นอยู่ ส่วน less จะแยกการแสดงผลออก เป็นอีกหน้า ทำให้ย้อนดูเนื้อหาก่อนหน้าได้ด้วย และมีฟีเจอร์เยอะกว่า



DATATH

Touch - การสร้างไฟล์ว่างเปล่า

touch : สร้างไฟล์เปล่า

คำสั่ง touch สำหรับการสร้างไฟล์เปล่า ขึ้นมาหนึ่งไฟล์

```
$ touch [FILENAME]
$ touch blank.txt
$ cat touch.txt
```

เมื่อลอง cat ดูก็จะไม่เห็นอะไร เพราะว่าเป็นไฟล์เปล่านั่นเอง คำสั่งนี้อาจจะไม่ค่อยได้ใช้มาก แต่ก็มีประโยชน์ในการสร้างไฟล์อย่างรวดเร็ว



mkdir : Makr directory สร้างโฟลเดอร์ใหม่

mkdir : make directory สร้างโฟลเดอร์ใหม่

คำสั่ง mkdir หรือ make directory สำหรับการสร้างโฟลเดอร์ใหม่

```
$ mkdir [DIRECTORY_NAME]
$ mkdir new_folder
$ ls
blank.txt hello.txt new_file.txt new_folder README-cloudshell.txt
```

เมื่อลอง ls ดู จะเห็น new_folder ที่เป็นตัวหนา และสีที่แตกต่าง บ่งบอกว่าเป็น directory mkdir ยังสามารถสร้าง folder ที่ซ้อนกันได้หลายชั้น โดยใช้ -p (p: parent คือสร้าง parent ด้วยถ้าไม่มี)

```
$ mkdir -p new_folder/in/folder
```

cp: การ copy file

cp : copy ก็อปปี้ไฟล์

คำสั่ง cp สำหรับ copy ไฟล์ จากต้นทาง ไปปลายทาง ถ้าปลายทางเป็น directory สามารถ copy หลายไฟล์เข้าไปใน directory นั้นได้

```
$ cp [SOURCE]... [DESTINATION]
$ cp hello.txt new_folder/
$ ls new_folder
in hello.txt
```

ถ้าต้องการ copy โฟลเดอร์ ต้องใส่ -r (r: recursive) เพื่อให้ copy ไปยังทุก ๆ ไฟล์ในโฟลเดอร์ต้นทาง

```
$ cp -r new_folder/ new_folder2
$ ls new_folder2
in hello.txt
```

move - เปลี่ยนชื่อไฟล์

mv : move ย้าย / เปลี่ยนชื่อไฟล์ หรือ directory

คำสั่ง mv หรือ move ใช้สำหรับการย้ายไฟล์ หรือ directory ไปอีกที่หนึ่ง เราใช้ mv ในการเปลี่ยนชื่อได้ด้วย

```
$ mv [SOURCE] [DESTINATION]
$ mv new_file.txt new_folder/ # ย้ายไฟล์เข้าไปใน folder new_folder
$ mv new_folder2 good_name # เปลี่ยนชื่อ folder เป็น good_name
$ ls
blank.txt good_name hello.txt new_folder README-cloudshell.txt
$ ls new_folder
hello.txt in new_file.txt
```

rm - remove การลบข้อมูล

rm : remove ลบไฟล์ หายถาวร

คำสั่ง rm สำหรับ remove เพื่อลบไฟล์ออก

```
$ rm [FILENAME]
$ rm blank.txt
$ ls
$ good_name hello.txt new_folder README-cloudshell.txt
```

ถ้าต้องการลบ directory ต้องใช้ -r (r: recursive) เพื่อให้ไปลบไฟล์ย่อยข้างในด้วย

```
$ rm -r new_folder
$ ls
$ good_name hello.txt README-cloudshell.txt
```

wget : Download ไฟล์

wget : download ไฟล์

คำสั่ง wget สำหรับ การ download ไฟล์จาก URL -0 ใช้สำหรับการตั้งชื่อไฟล์ที่จะโหลดมา

ls *.zip ตัว * คือ wildcard ที่ใช้แทน string อะไรก็ได้ ในที่นี้คือ หาไฟล์อะไรก็ได้.zip แล้วได้ผลลัพธ์ ออกมาเป็น data.zip

> นอกจากคำสั่ง wget ที่ใช้ download file แล้ว ยังมีคำสั่ง <u>curl</u> ที่ใช้ในการ download file และยิง API ได้เช่นกัน

(--.-) **DATAC**

Unzip และ zip ไฟล์

unzip : แตกไฟล์ zip

คำสั่ง unzip สำหรับการแตกไฟล์ zip ด้วย command-line

```
$ unzip [ZIP_FILE]
$ unzip data.zip
Archive: data.zip
inflating: ws2-output.csv
```

zip : สร้างไฟล์ zip

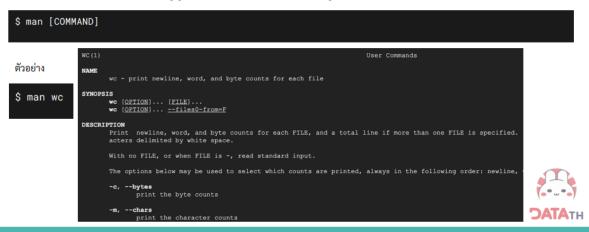
คำสั่ง zip สำหรับการสร้างไฟล์ zip ด้วย command-line

```
$ zip [ZIP_FILE.zip] [FILE]...
$ zip new_data.zip ws2_data.csv
adding: ws2_data.csv (deflated 76%)
```

การใช้ MAN และ VIM

Bonus : man อยากรู้คำสั่งไหน เปิดดูคู่มือได้เลย

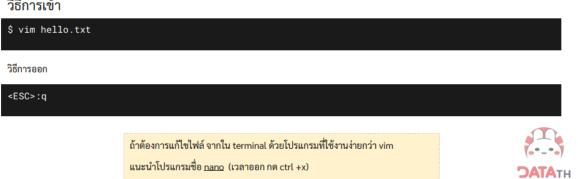
man มาจาก manual ใช้เพื่อดูคู่มือของคำสั่ง นั้น ๆ ได้ เมื่อดูเสร็จ กด q เพื่อ ออก



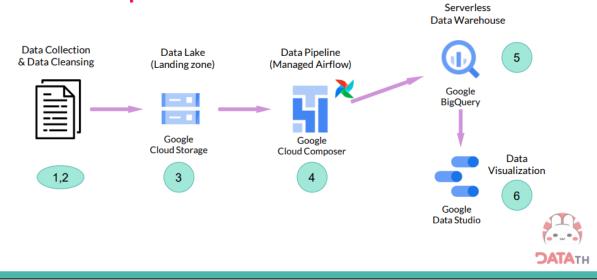
Bonus : ออกจาก vim ยังไง

vim คือ โปรแกรม text editor ใน terminal ที่มีชื่อเสียงโปรแกรมหนึ่ง พัฒนามาจาก vi (vim - Vi IMproved) หนึ่งในชื่อเสียงของ vim คือ เข้าแล้วออกไม่ได้

วิธีการเข้า



Workshops ทั้งหมด





Workshop 3 - Upload to Data Lake

Upload ไฟล์ข้อมูล เข้า Google Cloud Storage ที่เราจะใช้เป็น Data Lake



Prerequisite:

- 1. สมัครสมาชิก Google Cloud
- 2. สรู้าง Project (Google Cloud อาจจะสร้างให้อัตโนมัติ)
- 3. สร้าง Bucket เก็บข้อมูล

Input:

- CSV file จาก workshop 2



- ข้อมูลเข้าไปอยู่ใน Cloud Storage



gsutil คำสั่งในการใช้งานร่วมกับ Google Cloud

gsutil เป็น command ที่มาพร้อมกับ Cloud SDK ที่ต้อง install เพิ่มถ้าต้องการใช้งานผ่าน command line บน local หรือสามารถ ใช้งานผ่าน Cloud Shell ได้ทันที



การอ้างอิง path ใน GCS จะต้องขึ้นต้นด้วย gs:// เสมอ

Google Cloud Storage

Documentation การใช้งาน gsutil: https://cloud.google.com/storage/docs/how-to

gsutil Cheatsheet: https://bit.ly/gsutil-gist

Install Cloud SDK: https://cloud.google.com/sdk/docs/install

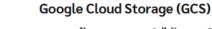


Cloud Storage

การสร้าง Bucket

- สร้าง Bucket บน Google Colud
- สร้างโดยใช้ gsutil

วิธี Upload Data เข้า Google Cloud Storage



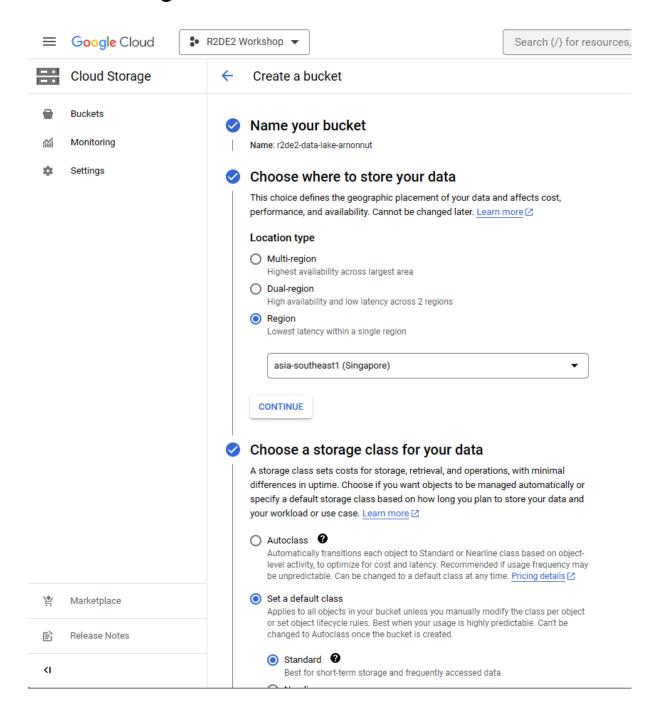
Google Cloud Storage

- 1. สร้าง Bucket ทำได้หลายวิธีดังนี้
 - Cloud Console บนเว็บ UI
 - gsutil command ผ่าน Cloud Shell
- 2. อัปโหลดข้อมูล ทำได้หลายวิธีดังนี้
 - อัปโหลดผ่าน Cloud Console บนเว็บ UI
 - ใช้คำสั่ง gsutil ผ่าน Cloud Shell
 - ใช้โค้ด Python ผ่าน Python SDK library

DATATH

Upload https://cloud.google.com/storage/docs/samples/storage-upload-file#storage_upload_file-python Download https://cloud.google.com/storage/docs/downloading-objects#storage-download-object-python

สร้างบน Google Cloud



Choose a storage class for your data

A storage class sets costs for storage, retrieval, and operations, with minimal differences in uptime. Choose if you want objects to be managed automatically or specify a default storage class based on how long you plan to store your data and your workload or use case. Learn more

Autoclass

Automatically transitions each object to Standard or Nearline class based on object-

Automatically transitions each object to Standard or Nearline class based on object-level activity, to optimize for cost and latency. Recommended if usage frequency may be unpredictable. Can be changed to a default class at any time. Pricing details

Set a default class

Applies to all objects in your bucket unless you manually modify the class per object or set object lifecycle rules. Best when your usage is highly predictable. Can't be changed to Autoclass once the bucket is created.

Best for backups and data accessed less than once a month

Coldline

Best for disaster recovery and data accessed less than once a quarter

Archive Best for long-term digital preservation of data accessed less than once a year

CONTINUE

Choose how to control access to objects

Choose how to control access to objects

Prevent public access

Restrict data from being publicly accessible via the internet. Will prevent this bucket from being used for web hosting. Learn more 🗷

Enforce public access prevention on this bucket

Access control

Uniform

Ensure uniform access to all objects in the bucket by using only bucket-level permissions (IAM). This option becomes permanent after 90 days. Learn more [2]

Fine-grained

Specify access to individual objects by using object-level permissions (ACLs) in addition to your bucket-level permissions (IAM). Learn more ☑

CONTINUE

Choose how to protect object data

Your data is always protected with Cloud Storage but you can also choose from these additional data protection options to prevent data loss. Note that object versioning and retention policies cannot be used together.

Protection tools

- None
- Object versioning (for data recovery)

For restoring deleted or overwritten objects. To minimize the cost of storing versions, we recommend limiting the number of noncurrent versions per object and scheduling them to expire after a number of days. Learn more 🗷

Retention policy (for compliance)

For preventing the deletion or modification of the bucket's objects for a specified minimum duration of time after being uploaded. Learn more 🖸

Data encryption @

- Google-managed encryption key
 - No configuration required
- Customer-managed encryption key (CMEK) Manage via Google Cloud Key Management Service

สร้างโดยใช้ gsutil

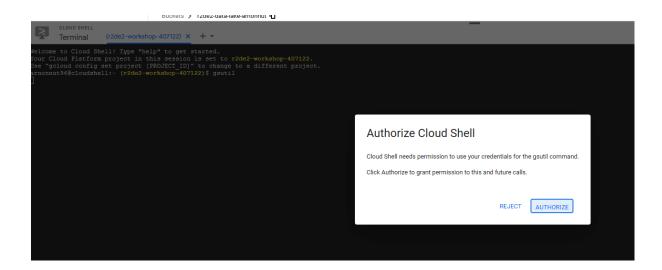
Link: แปล จาก original post

https://gist.github.com/fonylew/480ab35656572c5341510b59785ac2e8 · GitHub

Link: gsutil.sh · GitHub

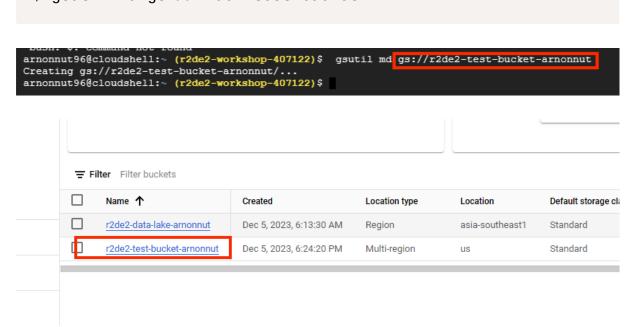
หลังจากสร้างครั้งแรก ระบบจะถามว่าให้ Authorize หรือไม่ หมายถึง gsutil จะขออนุญาตเข้าบนบัญชี google

& gsutil



สร้าง Bucket ใน terminal

\$ gsutil md gs://r2de2-test-bucket

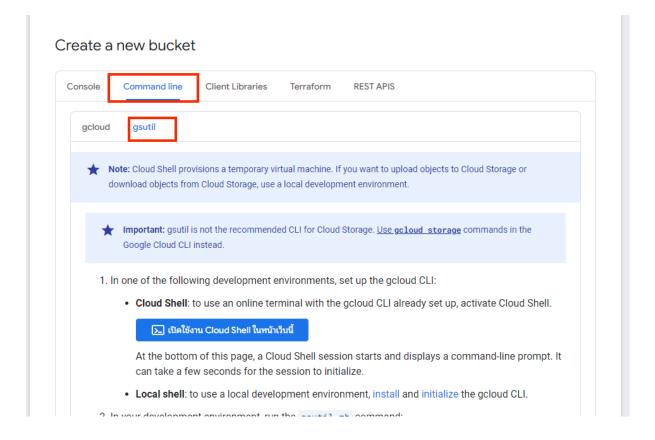


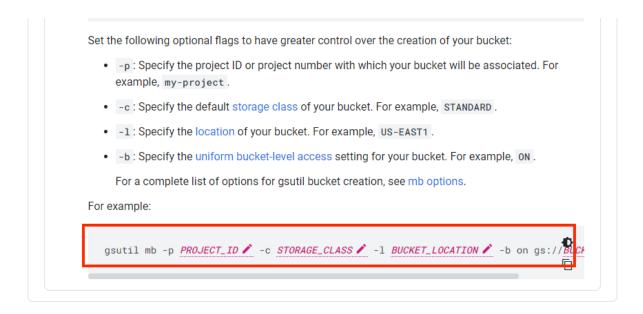


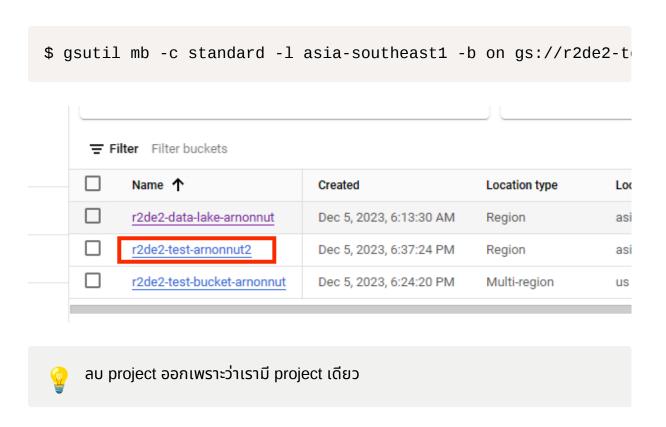
การสร้าง bucket แบบนี้จะสร้างเป็นค่าเริ่มต้น ต้องเพิ่มเงื่อนไขการปรับแจ่ง

สร้าง Bucket gsutil แบบมีเงื่อนไข

link : https://cloud.google.com/storage/docs/creating-buckets#storage-create-bucket-gsutil

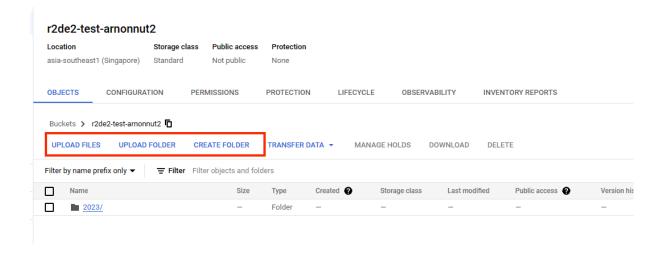






Upload ข้อมูล

Google cloud สามารถ upload ทาง UI ได้



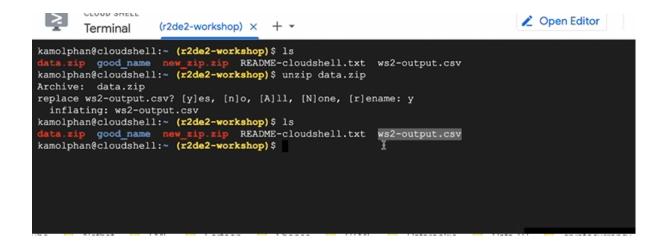
วิธี upload ผ่าน commandline

ไฟล์ที่ต้องการ upload workshop จาก link : https://file.designil.com/bhXYol+

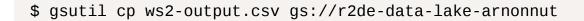
\$ wget -0 data.zip https://file.designil.com/bhXYol+

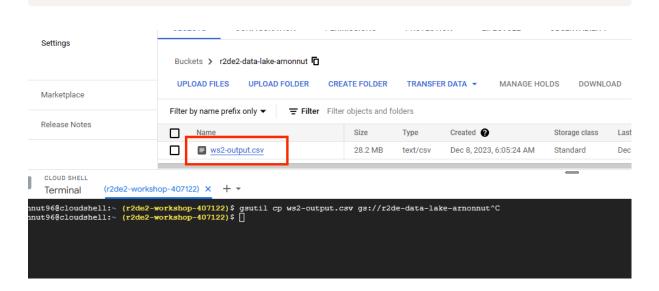
หลังจากนั้นให้ unzip หากมีไฟล์อยู่ระบบจะถามให้ replace หรือไม่ พิมพ์ Y ไป

```
unzip data.zip
```



วิธีการ upload



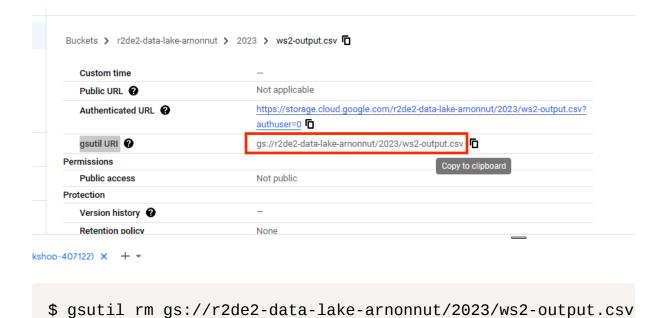


การ upload file โดยการสร้าง Folder ให้ด้วย

\$ gsutil cp ws2-output.csv gs://r2de2-data-lake-arnonnut/2023

วิธี remove

นำ gsutil URI มาใส่

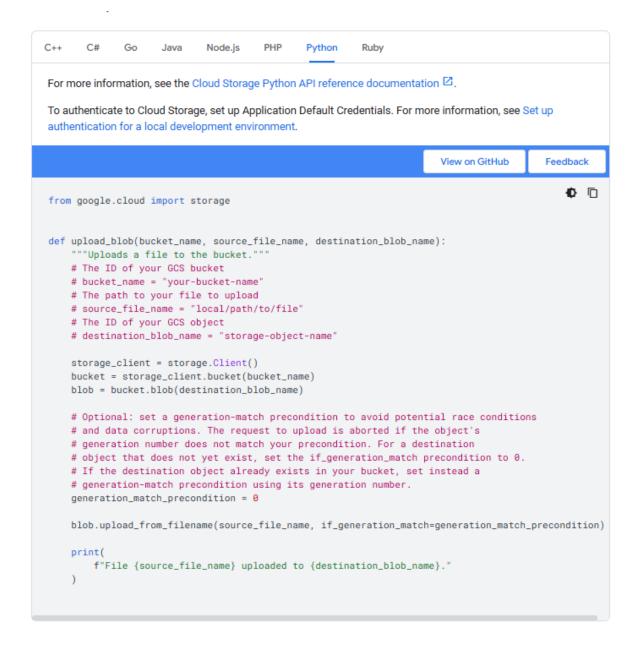


วิธีการ download

\$ gsutil -m cp gs://r2de2-data-lake-arnonnut/2023/ws2-output.

การใช้ Python ผ่าน Python Python SDK library

Link : <u>Upload an object | Cloud Storage | Google Cloud</u>

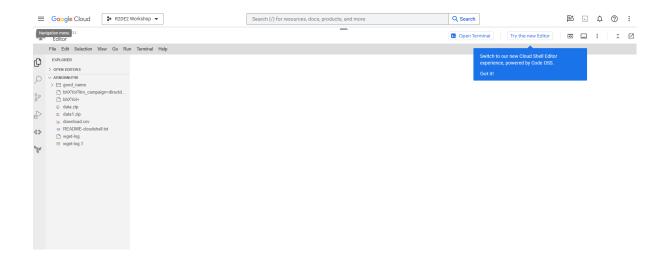


download package ก่อน

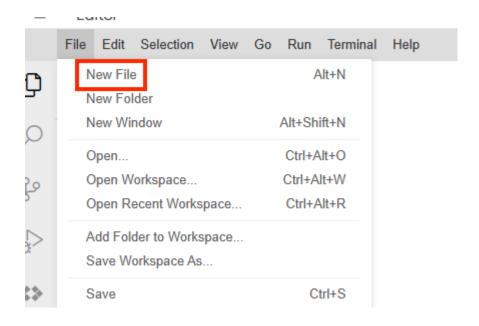
โดยปกติต้อง install ตามข้างบน แต่ใน clould shell ของ google จะลงไว้ให้แล้ว

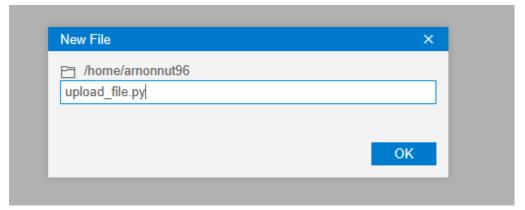
```
$ pip3 install google-cloud-storage
```

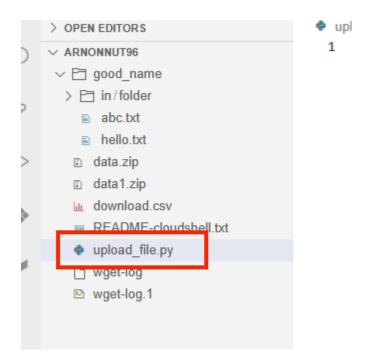
หลังจากนั้นกด Open Terminal



Newfile หลังจากนั้นวาง code ตามที่ copy มา



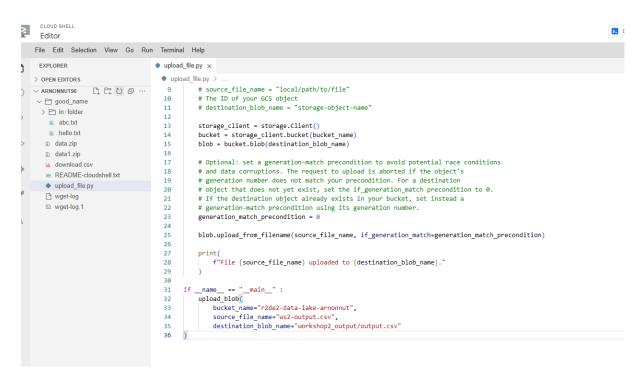




```
from google.cloud import storage
def upload_blob(bucket_name, source_file_name, destination_bl
    """Uploads a file to the bucket."""
    # The ID of your GCS bucket
    # bucket name = "your-bucket-name"
    # The path to your file to upload
    # source_file_name = "local/path/to/file"
    # The ID of your GCS object
    # destination_blob_name = "storage-object-name"
    storage_client = storage.Client()
    bucket = storage_client.bucket(bucket_name)
    blob = bucket.blob(destination_blob_name)
    # Optional: set a generation-match precondition to avoid
    # and data corruptions. The request to upload is aborted
    # generation number does not match your precondition. For
    # object that does not yet exist, set the if_generation_m
    # If the destination object already exists in your bucket
    # generation-match precondition using its generation numb
    generation_match_precondition = 0
```

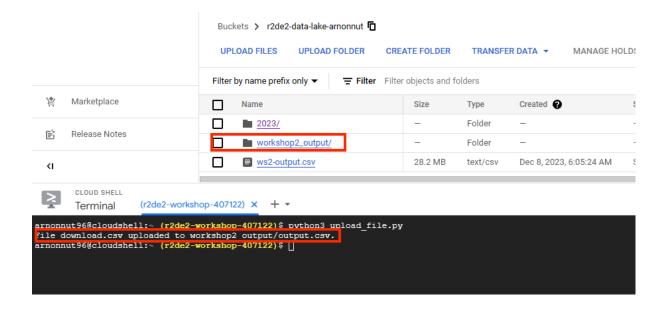
```
blob.upload_from_filename(source_file_name, if_generation_
print(
    f"File {source_file_name} uploaded to {destination_blook
}

if __name__ == "__main__" :
    upload_blob(
        bucket_name="r2de2-data-lake-arnonnut",
            source_file_name="download.csv",
            destination_blob_name="workshop2_output/output.csv"
)
```



วิธีการ run code python จะเห็นได้ว่า fiel ถูก upload แล้ว

```
$ python3 upload_file.py
```



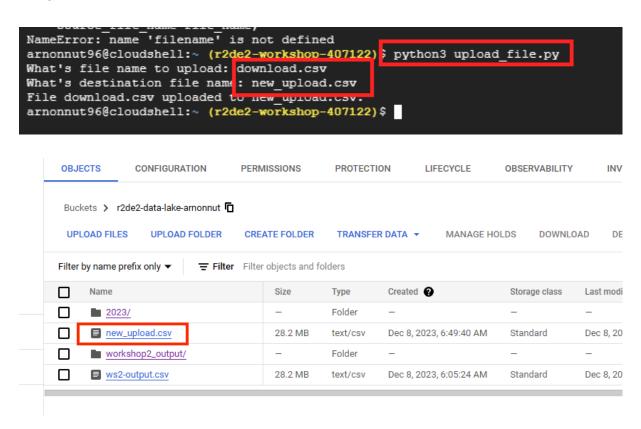
สร้าง input เพื่อสอบถามไฟล์ที่ต้องการ upload บน terminal

```
• upload_file.py > ...
   23
             generation_match_precondition = 0
   24
             blob.upload_from_filename(source_file_name, if_generation_match=generation_match_precondition)
   25
   26
   27
             print(
                f"File {source_file_name} uploaded to {destination_blob_name}."
   28
   29
         if __name__ == "__main__" :
   31
   32
             file_name = input("What's file name to upload: ")
             dest_file_name = input("What's destination file name: ")
   33
   34
   35
             upload_blob(
                 bucket_name="r2de2-data-lake-arnonnut",
   36
   37
                 source_file_name=file_name,
                 destination_blob_name=dest_file_name
   38
                                        ws2-output.csv
                                                                                       28.2 MB
                                                                                                    text/csv
  <I
         CLOUD SHELL
                        (r2de2-workshop-407122) × + ▼
         Terminal
arnonnut96@cloudshell: <a href="mailto:(r2de2-workshop-407122">(r2de2-workshop-407122)</a> python3 upload_file.py
File download.csv uploaded to workshop2 output/output.csv.
                                   2-workshop-407122) python3 upload file.py
What's file name to upload:
```

```
from google.cloud import storage
def upload_blob(bucket_name, source_file_name, destination_bl
    """Uploads a file to the bucket."""
    # The ID of your GCS bucket
    # bucket name = "your-bucket-name"
    # The path to your file to upload
    # source_file_name = "local/path/to/file"
    # The ID of your GCS object
    # destination blob name = "storage-object-name"
    storage client = storage.Client()
    bucket = storage_client.bucket(bucket_name)
    blob = bucket.blob(destination_blob_name)
    # Optional: set a generation-match precondition to avoid
    # and data corruptions. The request to upload is aborted
    # generation number does not match your precondition. For
    # object that does not yet exist, set the if_generation_m
    # If the destination object already exists in your bucket
    # generation-match precondition using its generation numb
    generation_match_precondition = 0
    blob.upload_from_filename(source_file_name, if_generation_
    print(
        f"File {source_file_name} uploaded to {destination_bl
    )
if __name__ == "__main__" :
    file_name = input("What's file name to upload: ")
    dest_file_name = input("What's destination file name: ")
    upload_blob(
        bucket_name="r2de2-data-lake-arnonnut",
        source_file_name=file_name,
```

```
destination_blob_name=dest_file_name
)
```

ลอง upload ไฟล์ download.csv ขึ้น Cloud โดยใช้ชื่อ new_upload.csv จากรูปจะเห็นได้ว่าไฟล์ upload ขึ้นมาแล้ว



เพิ่มเงื่อนไขการการ Download เข้าไปจาก code เดิม และแก้ code เงื่อนไขเพิ่มเติม

 $copy\ code: \underline{https://cloud.google.com/storage/docs/downloading-objects\#storage-\\ \underline{download-object-portion-python}$

```
Editor
 File Edit Selection View Go Run Terminal Help
                                 workshop3.py × • wtest.py(Deleted)
                                    workshop3.py >
→ ARNONNUT96
                                    57
    > P good name
                                    58 if __name_
                                                     == "__main__" :
                                           upload = input ("Upload (u) or Download (d)?")
file_name = input("What's local name to : ")
      data.zip
     data1.zip
                                    60
                                    61
                                                 gcs_file_name = input("What's gcs file name: ")
      README-cloudshell.txt
                                               bucket name ="r2de2-data-lake-arnonnut"

□ waet-loa

                                     64
      wget-log.1
                                                 if upload.lower() =="upload" or upload.lower() =="u":
                                     65
    workshop3.py
                                    66
                                                     upload_blob(
                                                         bucket_name=bucket_name,
                                     68
                                                         source_file_name=file_name
                                     69
                                                        destination_blob_name=gcs_file_name
                                                 elif upload.lower() =="download" or upload.lower() =="d":
                                                     download_blob(
                                     72
                                     73
                                                        bucket_name=bucket_name
                                                         source_blob_name=gcs_file_name,
                                     75
                                                         destination_file_name=file_name
                                     76
                                     77
                                                  else:
                                                     print("Please input upload (u) or download (d)")
                                   ⚠ Problems
                                                 > Python ×
                                  Welcome to Cloud Shell! Type "help" to get started.
                                  Vour Cloud Platform project in this session is set to r2de2-workshop-407122.

Use "gcloud config set project [PROJECT_ID]" to change to a different project.
```

```
from google.cloud import storage

def download_blob(bucket_name, source_blob_name, destination_"""Downloads a blob from the bucket."""
  # The ID of your GCS bucket
  # bucket_name = "your-bucket-name"

# The ID of your GCS object
  # source_blob_name = "storage-object-name"

# The path to which the file should be downloaded
  # destination_file_name = "local/path/to/file"

storage_client = storage.Client()

bucket = storage_client.bucket(bucket_name)

# Construct a client side representation of a blob.
  # Note `Bucket.blob` differs from `Bucket.get_blob` as it
  # any content from Google Cloud Storage. As we don't need
```

```
# using `Bucket.blob` is preferred here.
    blob = bucket.blob(source_blob_name)
    blob.download_to_filename(destination_file_name)
    print(
        "Downloaded storage object {} from bucket {} to local
            source_blob_name, bucket_name, destination_file_n
        )
    )
def upload_blob(bucket_name, source_file_name, destination_bl
    """Uploads a file to the bucket."""
    # The ID of your GCS bucket
    # bucket name = "your-bucket-name"
    # The path to your file to upload
    # source file name = "local/path/to/file"
    # The ID of your GCS object
    # destination_blob_name = "storage-object-name"
    storage_client = storage.Client()
    bucket = storage client.bucket(bucket name)
    blob = bucket.blob(destination_blob_name)
    # Optional: set a generation-match precondition to avoid
    # and data corruptions. The request to upload is aborted
    # generation number does not match your precondition. For
    # object that does not yet exist, set the if_generation_m
    # If the destination object already exists in your bucket
    # generation-match precondition using its generation numb
    generation_match_precondition = 0
    blob.upload_from_filename(source_file_name, if_generation_
    print(
        f"File {source_file_name} uploaded to {destination_bl
    )
if __name__ == "__main__" :
```

```
upload = input ("Upload (u) or Download (d)?")
file name = input("What's local name to : ")
gcs_file_name = input("What's gcs file name: ")
bucket name ="r2de2-data-lake-arnonnut"
if upload.lower() =="upload" or upload.lower() =="u":
    upload_blob(
        bucket_name=bucket_name,
        source_file_name=file_name,
        destination_blob_name=gcs_file_name
elif upload.lower() =="download" or upload.lower() ==
    download_blob(
        bucket_name=bucket_name,
        source blob name=gcs file name,
        destination_file_name=file_name
else:
    print("Please input upload (u) or download (d)")
```

Test Download File

จะเห็นได้ว่าเรา download file ชื่อว่า new_download.csv av terminal โดยใช้ชื่อว่า workshop3 dowload.csv ขนาดไฟล์จะเท่ากัน