# Requirement

**ทำบน Google cloud platform ใน Project ที่เตรียมไว้ให้เท่านั้น**

1. สร้าง Data Lake ด้วย Google Storage
   ติดตั้ง Apache Airflow เพื่อทำ scheduled job โดยให้ทำงานทุก 1 ชั่วโมง
   สำหรับ Clone ข้อมูลจาก ฐานข้อมูลตั้งต้น (users table, user_log table)
   ไปเก็บแบบ Raw data ที่ Data Lake

2. สร้าง Data Warehouse แล้วเชื่อมต่อเข้ากับ Google Big Query
   เพิ่ม job ใน Apache Airflow เพื่อทำการ Transform ข้อมูลจาก Data Lake แล้วเก็บข้อมูลที่แปลงแล้วที่ Data Warehouse โดยให้งานต่อหลังจาก job load ข้อมูลเข้า data lake เสร็จสิ้น

   โดยสิ่งที่ต้อง Transform คือ Table ของ user_log
   ให้เปลี่ยนชื่อ column จาก status เป็น success และเก็บข้อมูลเป็นชนิด Boolean
   ทำการแปลงข้อมูล 0 เป็นค่า FALSE และแปลงข้อมูล 1 เป็นค่า TRUE

3. ติดตั้ง Apache Spark
   ติดตั้ง Jupyter Notebook สำหรับเขียนโค้ด Python3
   ติดตั้ง library PySpark, Pandas, Numpy

4. เขียนโค้ด Python3 ที่ Jupyter Notebook โดยใช้ PySpark เรียกใช้งาน BigQuery เพื่ออ่านข้อมูลจาก Data Warehouse สำหรับเป็นตัวอย่างการใช้งานให้แก่ Data Scientist

5. ส่งมอบ
   - file code ทั้งหมดที่เขียนผ่าน Git
   - document อธิบายว่าใช้ server ตัวไหนบ้าง มีการติดตั้งหรือเก็บไฟล์ไว้ที่ใดบ้าง
   - ลิงค์สำหรับเข้าใช้งาน Jupyter Notebook

## Design


การทำงานจะเริ่มจาก Query ข้อมูล 2 tables ได้แก่ USERS และ USERLOG และทำการเก็บไฟล์ของข้อมูลที่ทำการ Query เป็นไฟล์ .csv และทำการ upload ไฟล์ของข้อมูลไปยัง google cloud storage หลังจากนั้น จะทำการนำเอาไฟล์ excel ที่ upload ไปบน google cloud storage มาทำการประมวลผลตามเงื่อนไขที่กำหนดและ SAVE เป็นไฟล์ใหม่เป็นไฟล์ .csv หลังจากได้ข้อมูลที่ผ่านการประมวลแล้วจะทำการอัพโหลดไฟล์กลับไปยัง google cloud storage อีกครั้งแต่จะอยู่กันคนละ Folder บน bucket ของ google cloud storage โดยขั้นตอนการทำงานทั้งหมดนั้นจะควบคุมการทำงานโดยใช้ Apache Airflow ควบคู่กับ Code Python

Buckets > thanyaboon-bluepi-gcs > RAWDATA > PROFILE > USERS

UPLOAD FILES | UPLOAD FOLDER | CREATE FOLDER | MANAGE HOLDS | DOWNLOAD | DELET

Filter by name prefix only | Filter | Filter objects and folders

| Name | Size | Type | Created time |
|---|---|---|---|
| users_2021_06_14_24:12.csv | 315 B | text/csv | Jun 15, 2021, 12:12:16 AM |
| users_2021_06_14_24:2.csv | 315 B | text/csv | Jun 15, 2021, 12:02:21 AM |
| users_2021_06_14_24:24.csv | 315 B | text/csv | Jun 15, 2021, 12:24:11 AM |
| users_2021_06_14_24:26.csv | 315 B | text/csv | Jun 15, 2021, 12:26:40 AM |
| users_2021_06_14_24:27.csv | 315 B | text/csv | Jun 15, 2021, 12:27:50 AM |
| users_2021_06_14_24:28.csv | 315 B | text/csv | Jun 15, 2021, 12:28:48 AM |
| users_2021_06_14_24:30.csv | 315 B | text/csv | Jun 15, 2021, 12:30:43 AM |
| users_2021_06_14_24:4.csv | 315 B | text/csv | Jun 15, 2021, 12:04:59 AM |
| users_2021_06_14_24:9.csv | 315 B | text/csv | Jun 15, 2021, 12:09:21 AM |

จากรูปแสดงข้อมูลดิบที่ได้จาก table users

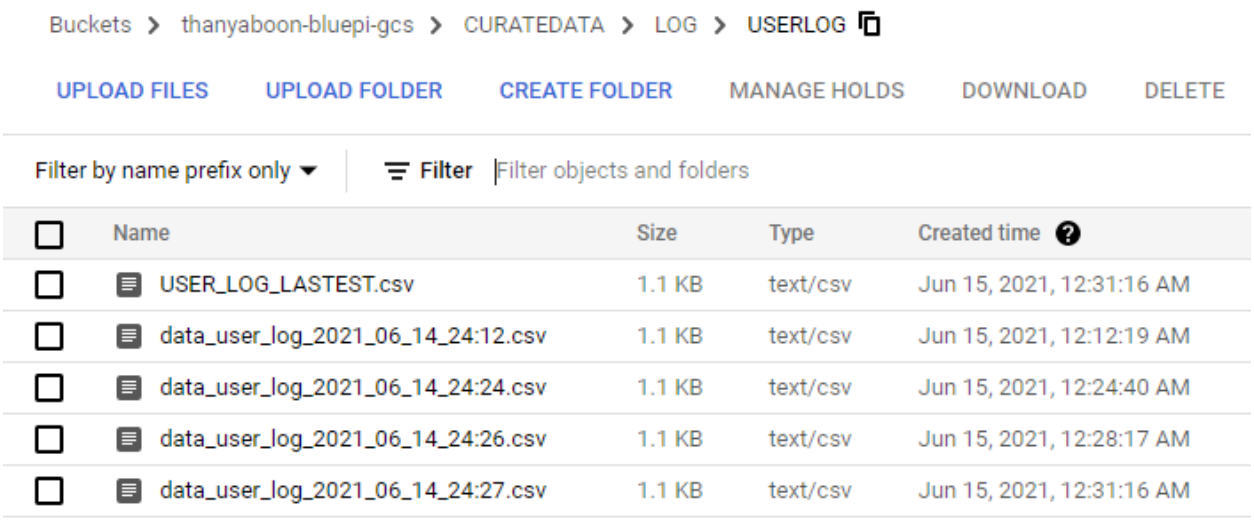
Buckets > thanyaboon-bluepi-gcs > RAWDATA > LOG > USERLOG

UPLOAD FILES | UPLOAD FOLDER | CREATE FOLDER | MANAGE HOLDS | DOWNLOAD | DELETE

Filter by name prefix only | Filter | Filter objects and folders

| Name | Size | Type | Created time |
|---|---|---|---|
| user_log_2021_06_14_24:12.csv | 1.1 KB | text/csv | Jun 15, 2021, 12:12:17 AM |
| user_log_2021_06_14_24:2.csv | 1.1 KB | text/csv | Jun 15, 2021, 12:02:22 AM |
| user_log_2021_06_14_24:24.csv | 1.1 KB | text/csv | Jun 15, 2021, 12:24:12 AM |
| user_log_2021_06_14_24:26.csv | 1.1 KB | text/csv | Jun 15, 2021, 12:26:41 AM |
| user_log_2021_06_14_24:27.csv | 1.1 KB | text/csv | Jun 15, 2021, 12:27:51 AM |
| user_log_2021_06_14_24:28.csv | 1.1 KB | text/csv | Jun 15, 2021, 12:28:49 AM |
| user_log_2021_06_14_24:30.csv | 1.1 KB | text/csv | Jun 15, 2021, 12:30:44 AM |
| user_log_2021_06_14_24:4.csv | 1.1 KB | text/csv | Jun 15, 2021, 12:05:00 AM |
| user_log_2021_06_14_24:9.csv | 1.1 KB | text/csv | Jun 15, 2021, 12:09:21 AM |

จากรูปแสดงข้อมูลดิบที่ได้จาก table user_log




จากรูปแสดงข้อมูลจาก table user_log ที่ได้ทำการ ประมวลผลเรียบร้อยแล้ว ไฟล์ USER_LOG_LASTEST คือ ไฟล์ข้อมูลล่าสุดที่ทำการประมวลผลเสร็จสิ้น

## Service

Service ของ google cloud platform ที่ใช้งานทั้งหมดประกอบด้วย

1. VM instance จำนวน 1 เครื่อง
   - Name : thanyaboon-bluepi-vm
   - Internal IP : 10.0.0.3
   - External IP : 35.198.209.234 (อาจจะมีการเปลี่ยนเมื่อทำการ start vm)

   1.1 Package ภายใน VM instance ประกอบด้วย

   - Virtual VM Airflow ใช้ในการลง Package Apache Airflow สำหรับการทำงานของ Airflow และ Start Web UI ของ Airflow รวมไปถึงการ Start Airflow Scheduler ในการลง Apache Airflow นั้นจำเป็นที่จะต้องสร้าง postgre db (รายละเอียดของ Database มีการอธิบายในข้อถัดไป)

คำสั่งที่ใช้ในการทำงาน Apache Airflow มีดังนี้

```
sudo su airflow
cd /srv/airflow
source bin/activate
export AIRFLOW_HOME=/srv/airflow
```

จากรูป code ส่วนนี้ใช้การเข้าถึง VM ของ Apache Airflow

```
airflow webserver -p 8080
```

```
airflow scheduler
```

จากภาพเป็นคำสั่งที่ใช้ในการ Start Web UI และ Scheduler ของ Airflow

ถ้าผู้ใช้งานต้องการสร้าง Airflow pipeline ให้ผู้ใช้งานนำไฟล์ python ไปวางที่ path /srv/airflow/dags และ DAG ID จากไฟล์ python จะไปปรากฏบนหน้า web ui

(User สำหรับเข้าใช้งาน airflow บนหน้า web ui คือ

username : admin

password : P@ssw0rd)

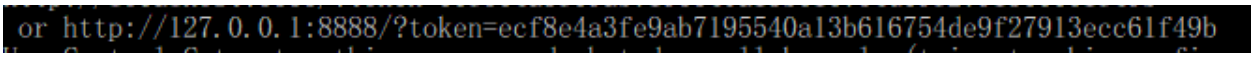
- Virtual VM Jupyter ใช้สำหรับ start server ของ jupyter และเก็บ package ทั้งหมดที่จำเป็นต่อการใช้งานของ python รวมไปถึง library ต่างๆ ที่ผู้ใช้งานต้องการ โดยคำสั่งที่ใช้ในการทำงานมีดังนี้

pipenv shell

jupyter notebook --ip=0.0.0.0 --port=8888 --no-browser &

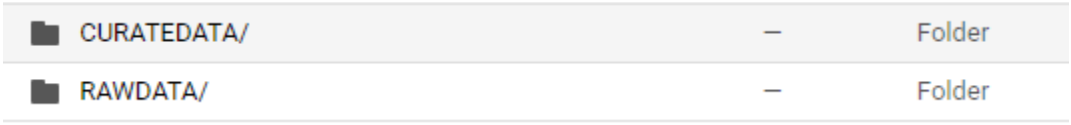
เมื่อ server เริ่มรันแล้วให้ไปที่ browser บนเครื่อง และ ใช้ url : external ip:8888 ในการเข้าถึงหน้า web ui ของ jupyter เช่น **35.198.209.234:8888** (อาจมีการเปลี่ยนแปลง

External ip เมื่อมีการ start VM ใหม่ ) หลังจากที่เข้าไปยัง IP ที่กล่าวมาข้างต้นแล้วให้ทำการคัดลอก token ที่แสดงอยู่บน Shell มาใส่ในหน้า web ui เพื่อทำการเข้าใช้งาน Jupyter




ภายใน VM Instance และ Vertual VM jupyter ได้ทำการลง python และ apache pyspark รวมไปถึง library ที่จำเป็นต่อการใช้งาน และ library ตามที่โจทย์กำหนดเรียบร้อย

2. Bucket ที่ใช้สำหรับเก็บไฟล์ข้อมูลจาก postges db จำนวน 1 bucket
   - Name : thanyaboon-bluepi-gcs
   - ภายใน bucket จะประกอบไปด้วย 2 folder ได้แก่ CURATEDATA และ RAWDATA ดังรูป




   - CURATEDATA ใช้สำหรับเก็บข้อมูลที่ผ่านการประมวลผลเรียบร้อยแล้ว ภายใน CURATEDATA จะมีอีก Folder LOG ซึ่งใช้สำหรับแยกประเภทของข้อมูล และภายในของ LOG ขะมี Folder USERLOG ที่ใช้ในการเก็บข้อมูลหลังจากประมวลผลของ table user_log
   - RAWDATA ใช้สำหรับเก็บข้อมูลที่ยังไม่ได้ประมวลผล (ข้อมูลดิบจาก DATABASE) ภายในของ RAWDATA จะประกอบไปด้วย 2 Folder ได้แก่ LOG และ PROFILE ใช้สำหรับแยกประเภทของข้อมูลดิบตามแต่ข้อมูลของ table ซึ่งภายใน Folder LOG จะมี folder USERLOG อยู่ในภายเช่นเดียวกับ folder PROFILE จะมี folder USERS อยู่ภายใน

3. SQL Instance จำนวน 1 instance ใช้สำหรับเก็บข้อมูลของ Apache Airflow โดย database ที่เลือก คือ Postgres Database
   - Name : thanyaboon-bluepi-de-exam:asia-southeast1:airflow-db
   - Instance ID : airflow-db
   - Plubic IP : 34.87.29.160
   - Private IP : 172.19.32.3
4. Bigquery ที่ใช้งานจำนวน 1 dataset ใช้สำหรับแสดงผลข้อมูลที่ทำการประมวลผลเรียบร้อย
   - Project Name : thanyaboon-bluepi-de-exam
   - Dataset Name :LOG
   - Table Name : USER_LOG


## Table schema

Filter Enter property name or value

| Field name | Type | Mode | Policy Tags | Description |
|---|---|---|---|---|
| created_at | TIMESTAMP | NULLABLE | | |
| updated_at | TIMESTAMP | NULLABLE | | |
| id | STRING | NULLABLE | | |
| user_id | STRING | NULLABLE | | |
| action | STRING | NULLABLE | | |
| Success | BOOLEAN | NULLABLE | | |