

การบันทึกข้อมูลสถานการณ์ผู้ติดเชื้อ COVID-19 ประจำวัน
โดยใช้ Apache Airflow และ MySQL ด้วย Docker



ผู้จัดทำ

นายวัฒนา แซ่อึ้ง รหัสนักศึกษา 63606011

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
King Mongkut's Institute of Technology Ladkrabang

เสนอ

ดร.ชยานนท์ ทรัพย์อาภา

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยา
ศาสตร์มหาบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ประจำปีการศึกษา 2564

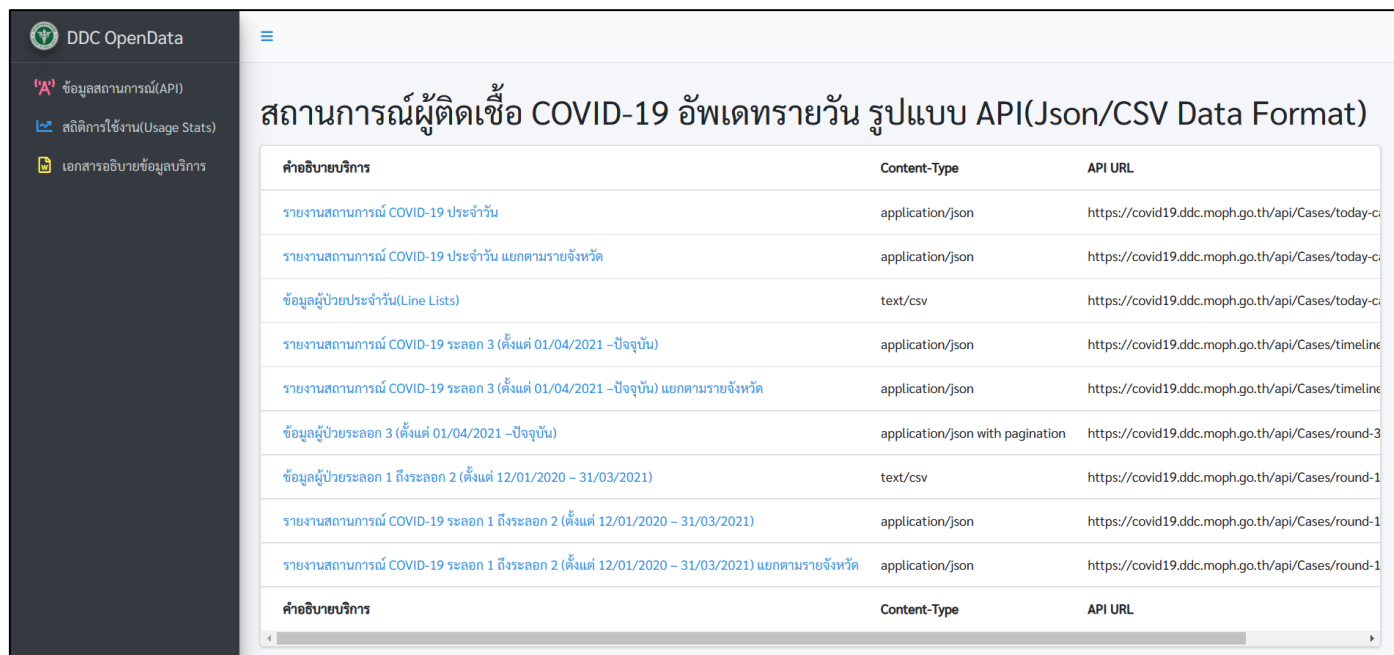
บทนำ

สถานการณ์ผู้ติดเชื้อ COVID-19 ภายในประเทศไทยมียอดผู้ติดเชื้อรวมมากกว่า 2 ล้านคน ซึ่งอยู่ในเกณฑ์ที่น่าเป็นห่วง และยังคงติดตามสถานการณ์อย่างใกล้ชิด ข้อมูลตัวเลขสถิติต่างๆที่เก็บบันทึกไว้จึงมีความสำคัญอย่างมาก เช่น จำนวนผู้ติดเชื้อรวม จำนวนผู้ติดเชื้อรายใหม่ หรือจำนวนผู้เสียชีวิต เนื่องจากข้อมูลนี้สามารถนำมาวิเคราะห์ เพื่อใช้วางแผน และคาดการณ์สถานการณ์ล่วงหน้าได้ การบันทึกข้อมูลประจำวันจึงมีความสำคัญอย่างมาก ดังนั้น ผู้จัดทำจึงมีความคิดที่จะออกแบบระบบสำหรับบันทึกข้อมูลผู้ติดเชื้อ COVID-19 ประจำวัน โดยประกอบไปด้วยหัวข้อดังนี้

1. การเตรียมข้อมูล
2. การออกแบบระบบ
3. สรุปผล

1. การเตรียมข้อมูล

ขั้นตอนการเตรียมข้อมูล ผู้จัดทำได้ค้นหาข้อมูลผู้ติดเชื้อ COVID-19 จากเว็บไซต์กรมควบคุมโรค กระทรวงสาธารณสุข <https://covid19.ddc.moph.go.th> ดังรูปที่ [1.1] ซึ่งเป็นข้อมูลที่มีความความถูกต้อง และน่าเชื่อถือ และเลือกเอาเฉพาะข้อมูลผู้ติดเชื้อ COVID-19 ประจำวันเท่านั้น ผ่านรูปแบบ API โดยโครงสร้างข้อมูลเป็นแบบ JSON Format

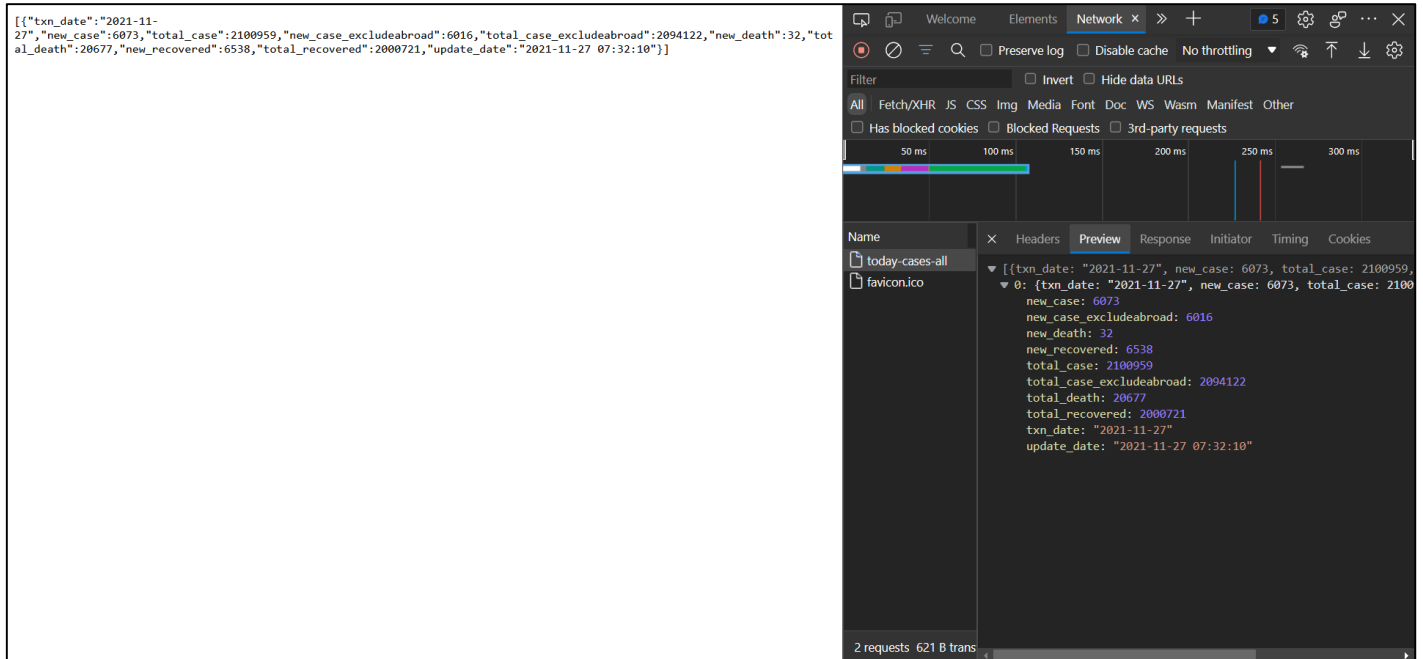


The screenshot shows the 'DDC OpenData' website. On the left is a dark sidebar with navigation links: 'ข้อมูลสถานการณ์(API)', 'สถิติการใช้งาน(Usage Stats)', and 'เอกสารอธิบายข้อมูลบริการ'. The main content area has a title 'สถานการณ์ผู้ติดเชื้อ COVID-19 อัปเดตรายวัน รูปแบบ API(Json/CSV Data Format)'. Below the title is a table with three columns: 'คำอธิบายบริการ', 'Content-Type', and 'API URL'. The table lists various data services for COVID-19, including daily cases, cases by province, and cases by round, with links to the corresponding API endpoints.

คำอธิบายบริการ	Content-Type	API URL
รายงานสถานการณ์ COVID-19 ประจำวัน	application/json	https://covid19.ddc.moph.go.th/api/Cases/today-c
รายงานสถานการณ์ COVID-19 ประจำวัน แยกตามรายจังหวัด	application/json	https://covid19.ddc.moph.go.th/api/Cases/today-c
ข้อมูลผู้ป่วยประจำวัน(Line Lists)	text/csv	https://covid19.ddc.moph.go.th/api/Cases/today-c
รายงานสถานการณ์ COVID-19 ระลอก 3 (ตั้งแต่ 01/04/2021 – ปัจจุบัน)	application/json	https://covid19.ddc.moph.go.th/api/Cases/timeline
รายงานสถานการณ์ COVID-19 ระลอก 3 (ตั้งแต่ 01/04/2021 – ปัจจุบัน) แยกตามรายจังหวัด	application/json	https://covid19.ddc.moph.go.th/api/Cases/timeline
ข้อมูลผู้ป่วยระลอก 3 (ตั้งแต่ 01/04/2021 – ปัจจุบัน)	application/json with pagination	https://covid19.ddc.moph.go.th/api/Cases/round-3
ข้อมูลผู้ป่วยระลอก 1 ถึงระลอก 2 (ตั้งแต่ 12/01/2020 – 31/03/2021)	text/csv	https://covid19.ddc.moph.go.th/api/Cases/round-1
รายงานสถานการณ์ COVID-19 ระลอก 1 ถึงระลอก 2 (ตั้งแต่ 12/01/2020 – 31/03/2021)	application/json	https://covid19.ddc.moph.go.th/api/Cases/round-1
รายงานสถานการณ์ COVID-19 ระลอก 1 ถึงระลอก 2 (ตั้งแต่ 12/01/2020 – 31/03/2021) แยกตามรายจังหวัด	application/json	https://covid19.ddc.moph.go.th/api/Cases/round-1
คำอธิบายบริการ	Content-Type	API URL

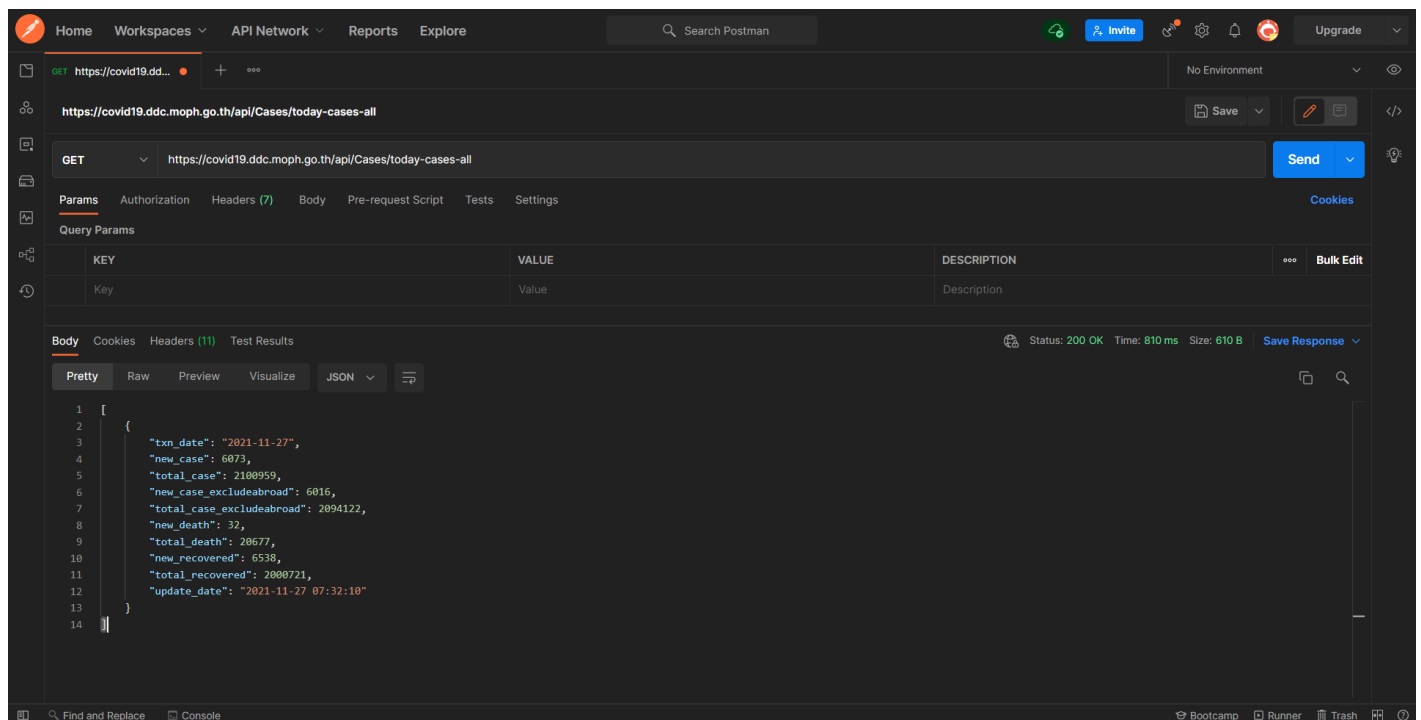
รูปที่ 1.1 เว็บไซต์กรมควบคุมโรค กระทรวงสาธารณสุข

สามารถดึงข้อมูลผ่านเว็บไซต์โดยคลิกที่ลิงก์ <https://covid19.ddc.moph.go.th/api/Cases/today-cases-all> ดังรูปที่[1.2] หรือดึงข้อมูลผ่านโปรแกรม Postman ดังรูปที่[1.3] เพื่อทดสอบว่าข้อมูลมีความพร้อมใช้งานหรือไม่



รูปที่1.2 การดึงข้อมูลผ่านเว็บไซต์

<https://covid19.ddc.moph.go.th/api/Cases/today-cases-all>



รูปที่1.3 การดึงข้อมูลผ่านโปรแกรม Postman

โดยเว็บไซต์กรมควบคุมโรค กระทรวงสาธารณสุขจะมีเอกสารคำอธิบายข้อมูลแต่ละบริการ ดังตารางที่ [1.1] เพื่อให้ผู้ใช้งานสามารถเข้าไปดูรายละเอียดข้อมูลได้

เอกสารคำอธิบายข้อมูลแต่ละบริการ

คำอธิบาย: ข้อมูลจำนวนผู้ป่วยระลอก 1 ถึงระลอก 2 (ตั้งแต่ 12/01/2020 – 31/03/2021)

Column Name	Data Type	comment
txn_date	date	วันแถลง
province	Varchar(100)	จังหวัดกักกัน
new_case	Int	จำนวนผู้ป่วยรายใหม่
total_case	Int	จำนวนผู้ป่วยสะสม
new_case_excludeabroad	Int	จำนวนผู้ป่วยรายใหม่(ไม่นับมาจากต่างประเทศ)
total_case_excludeabroad	Int	จำนวนผู้ป่วยสะสม(ไม่นับมาจากต่างประเทศ)
new_death	Int	จำนวนผู้ป่วยตายรายใหม่
total_death	Int	จำนวนผู้ป่วยตายสะสม
new_recovered	Int	จำนวนผู้ป่วยรักษาหายรายใหม่
total_recovered	Int	จำนวนผู้ป่วยรักษาหายสะสม

ตารางที่ 1.1 เอกสารคำอธิบายข้อมูลแต่ละบริการ

2. การออกแบบระบบ

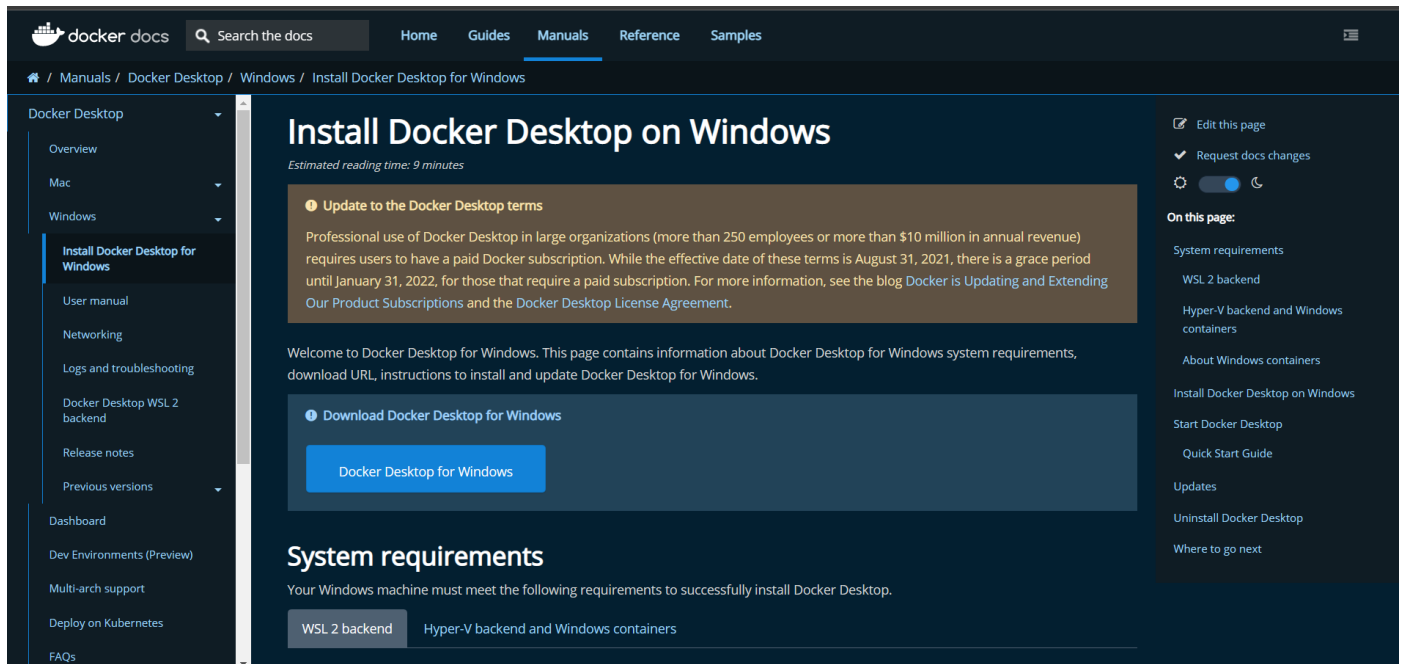
ในการออกแบบระบบ ผู้จัดทำได้ใช้ Docker ซึ่งเป็นแพลตฟอร์มซอฟต์แวร์ที่ช่วยให้สร้าง ทดสอบ และติดตั้งแอปพลิเคชันใช้จริงได้อย่างรวดเร็ว โดยติดตั้งซอฟต์แวร์ Apache Airflow เพื่อใช้สำหรับงานด้านวิศวกรรมข้อมูล และฐานข้อมูล MySQL โดยมีขั้นตอนดังนี้

- 2.1 การติดตั้งซอฟต์แวร์ Docker, Apache Airflow และ MySQL
- 2.2 การเขียนโปรแกรมบันทึกข้อมูลผู้ติดเชื้อ COVID-19 ด้วยภาษา Python
- 2.3 การทดสอบระบบ

2.1 การติดตั้งซอฟต์แวร์ Docker, Apache Airflow และ MySQL

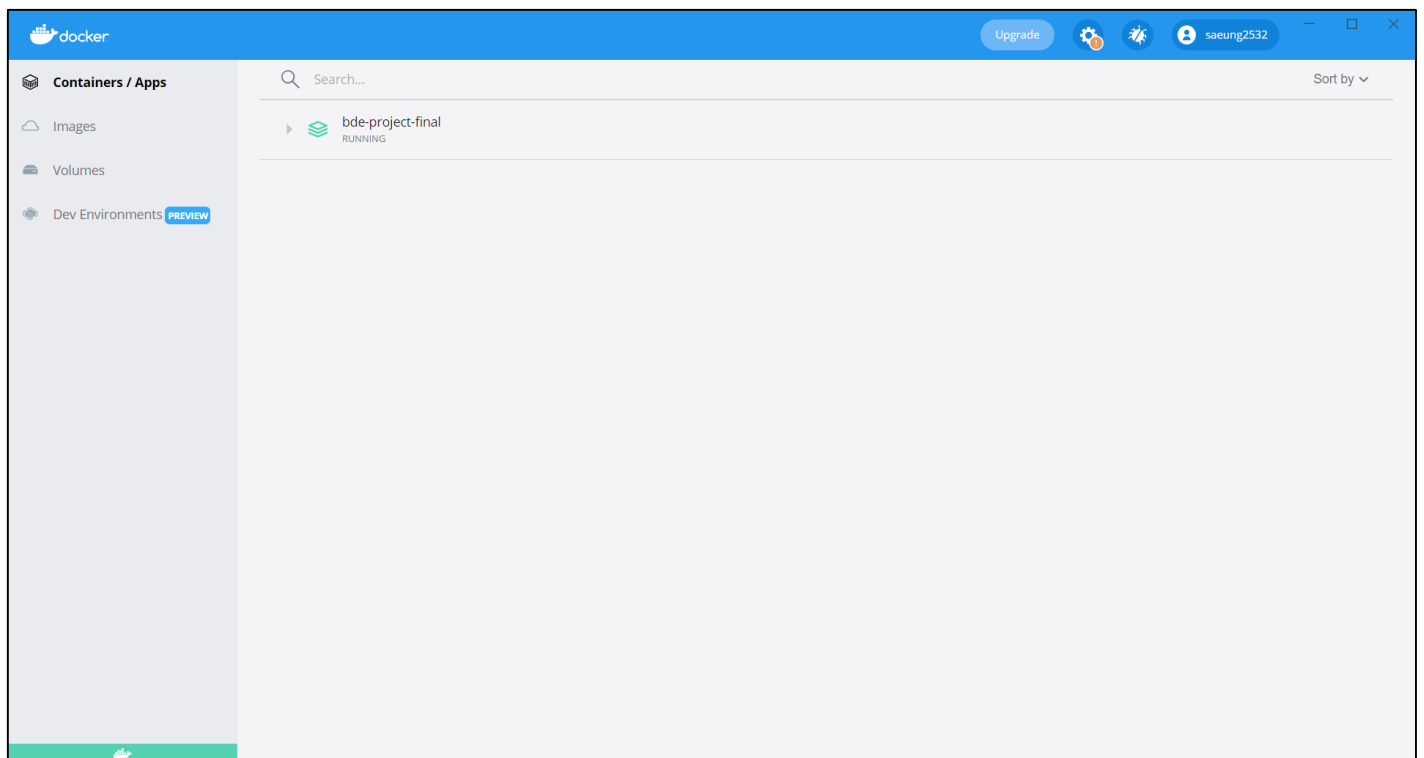
2.1.1 ดาวน์โหลดและติดตั้งซอฟต์แวร์ Docker ผ่านเว็บไซต์ ดังรูปที่[2.1]

<https://docs.docker.com/desktop/windows/install/>



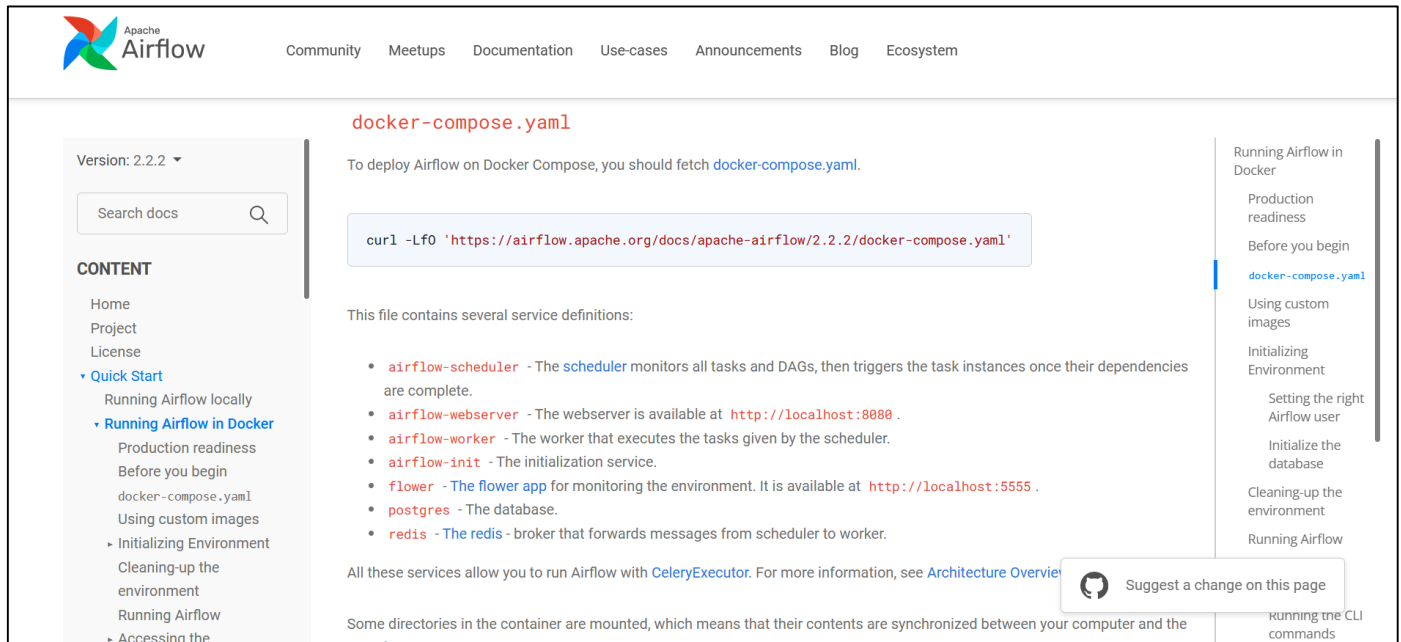
รูปที่ 2.1 ดาวน์โหลด Docker ผ่านเว็บไซต์

2.1.2 หลังจากติดตั้งเสร็จแล้ว เปิดโปรแกรมโดยคลิกที่ Docker Desktop ดังรูปที่ [2.2] โดยส่วนการทำงานหลักของ Docker ประกอบไปด้วย Container, Image และ Volumes



รูปที่ 2.2 โปรแกรม Docker Desktop

2.1.3 ตั้งซอฟต์แวร์ Apache Airflow และฐานข้อมูล MySQL ผ่าน Docker Compose โดยดาวน์โหลดไฟล์ docker-compose.yaml ผ่านเว็บไซต์ ดังรูปที่ [2.3] <https://airflow.apache.org/docs/apache-airflow/stable/start/docker.html>



รูปที่ 2.3 ดาวน์โหลดไฟล์ docker-compose.yaml

2.1.4 สร้าง Project Docker ดังรูปที่ [2.4] โดยมีโครงสร้างดังนี้

- Project

|-- airflow

 |--- Dockerfile

|-- dags

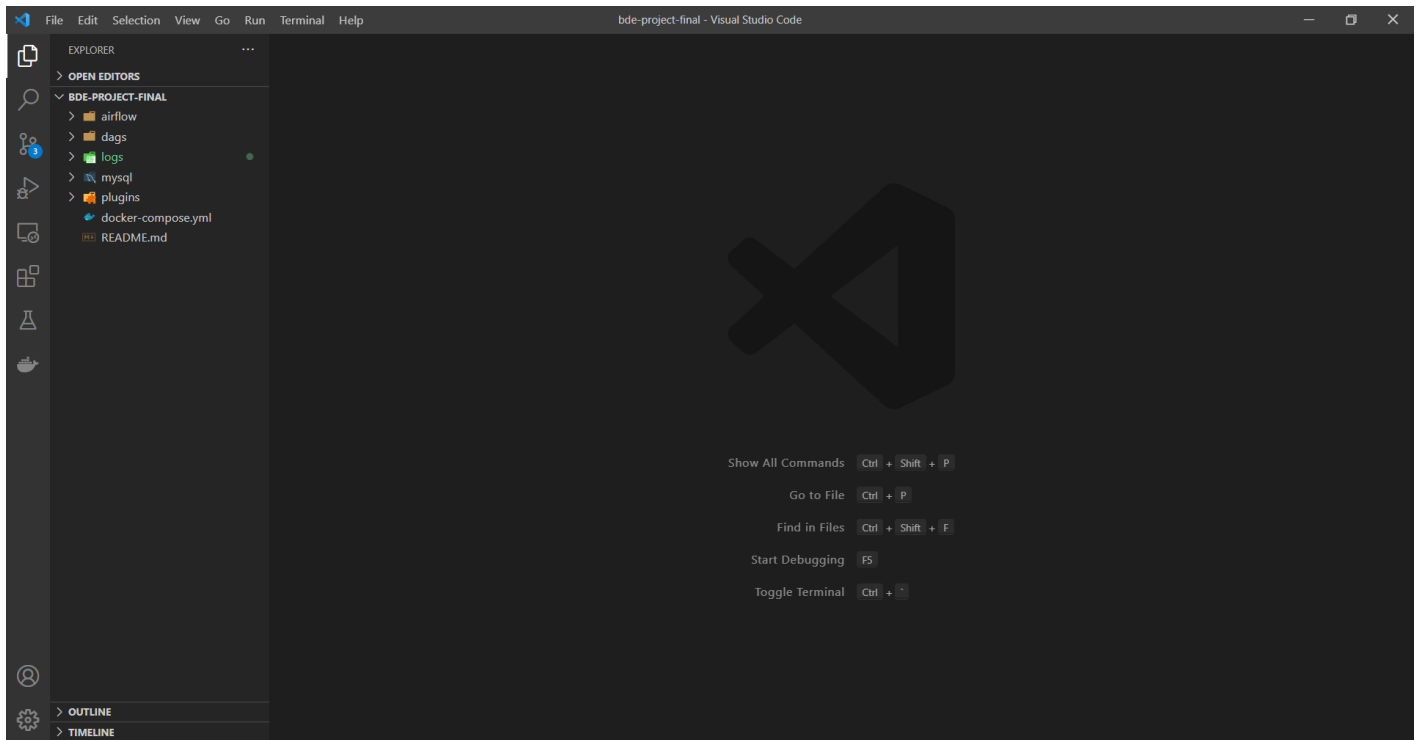
 |--- covid19_daily.py

|-- mysql

 |--- Dockerfile

|-- plugins

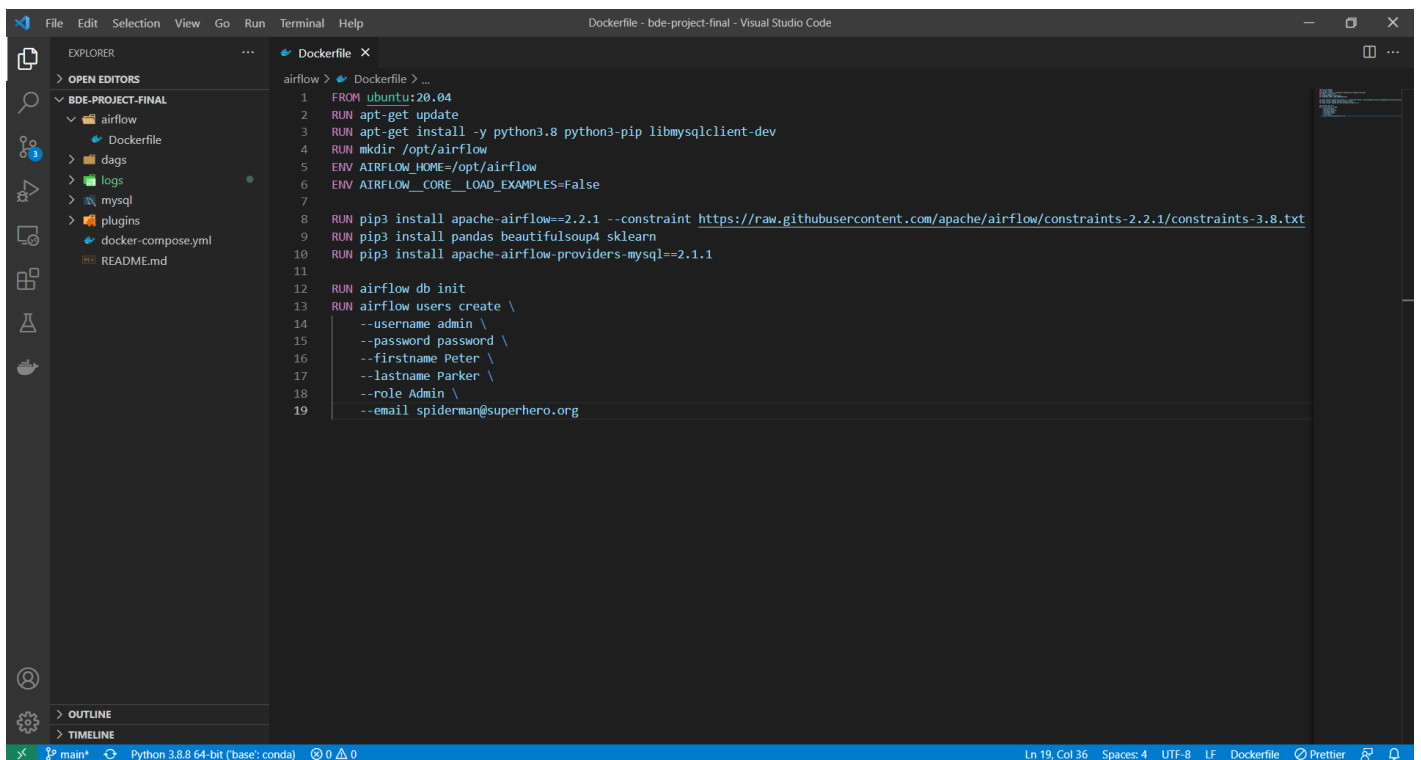
|-- docker-compose.yaml



ดังรูปที่ 2.4 โครงสร้างของโปรเจก

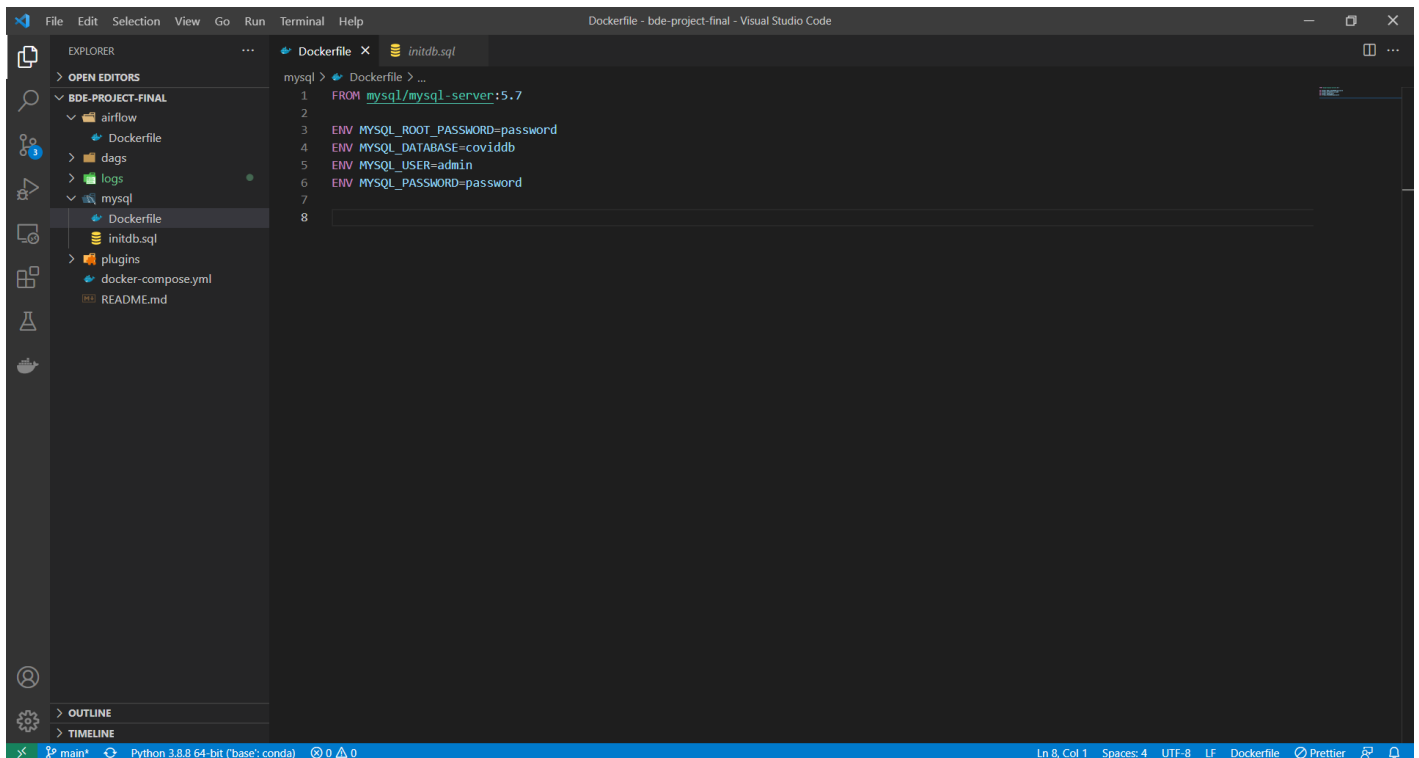
2.1.5 ไฟล์ Dockerfile ใน Folder airflow ระบุค่าพารามิเตอร์ ดังรูปที่ [2.5] โดยมีรายละเอียดดังนี้

- ติดตั้ง Ubuntu เวอร์ชัน 20.04
- ติดตั้ง Python เวอร์ชัน 3.8
- กำหนด username: admin และ password: password สำหรับ login เข้าสู่ระบบ



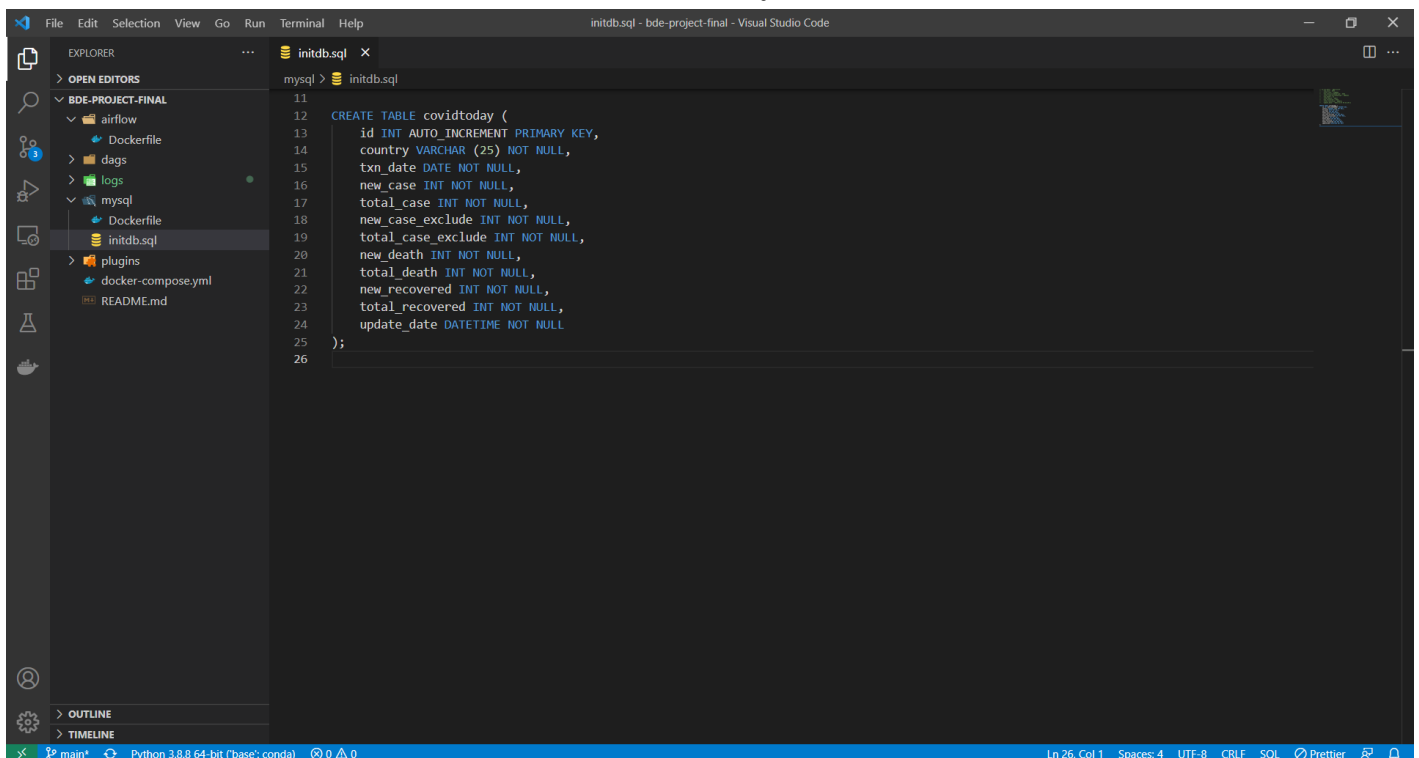
รูปที่ 2.5 ไฟล์ Dockerfile ใน Folder airflow

- 2.1.6 ไฟล์ Dockerfile ใน Folder mysql ระบุค่าพารามิเตอร์ ดังรูปที่ [2.6] โดยมีรายละเอียดดังนี้
- ติดตั้ง MySQL เวอร์ชัน 5.7
 - กำหนด username: admin และ password: password สำหรับ login เข้าสู่ระบบ



รูปที่ 2.6 ไฟล์ Dockerfile ใน Folder mysql

- 2.1.7 ไฟล์ initdb.sql ใน Folder mysql ระบุค่าพารามิเตอร์ ดังรูปที่ [2.7] โดยมีรายละเอียดดังนี้
- คำสั่งสร้าง table covidtoday สำหรับบันทึกข้อมูล



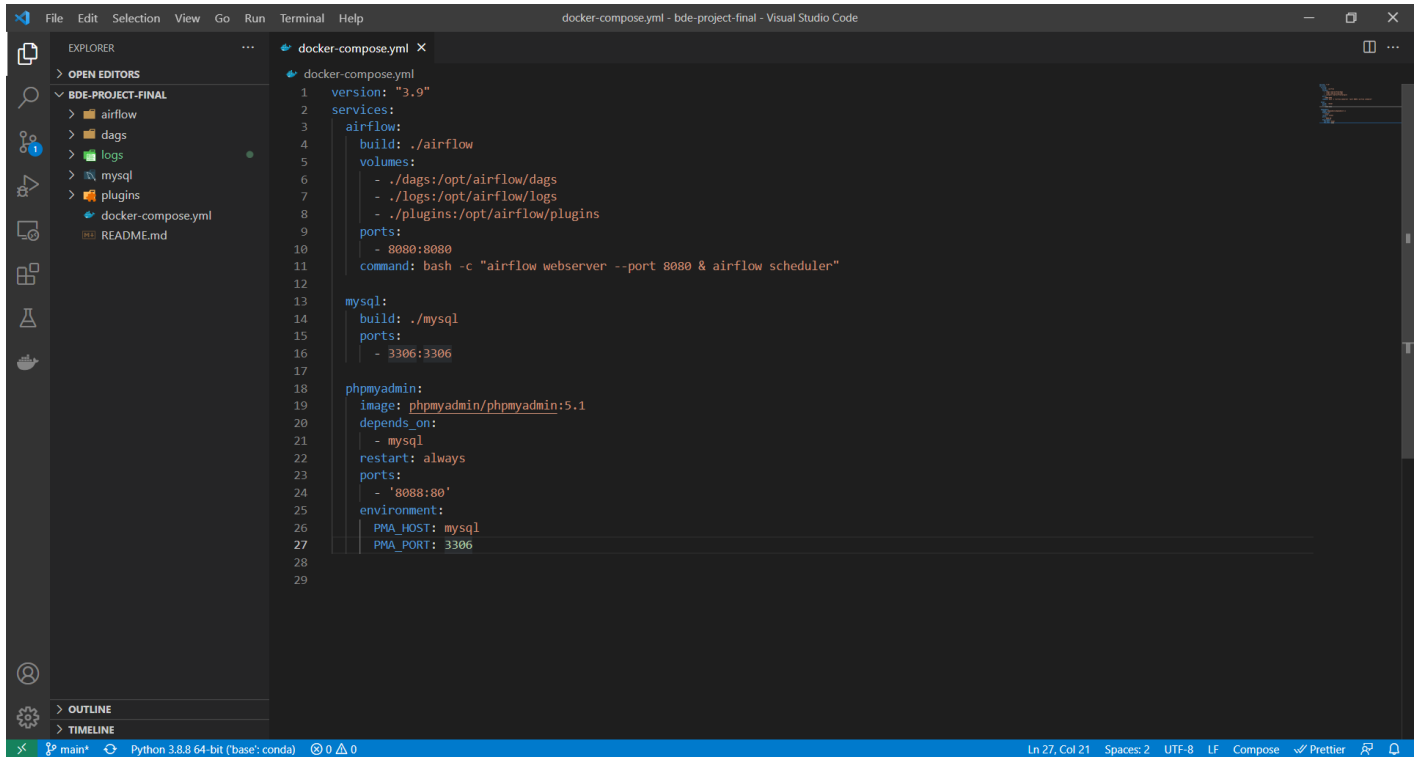
รูปที่ 2.7 ไฟล์ initdb.sql

2.1.8 ในไฟล์ docker-compose.yaml ผู้จัดทำระบุเฉพาะ service ที่ใช้งานเท่านั้น ดังรูปที่ [2.8] โดยมีรายละเอียดดังนี้

-airflow ระบุ port 8080

-mysql ระบุ port 3306

-phpmyadmin ระบุ port 8088



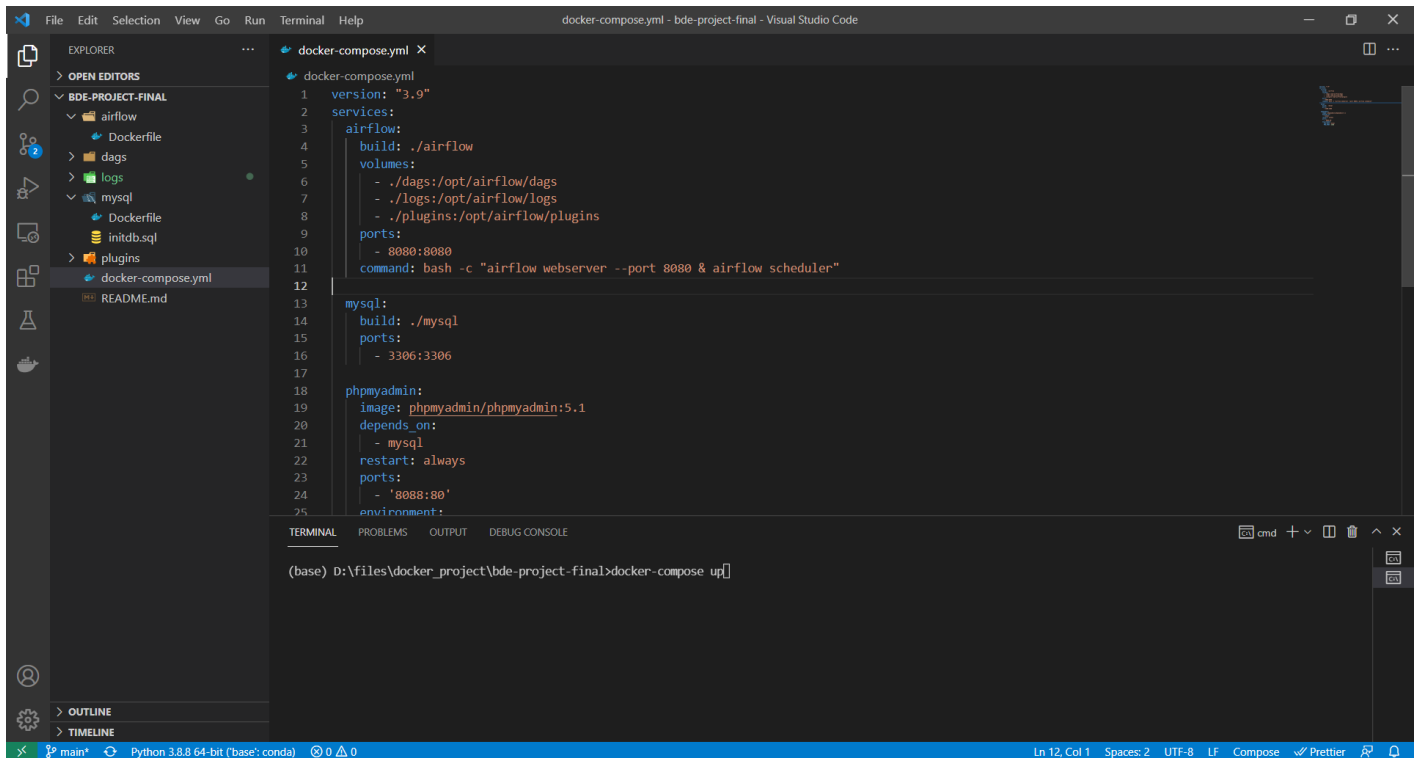
```
1 version: "3.9"
2 services:
3   airflow:
4     build: ./airflow
5     volumes:
6       - ./dags:/opt/airflow/dags
7       - ./logs:/opt/airflow/logs
8       - ./plugins:/opt/airflow/plugins
9     ports:
10      - 8080:8080
11     command: bash -c "airflow webserver --port 8080 & airflow scheduler"
12
13   mysql:
14     build: ./mysql
15     ports:
16      - 3306:3306
17
18   phpmyadmin:
19     image: phpmyadmin/phpmyadmin:5.1
20     depends_on:
21       - mysql
22     restart: always
23     ports:
24      - '8088:80'
25     environment:
26       PMA_HOST: mysql
27       PMA_PORT: 3306
28
29
```

รูปที่ 2.8 ไฟล์ docker-compose.yaml

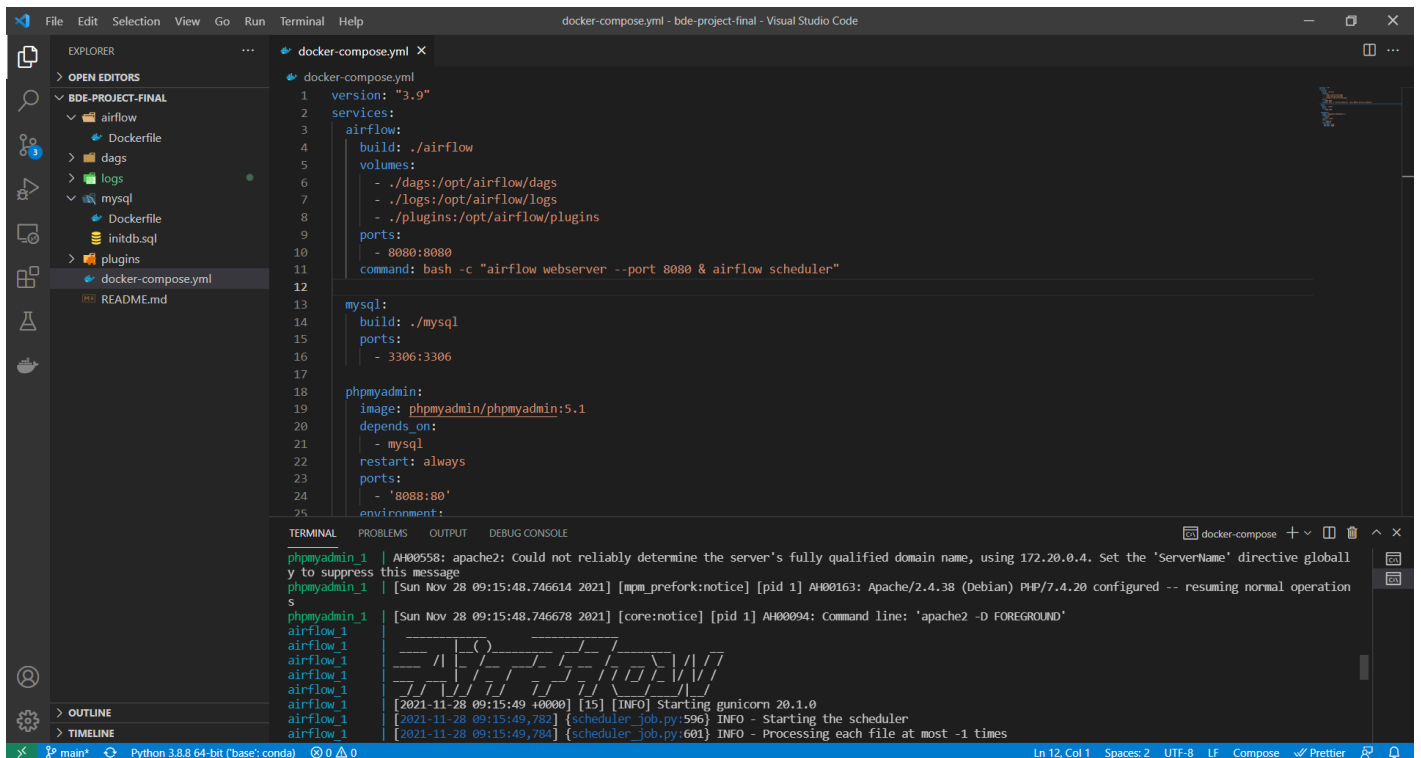
2.1.9 เปิด Terminal พิมพ์คำสั่ง ดังรูปที่ [2.9] ดังนี้

- docker-compose up airflow-init

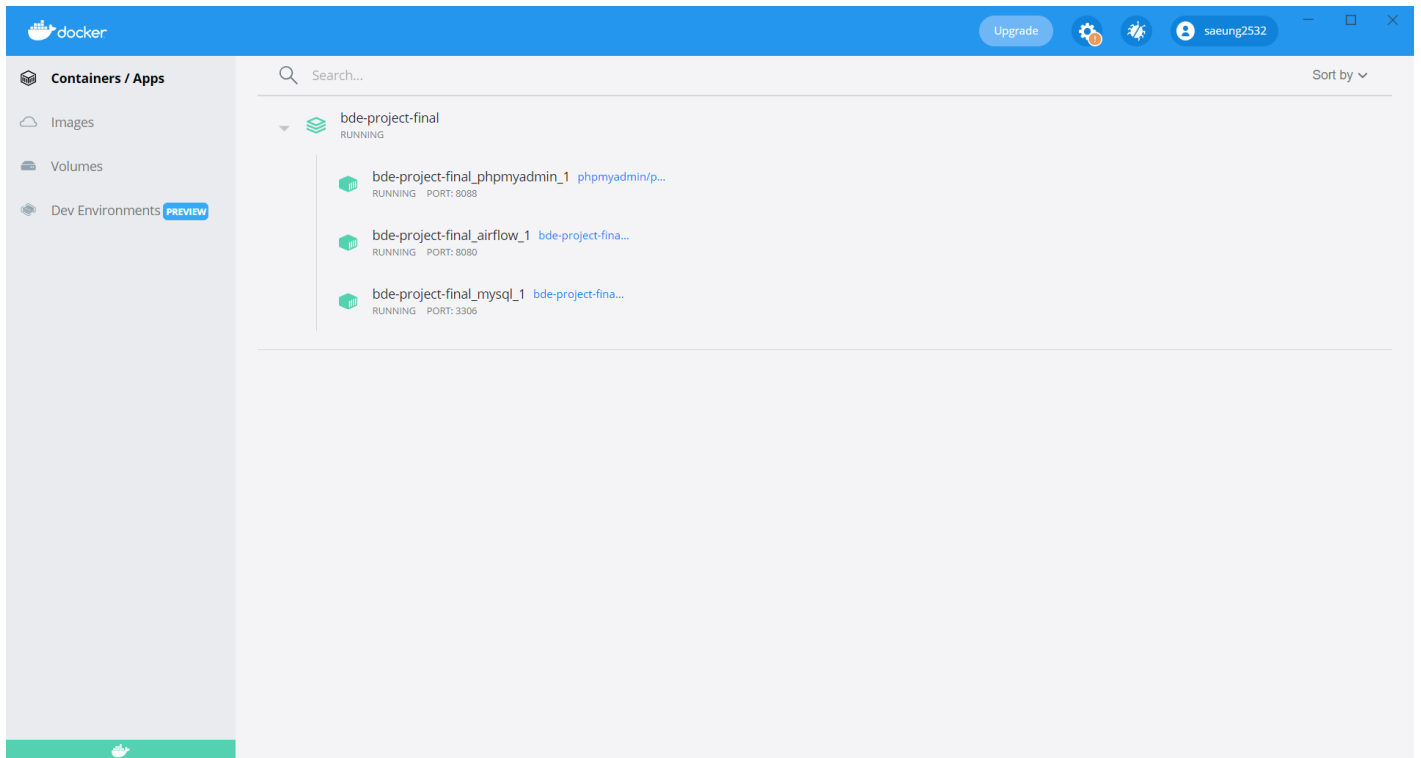
- docker-compose up #เพื่อรัน service ภายใน docker-compose.yaml



รูปที่ 2.9 พิมพ์คำสั่ง docker-compose up เพื่อรัน service ภายใน docker-compose.yaml



รูปที่ 2.10 รัน service ภายใน docker-compose.yaml เสร็จสิ้น

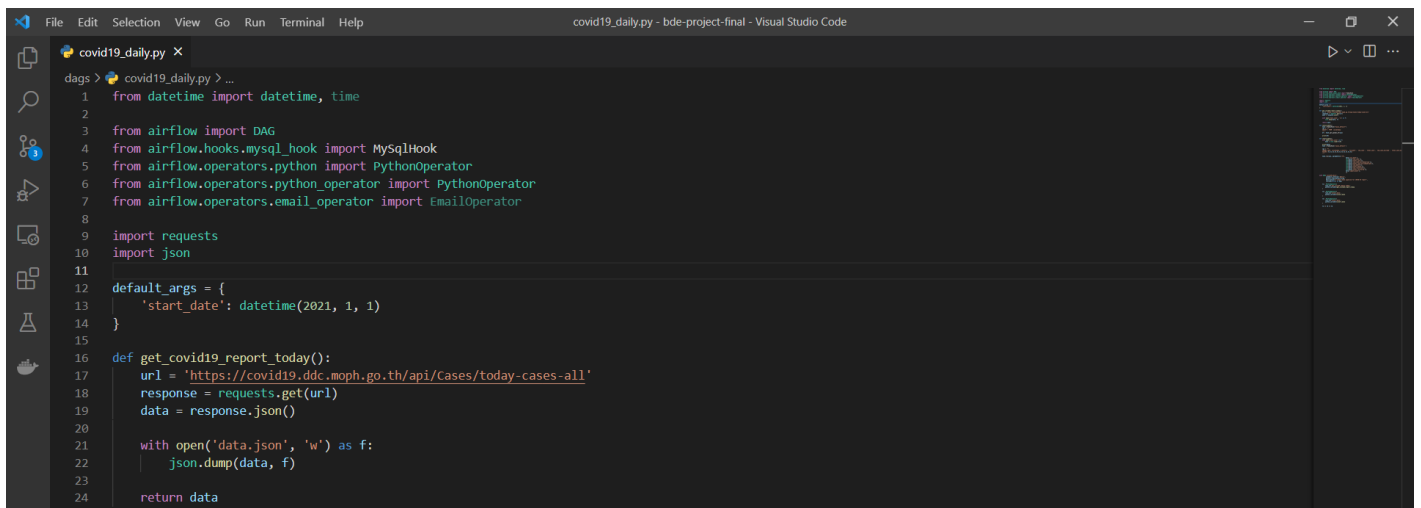


รูปที่ 2.11 service ที่รันภายใน docker ทั้งหมด

2.2 การเขียนโปรแกรมบันทึกข้อมูลผู้ติดเชื้อ COVID-19 ด้วยภาษา Python

2.2.1 ไฟล์ covid19_daily.py ใน Folder dags โดยมีรายละเอียดดังนี้

- ฟังก์ชันการดึงข้อมูล API จากเว็บไซต์ <https://covid19.ddc.moph.go.th/api/Cases/today-cases-all> และบันทึกเป็นโครงสร้างแบบ JSON Format ดังรูปที่ [2.12]
- ฟังก์ชันการดึงข้อมูลจากฐานข้อมูล MySQL ดังรูปที่ [2.13]
- ฟังก์ชันการบันทึกข้อมูลลงฐานข้อมูล MySQL ดังรูปที่ [2.13]

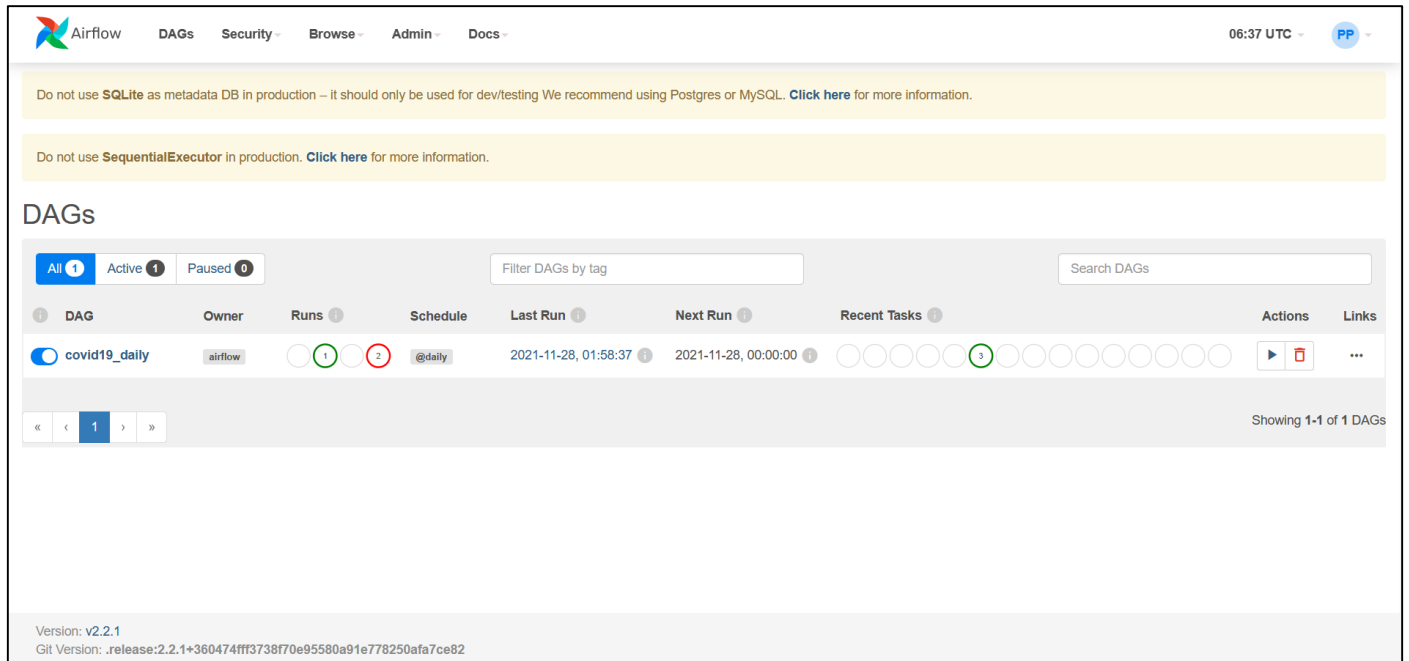


รูปที่ 2.12 ฟังก์ชันการดึงข้อมูล API

จากเว็บไซต์ <https://covid19.ddc.moph.go.th/api/Cases/today-cases-all>

2.3 การทดสอบระบบ

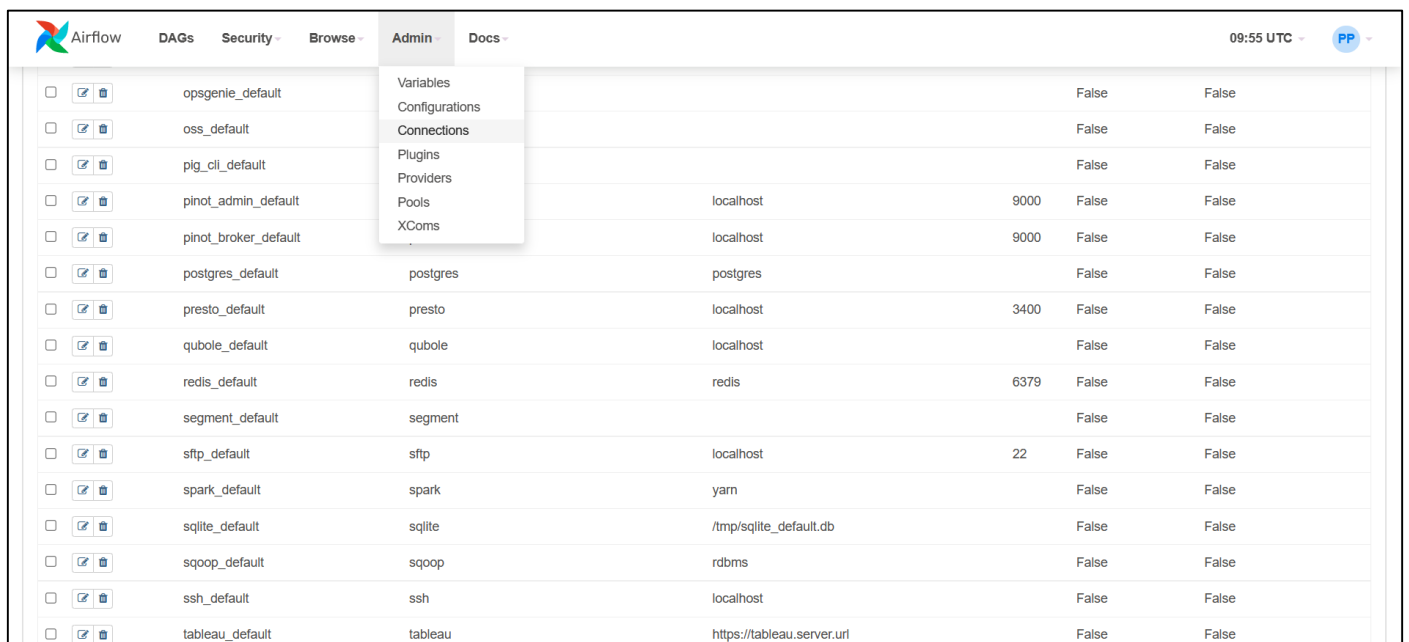
2.3.1 เข้า Apache Airflow ผ่านลิงก์ <http://localhost:8080/> และ login เข้าสู่ระบบ ดังรูปที่ [2.15]



รูปที่ 2.15 โปรแกรม Airflow

2.3.2 เพิ่ม Connections Database MySQL โดยทำตามขั้นตอนดังนี้

- คลิกที่ปุ่ม Admin → Connections ดังรูปที่ [2.16]
- เลือก Conn_id: mysql_default → Edit ดังรูปที่ [2.17]



รูปที่ 2.16 เพิ่ม Connections Database MySQL

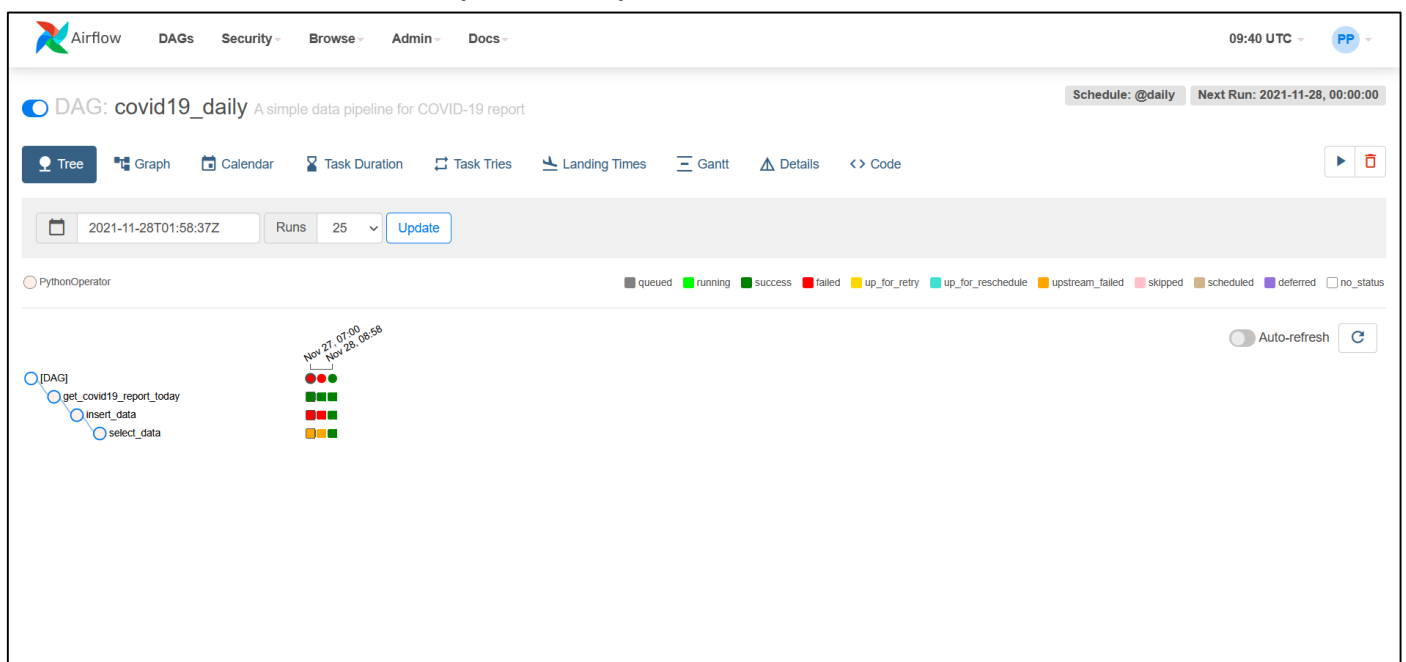
The screenshot shows the 'Edit Connection' form in the Airflow web interface. The form includes the following fields:

- Connection Id ***: mysql_default
- Connection Type ***: MySQL (with a dropdown arrow and a note: 'Connection Type missing? Make sure you've installed the corresponding Airflow Provider Package.')
- Description**: (empty text area)
- Host**: mysql
- Schema**: coviddb
- Login**: admin
- Password**: (masked with dots)
- Port**: (empty text field)

รูปที่ 2.17 ระบุ Connections Database MySQL

2.3.3 ในส่วนการทำงาน DAGs ดังรูปที่ [2.15] ประกอบไปด้วย DAG ต่างๆ ที่อยู่ภายใน Folder dags เมื่อคลิกที่ covid19_daily จะแสดงแทสก์งาน ดังรูปที่ [2.18] โดยมีรายละเอียดดังนี้

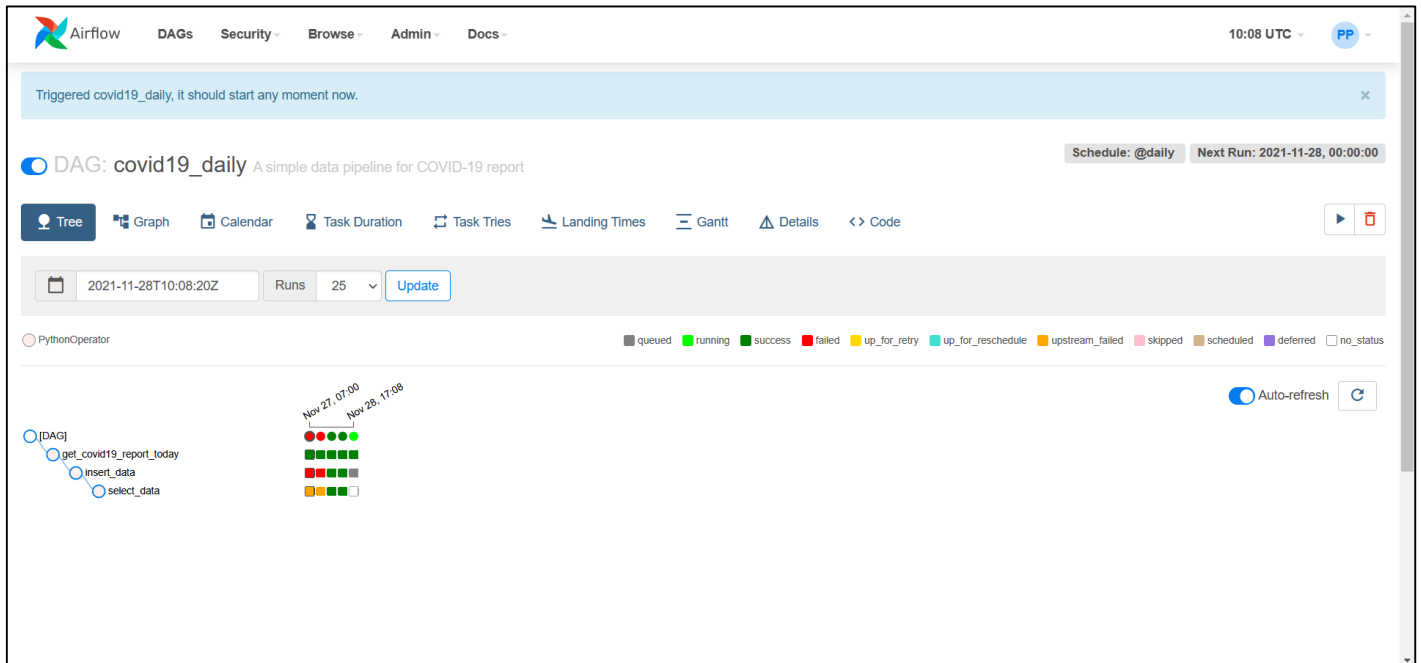
- get_covid19_report_today #ดึงข้อมูลผู้ป่วย COVID-19 จาก API
- insert_data #บันทึกข้อมูลลงฐานข้อมูล MySQL
- select_data #ดึงข้อมูลจากฐานข้อมูล MySQL



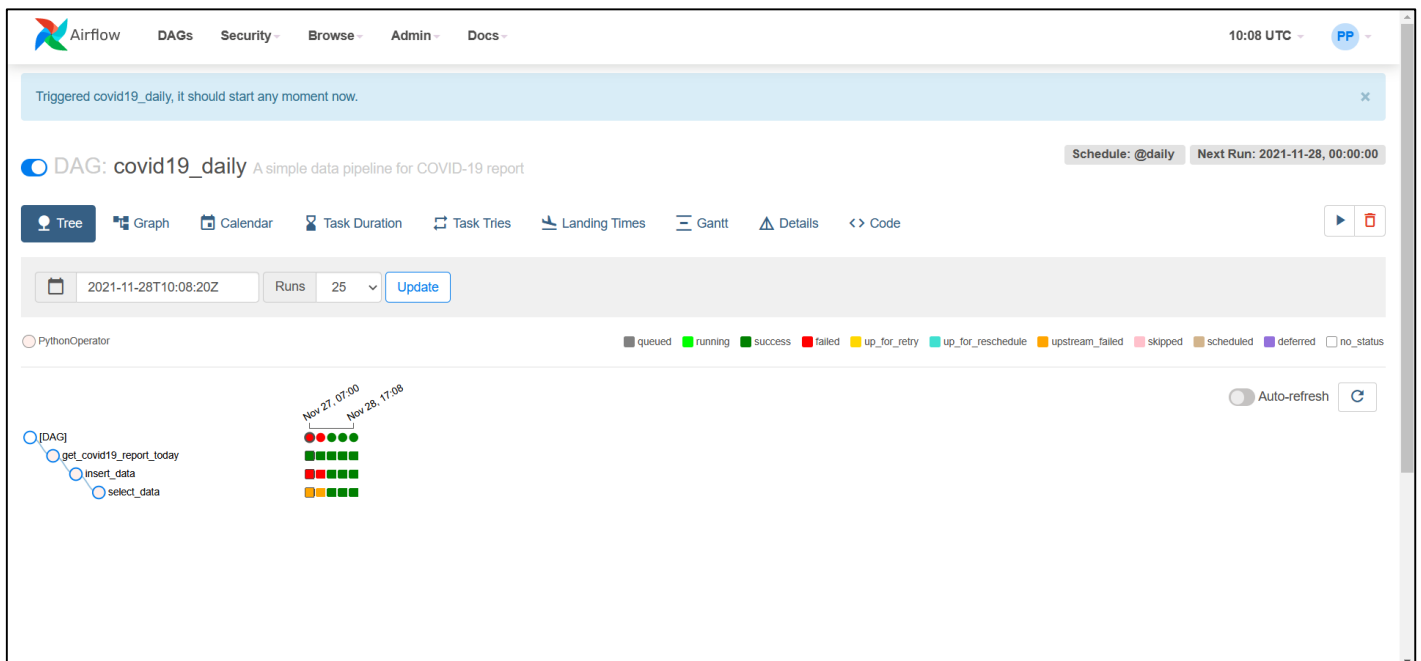
รูปที่ 2.18 แสดงแทสก์งาน ภายใน DAG covid19_daily

2.3.4 คลิกที่ปุ่ม Trigger DAG เพื่อรัน DAG covid19_daily ดังรูปที่ [2.19] โดยสามารถสังเกตสถานะการทำงานจากสี ได้ดังนี้

- สีเทา #รอคิว
- สีเขียวอ่อน #กำลังทำงาน
- สีเขียวเข้ม #ทำงานเสร็จสิ้น
- สีแดง #ทำงานล้มเหลว

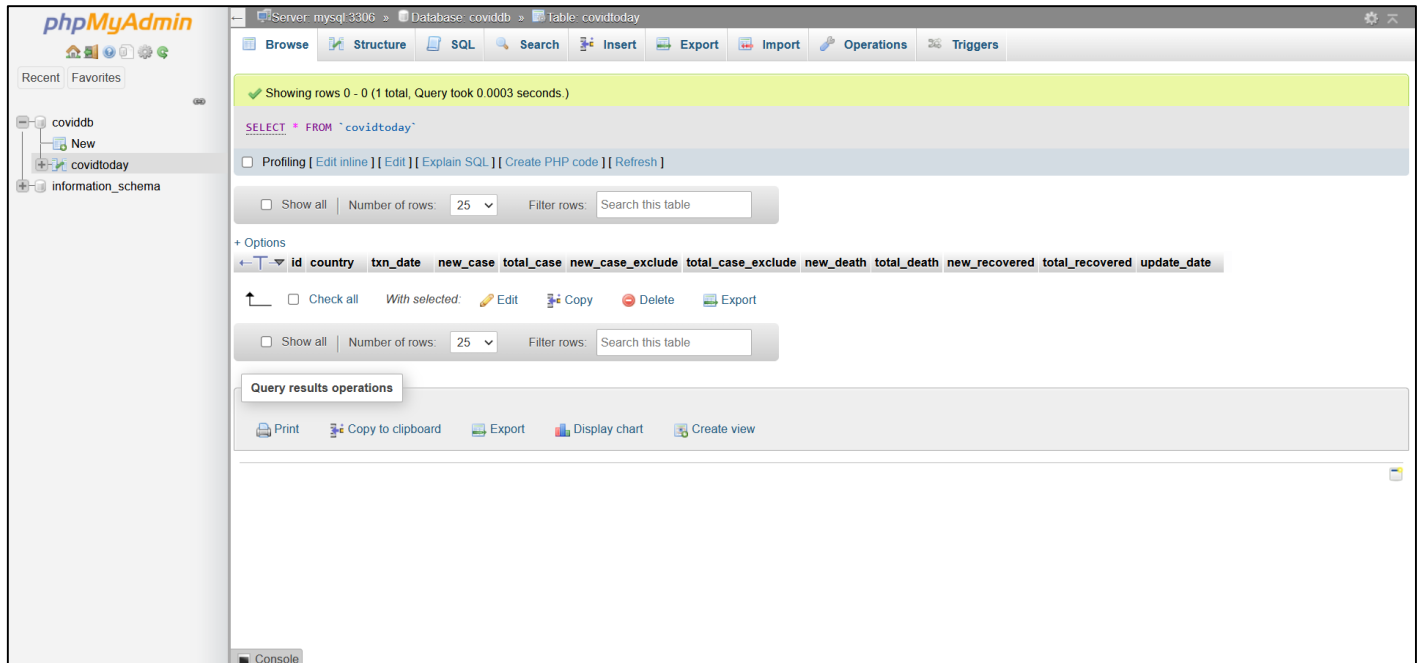


รูปที่ 2.19 คลิกปุ่ม Trigger DAG เพื่อรัน DAG covid19_daily



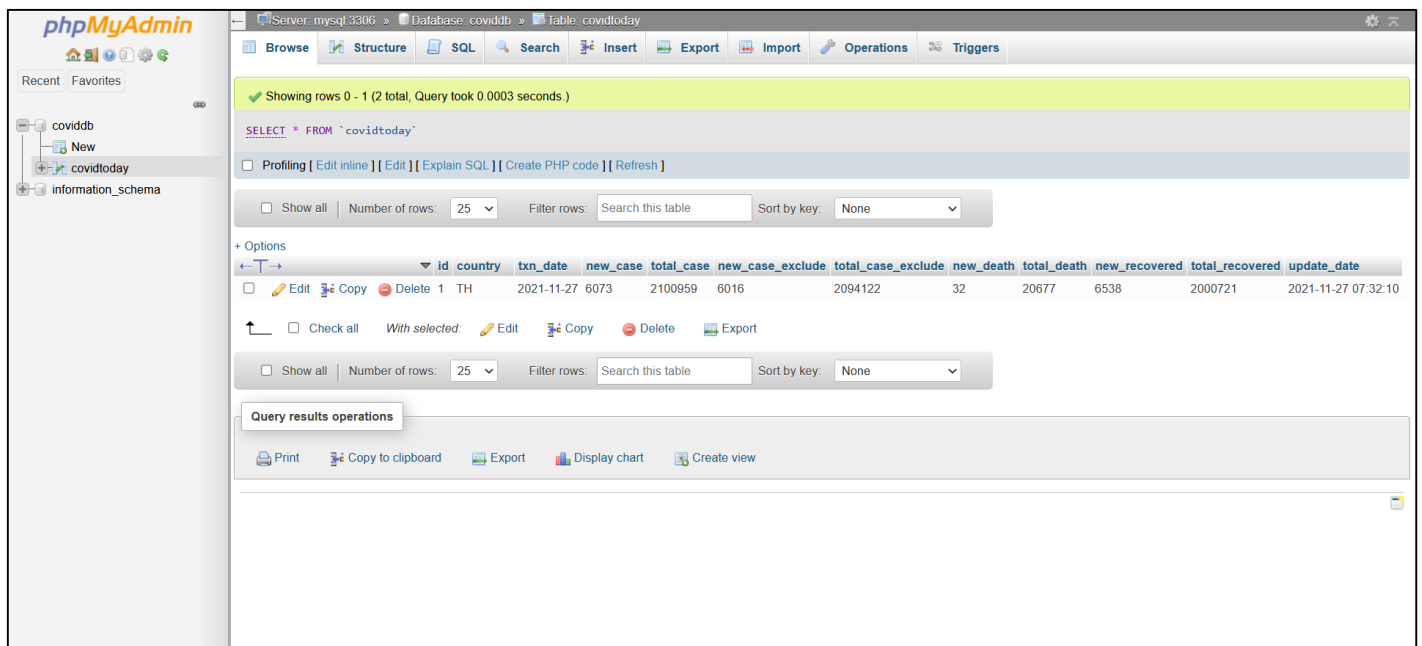
รูปที่ 2.20 DAG covid19_daily ทำงานเสร็จสิ้น (ขึ้นสีเขียวเข้มทั้งหมด)

2.3.5 เข้า phpMyAdmin ผ่านลิงก์ <http://localhost:8088/> และ login เข้าสู่ระบบ ดังรูปที่ [2.21] เพื่อตรวจสอบข้อมูลว่าบันทึกเข้าถูกต้องหรือไม่



รูปที่ 2.21 phpMyAdmin

2.3.6 เมื่อดูข้อมูลจาก table covidtoday จะมีข้อมูลที่ได้จากรัน DAG covid19_daily บันทึกอยู่ ดังรูปที่ [2.22]



3. สรุปผล

การจากออกแบบระบบการบันทึกข้อมูลสถานการณ์ผู้ติดเชื้อ COVID-19 ประจำวัน โดยใช้ Apache Airflow และ MySQL ด้วย Docker พบว่าระบบสามารถดึงข้อมูลจาก API และบันทึกลงฐานข้อมูลได้ โดยระบบทำงานได้อย่างรวดเร็ว มีประสิทธิภาพ สามารถนำไปประยุกต์ใช้ในงานอื่นๆได้ และสุดท้ายขอขอบคุณ ดร.ชยานนท์ ทรัพย์อาภา อาจารย์วิชา Big Data Engineer ที่สอนหลักทฤษฎีและปฏิบัติ ในการทำงานด้านวิศวกรรมข้อมูล

บรรณานุกรม

1. สถานการณ์ผู้ติดเชื้อ COVID-19 อัปเดตรายวัน รูปแบบ API(Json/CSV Data Format) :
<https://covid19.ddc.moph.go.th/>
2. Data Engineer Lab : <https://github.com/seedgit/data-engineer-lab>
3. Docker Install : <https://docs.docker.com/desktop/windows/install/>
4. Apache Airflow Install : <https://airflow.apache.org/docs/apache-airflow/stable/start/docker.html>
5. MySQL Install : https://hub.docker.com/_/mysql