

# Project: Creating and Automating a Set of Data Pipelines with Airflow

สิ่งที่คาดหวังในโปรเจกต์นี้

1. นำโค้ดจาก [Project: Building a Data Modeling with Postgres \(SQL\)](#) มาสร้าง data pipeline โดยใช้ Airflow
2. มีการเขียน documentation อธิบายสิ่งที่ตัวเองทำลงไป รวมไปถึงการออกแบบ data model
3. มี instruction ในการรันโค้ดของตัวเอง

```
gitpod /workspace/swu-ds525/05-creating-and-scheduling-data-pipelines (main) $  
docker-compose up
```

วิธีทำ Over All

- ① import DAG, time-zone
- ② สร้าง Data pipeline ชื่อ "MyDag"

③ save file python

④ กด กดเน้นที่ Airflow รอสักพักจะขึ้นที่ Airflow

$t_1 \gg t_2$

⑤ bash operator, python operator

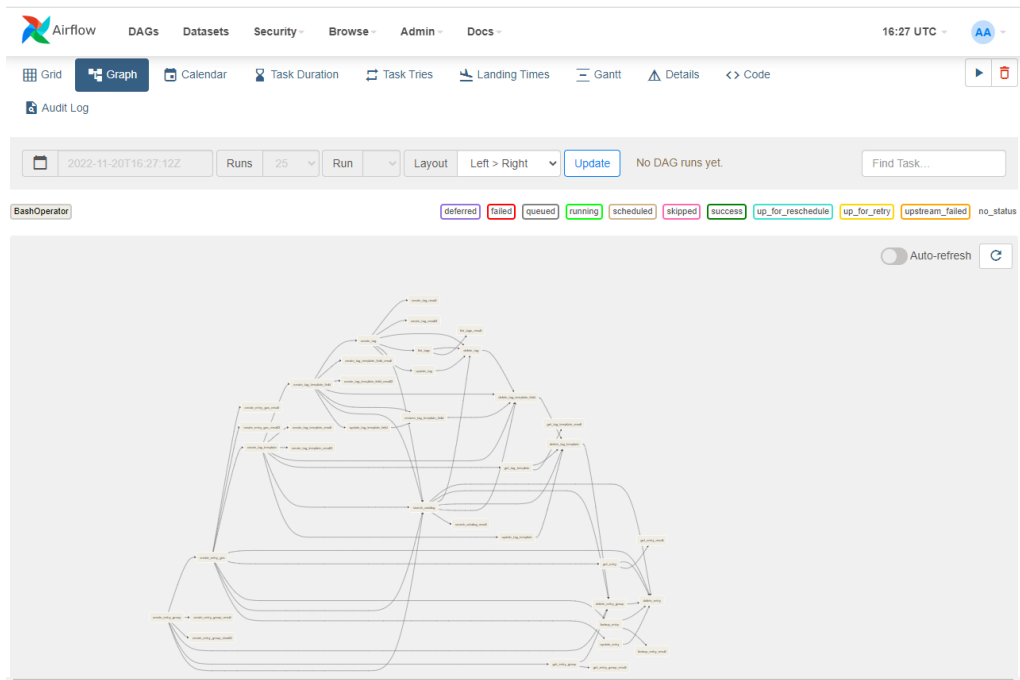
⑥ grid view

ค้นหา ย่อทาส์  
trigger 1 ที่ได้ 1 day run

clear task

$t_1 \rightarrow details \rightarrow clear \rightarrow confirm$   
จะ run ใหม่อีกรอบ

cronitor  $\rightarrow$  expression เพื่อ set ใน run croniter - เพื่อใช้



05-creating-and-scheduling-data-pipelines > dags > my\_dag\_revise.py

```

1  #import DAG เป็น pipeline
2  from airflow import DAG
3  #import Timezone
4  from airflow.utils import timezone
5  #step 2 ใช้ EmptyOperator เลย import EmptyOperator เข้ามา
6  from airflow.operators.empty import EmptyOperator
7
8
9  #context manager เป็นการประกาศหัว
10 # "my_dag" ชื่อเดียวกับชื่อ file
11 #start date 2022, 10, 8
12 # schedule = None ยังไม่schedule
13 # step10
14 # schedule
15 # schedule เที่ยงคืน คือ 0
16 #v1 schedule = None ยังไม่set schedule
17 with DAG(
18     "my_dag",
19     start_date = timezone.datetime(2022, 10, 8),
20     schedule = None,
21 ):
22     t1 = EmptyOperator( task_id = "t1")
23     t2 = EmptyOperator( task_id = "t2")

```

Airflow DAGs Datasets Security Browse Admin Docs 16:49 UTC AA

example2 example3

latest\_only\_with\_trigger example3 airflow 4:00:00 2022-11-20, 12:48:07

my\_dag workshop airflow 2022-11-20, 16:24:00 2022-11-20, 00:00:00

my\_dag2 workshop airflow 2022-11-20, 16:00:00 2022-11-20, 16:30:00

my\_dag\_revise airflow None

Airflow DAGs Datasets Security Browse Admin Docs 16:55 UTC AA

### DAGs

All 47 Active 3 Paused 44 Filter DAGs by tag Search DAGs Auto-refresh

DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks	Actions
my_dag workshop	airflow	325 3	0 0 * * *	2022-11-20, 16:24:00	2022-11-20, 00:00:00	4	▶ 🗑
my_dag2 workshop	airflow	7	*30 * * * *	2022-11-20, 16:00:00	2022-11-20, 16:30:00	5	▶ 🗑
my_dag_revise workshop	airflow		None				▶ 🗑

Trigger DAG  
Trigger DAG w/ config

Airflow DAGs Datasets Security Browse Admin Docs 16:56 UTC AA

Triggered my\_dag\_revise, it should start any moment now.

### DAGs

All 47 Active 3 Paused 44 Filter DAGs by tag Search DAGs Auto-refresh

DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks	Actions
my_dag workshop	airflow	325 3	0 0 * * *	2022-11-20, 16:24:00	2022-11-20, 00:00:00	4	▶ 🗑
my_dag2 workshop	airflow	7	*30 * * * *	2022-11-20, 16:00:00	2022-11-20, 16:30:00	5	▶ 🗑
my_dag_revise workshop	airflow	1	None	2022-11-20, 16:55:59		3	▶ 🗑

Showing 1-3 of 3 DAGs

Airflow DAGs Datasets Security Browse Admin Docs 16:56 UTC AA

#### DAG: my\_dag\_revise

success Schedule: None Next Run: None

Grid Graph Calendar Task Duration Task Times Landing Times Gantt Details <> Code

Audit Log

2022-11-20T16:56:00Z Runs 25 Run manual\_\_2022-11-20T16:55:59.972252+00:00 Layout Left > Right Find Task...

Update

EmptyOperator deferred failed queued running scheduled skipped success up\_for\_reschedule up\_for\_retry upstream\_failed no\_status

Auto-refresh

t1

t2

05-creating-and-scheduling-data-pipelines > dags > my\_dag\_revise.py

```
1  #import DAG เป็น pipeline
2  from airflow import DAG
3  #import Timezone
4  from airflow.utils import timezone
5  #step 2 ใช้ EmptyOperator เลย import EmptyOperator เข้ามา
6  from airflow.operators.empty import EmptyOperator
7
8
9  #context manager เป็นการประกาศหัว
10 # "my_dag" ชื่อเดียวกับชื่อ file
11 #start date 2022, 10, 8
12 # schedule = None ยังไม่schedule
13 # step10
14 # schedule
15 # schedule เพียงคืน คือ 0
16 # tags = ['workshop'] จะทำให้เป็นชัดเจน
17 #v1 schedule = None ยังไม่set schedule
18 with DAG(
19     "my_dag_revise",
20     start_date = timezone.datetime(2022, 10, 8),
21     schedule = None,
22     tags = ["workshop"],
23 ):
24     t1 = EmptyOperator( task_id = "t1")
25     t2 = EmptyOperator( task_id = "t2")
26
27 #t1 รันก่อน t2
28 t1 >> t2
29
```

Airflow DAGs Datasets Security Browse Admin Docs 16:59 UTC AA

Triggered my\_dag\_revise, it should start any moment now.

DAG: my\_dag\_revise **queued** Schedule: None Next Run: None

Grid Graph Calendar Task Duration Task Tries Landing Times Gantt Details <> Code

Audit Log

2022-11-20T16:59:28Z Runs 25 Run manual\_\_2022-11-20T16:59:27.841727+00:00 Layout Left > Right Find Task...

Update

EmptyOperator deferred failed queued **running** scheduled skipped success up\_for\_reschedule up\_for\_retry upstream\_failed no\_status

Auto-refresh

t1 → t2

05-creating-and-scheduling-data-pipelines > dags > my\_dag\_revise.py

```
25 # ):
26 #     t1 = EmptyOperator( task_id = "t1")
27 #     t2 = EmptyOperator( task_id = "t2")
28
29 # #t1 รันก่อน t2
30 #     t1 >> t2
31
32 #v2
33 with DAG(
34     "my_dag_revise",
35     start_date = timezone.datetime(2022, 10, 8),
36     schedule = None,
37     tags=["workshop"],
38 ):
39     t1 = EmptyOperator( task_id = "t1")
40
41     echo_hello = BashOperator(
42         task_id = "echo_hello",
43         bash_command= "echo 'hello'",
44     )
45
46     t2 = EmptyOperator( task_id = "t2")
47
48 #t1 รันก่อน t2
49     t1 >> t2
```

DAG: my\_dag\_revise success Schedule: None Next Run: None

Grid Graph Calendar Task Duration Task Tries Landing Times Gantt Details <> Code

Audit Log

2022-11-20T16:59:28Z Runs 25 Run manual\_\_2022-11-20T16:59:27.841727+00:00 Layout Left > Right Find Task...

Update

BashOperator EmptyOperator

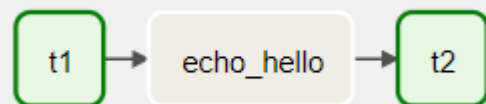
deferred failed queued running scheduled skipped success up\_for\_reschedule up\_for\_retry upstream\_failed no\_status

Auto-refresh

echo\_hello

t1 → t2

```
#t1 รันก่อน t2
t1 >> echo_hello >> t2
```



DAG: my\_dag\_revise

Schedule: NoneNext Run: None

GridGraphCalendarTask DurationTask TriesLanding TimesGanttDetailsCodeAudit Log

11/20/2022 05:12:27 PM25All Run TypesAll Run StatesClear Filters

Auto-refresh

Duration

00:00:03

00:00:01

00:00:00

t1

echo\_hello

t2

DAG

my\_dag\_revise

DAG Details

DAG Runs Summary

Total Runs Displayed	3
Total success	3
First Run Start	2022-11-20, 16:56:00 UTC
Last Run Start	2022-11-20, 17:12:19 UTC
Max Run Duration	00:00:03
Mean Run Duration	00:00:01
Min Run Duration	00:00:00

DAG Summary

Total Tasks	3
Empty Operators	2
BashOperator	1

Triggered my\_dag\_revise, it should start any moment now.

DAG: my\_dag\_revise

Schedule: NoneNext Run: None

GridGraphCalendarTask DurationTask TriesLanding TimesGanttDetailsCodeAudit Log

11/20/2022 05:14:10 PM25All Run TypesAll Run StatesClear Filters

Auto-refresh

Duration

00:00:03

00:00:01

00:00:00

t1

echo\_hello

t2

DAG

my\_dag\_revise

Run

2022-11-20, 17:12:18 UTC

Task

echo\_hello

Task Instance DetailsRendered TemplateLogXComList Instances, all runsFilter Upstream

DetailsLogs

Task Actions

Ignore All DepsIgnore Task StateIgnore Task DepsRun

PastFutureUpstreamDownstreamRecursiveFailedClear

PastFutureUpstreamDownstreamMark Failed

PastFutureUpstreamDownstreamMark Success

Status

success

Task ID

echo\_hello

Run ID

manual\_\_2022-11-20T17:12:18.932212+00:00

Operator

BashOperator

Duration

00:00:00

Started

2022-11-20, 17:12:21 UTC

Ended

2022-11-20, 17:12:22 UTC

Airflow DAGs Datasets Security Browse Admin Docs 17:14 UTC

Triggered my\_dag\_revise, it should start any moment now.

DAG: my\_dag\_revise Scheduler: None Next Run: None

Grid Graph Calendar Task Duration Task Tries Landing Times Gantt Details <> Code Audit Log

11/20/2022 05:14:10 PM 25 All Run Types All Run States Clear Filters

Auto-refresh

Duration Nov 20, 16:58

my\_dag\_revise Run 2022-11-20, 17:12:18 UTC Task echo\_hello

Task Instance Details Rendered Template Log XCom List Instances, all runs Filter Upstream

Details Logs

(by attempts)

All Levels All File Sources Wrap Full Logs Download See More

```

*** Reading Local File: /opt/airflow/logs/dag_id=my_dag_revise/run_id=manual_2022-11-20T17:12:18.93222+00:00/task_id=
[2022-11-20, 17:12:21 UTC] (taskinstance.py:1163) INFO - Dependencies all met for <taskinstance: my_dag_revise,echo_hello>
[2022-11-20, 17:12:21 UTC] (taskinstance.py:1163) INFO - Dependencies all met for <taskinstance: my_dag_revise,echo_hello>
[2022-11-20, 17:12:21 UTC] (taskinstance.py:1162) INFO -
[2022-11-20, 17:12:21 UTC] (taskinstance.py:1163) INFO - Starting attempt 1 of 1
[2022-11-20, 17:12:21 UTC] (taskinstance.py:1164) INFO -
[2022-11-20, 17:12:21 UTC] (taskinstance.py:1163) INFO - Executing <task(BashOperator): echo_hello on 2022-11-20 17:12:
[2022-11-20, 17:12:21 UTC] (standard_task_runner.py:34) INFO - Started process 24279 to run task
[2022-11-20, 17:12:21 UTC] (standard_task_runner.py:82) INFO - Running: [***, 'task', 'run', 'my_dag_revise', 'echo_h
[2022-11-20, 17:12:21 UTC] (standard_task_runner.py:83) INFO - Job 687: Subtask echo_hello
[2022-11-20, 17:12:21 UTC] (dagbag.py:525) INFO - Killing up the dagbag from /opt/***/dagrev_dag_revise.py
[2022-11-20, 17:12:21 UTC] (taskinstance.py:283) WARNING - Dependency <task(BashOperator): create_entry_group, delete_entry
[2022-11-20, 17:12:21 UTC] (taskinstance.py:283) WARNING - Dependency <task(BashOperator): delete_entry_group, create_entry
[2022-11-20, 17:12:21 UTC] (taskinstance.py:283) WARNING - Dependency <task(BashOperator): create_entry_group, delete_entry
[2022-11-20, 17:12:21 UTC] (taskinstance.py:283) WARNING - Dependency <task(BashOperator): delete_entry, create_entry_group
[2022-11-20, 17:12:21 UTC] (taskinstance.py:283) WARNING - Dependency <task(BashOperator): create_tag, delete_tag already
[2022-11-20, 17:12:21 UTC] (taskinstance.py:283) WARNING - Dependency <task(BashOperator): delete_tag, create_tag already
[2022-11-20, 17:12:21 UTC] (taskinstance.py:283) WARNING - Dependency <task(BashOperator): get_ip, prepare_email air

```

```

#v3
with DAG(
    "my_dag_revise",
    start_date = timezone.datetime(2022, 10, 8),
    schedule = None,
    tags=["workshop"],
):
    t1 = EmptyOperator( task_id = "t1")

    echo_hello = BashOperator(
        task_id = "echo_hello",
        bash_command= "echo 'hello'",
    )
    def _print_hey():
        print("Hey!")

    print_hey = PythonOperator(
        task_id = "print_hey",
        python_callable = _print_hey,
    )

    t2 = EmptyOperator( task_id = "t2")

    #t1 ขึ้นก่อน t2
    t1 >> echo_hello >> print_hey >> t2

```

Airflow DAGs Datasets Security Browse Admin Docs 17:24 UTC

DAG: my\_dag\_revise success Scheduler: None Next Run: None

Grid Graph Calendar Task Duration Task Tries Landing Times Gantt Details <> Code Audit Log

2022-11-20T17:17:00Z Runs 25 Run manual\_2022-11-20T17:16:59.347340+00:00 Layout Left ~ Right Update Find Task...

BashOperator EmptyOperator PythonOperator

Auto-refresh

11 echo\_hello print\_hey 12

← → ↻ cronstab.guru

poll\_form.php Canny edge detect... Your Repositories Simple guide on ho... A Neural Network P... Learner Lab Image Feature Extra... Canny edge detect... How to tune hyper...>

Cronitor

Cron Job Monitoring

crontab guru

The quick and simple editor for cron schedule expressions by Cronitor

"At 04:05."

next at 2022-11-21 04:05:00 random

5 4 \* \* \*

minute	hour	day (month)	month	day (week)
		*	any value	
		,	value list separator	
		-	range of values	
		/	step values	
		@yearly	(non-standard)	
		@annually	(non-standard)	
		@monthly	(non-standard)	
		@weekly	(non-standard)	
		@daily	(non-standard)	
		@hourly	(non-standard)	
		@reboot	(non-standard)	

We created Cronitor because cron itself can't alert you if your jobs fail or never start. Cronitor is easy to integrate and provides you with instant alerts when things go wrong.





```

with DAG(
    "my_dag_revise",
    start_date = timezone.datetime(2022, 10, 8),
    schedule = " * * * * * ",
    tags=["workshop"],
):
    t1 = EmptyOperator( task_id = "t1")

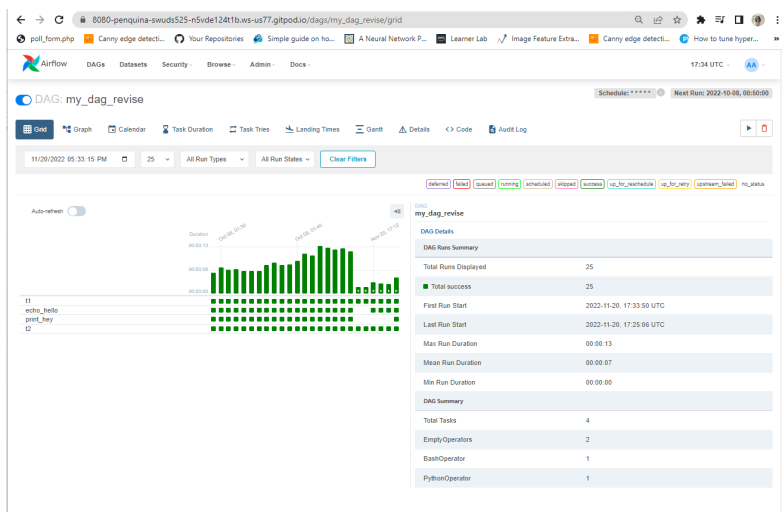
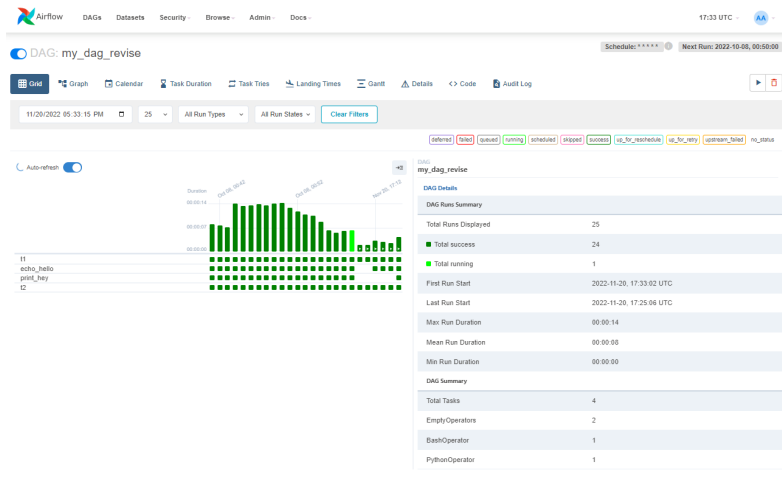
    echo_hello = BashOperator(
        task_id = "echo_hello",
        bash_command= "echo 'hello'",
    )
    def _print_hey():
        print("Hey!")

    print_hey = PythonOperator(
        task_id = "print_hey",
        python_callable = _print_hey,
    )

    t2 = EmptyOperator( task_id = "t2")

#t1 รันก่อน t2
t1 >> echo_hello >> print_hey >> t2

```



```
#t1 รันก่อน t2
#t1 >> echo_hello >> print_hey >> t2
t1 >> [ echo_hello, print_hey ] >> t2
```

