DATA ENGINEER BATCH 4
Mentee: Yovina Silvia
Mentor: Raja Fathurrahman

TASK I - Airflow

1. Create DAG that run in every 5 hours.

   Buat file tugas pada server yang ditentukan

```
MINGW64:/c/Users/USER/Alta,   raja_rahmanakmaludin@insta
USER@Luna MINGW64 ~
$ wsl
root@Luna:/mnt/c/Users/USER# ssh raja_rahmanakmaludin@34.101.224.54
raja_rahmanakmaludin@34.101.224.54's password:
Linux instance-20240714-035051 6.1.0-23-cloud-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.99-1 (2024-07-15) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jul 31 03:52:41 2024 from 202.43.94.42
raja_rahmanakmaludin@instance-20240714-035051:~$
raja_rahmanakmaludin@instance-20240714-035051:~$ ls
2022-01-01-1.json.gz  airflow-data  dbt-demo  extract-load-demo  ingestion-data  streaming-platform
raja_rahmanakmaludin@instance-20240714-035051:~$
raja_rahmanakmaludin@instance-20240714-035051:~$ cd airflow-data/docker/dags
raja_rahmanakmaludin@instance-20240714-035051:~/airflow-data/docker/dags$
raja_rahmanakmaludin@instance-20240714-035051:~/airflow-data/docker/dags$ sudo nano yovinasilvia-airflow-task1.py
[sudo] password for raja_rahmanakmaludin:
raja_rahmanakmaludin@instance-20240714-035051:~/airflow-data/docker/dags$
```

   codingannya:

   - Membuat DAG dengan ID 'yovina-airflow-task1', deskripsi 'Airflow Task 1', dan jadwal interval '0 */5 * * *', yang berarti DAG akan berjalan setiap 5 jam.
   - start_date diatur pada datetime(2023, 10, 18) untuk menentukan kapan DAG mulai berjalan.
   - catchup=False berarti Airflow tidak akan menjalankan DAG untuk waktu yang telah berlalu sebelum start_date.
   - Membuat tugas awal yang kosong dengan task_id='start'.

```
from datetime import datetime
from airflow import DAG
```

```
from airflow.operators.empty import EmptyOperator
from airflow.operators.python_operator import PythonOperator

# 1. Create DAG that run in every 5 hours.
dag = DAG(
    'yovina-airflow-task1',
    description='Airflow Task 1',
    schedule_interval='0 */5 * * *',
    start_date=datetime(2023, 10, 18),
    catchup=False
)
start = EmptyOperator(
    task_id='start',
    dag=dag,
)
```

2. Suppose we define a new task that push a variable to xcom.

   Fungsi push_variable_to_xcom digunakan untuk mendorong beberapa variabel ke XCom. Variabel ini disimpan dengan kunci 'job_role', 'job_role_1', 'job_role_2', dan 'job_role_3'.

```
def push_variable_to_xcom(ti=None):
    ti.xcom_push(key='job_role', value='Backend Engineer')
    ti.xcom_push(key='job_role_1', value='Data Engineer')
    ti.xcom_push(key='job_role_2', value='Frontend Engineer')
    ti.xcom_push(key='job_role_3', value='Quality Assurance')
```

3. How to pull multiple values at once?

   - Fungsi pull_multiple_value_once digunakan untuk menarik beberapa variabel dari XCom menggunakan ti.xcom_pull. Variabel yang ditarik kemudian dicetak.
   - Membuat PythonOperator untuk menjalankan fungsi push_variable_to_xcom.

- Membuat PythonOperator untuk menjalankan fungsi pull_multiple_value_once.
- Menentukan urutan eksekusi tugas. start dijalankan pertama, diikuti oleh push_variable_to_xcom, dan kemudian pull_multiple_value_once.

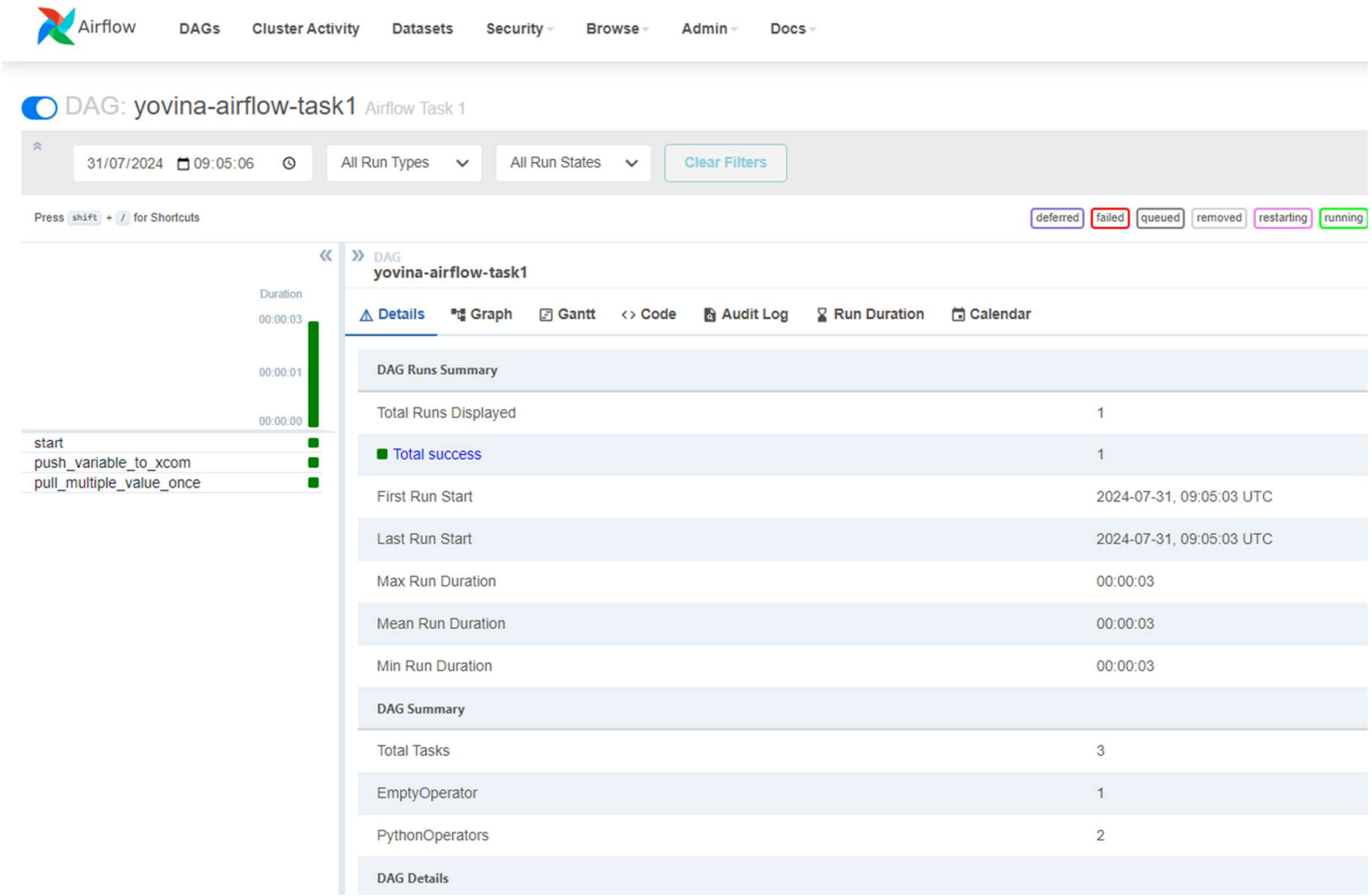
```
def pull_multiple_value_once(ti=None):
    job_role = ti.xcom_pull(task_ids='push_var_job_role', key='job_role')
    job_role_1 = ti.xcom_pull(task_ids='push_var_job_role', key='job_role_1')
    job_role_2 = ti.xcom_pull(task_ids='push_var_job_role', key='job_role_2')
    job_role_3 = ti.xcom_pull(task_ids='push_var_job_role', key='job_role_3')

    print(f'print job_role variable from xcom: {job_role}, {job_role_1}, {job_role_2}, {job_role_3}')

push_variable_to_xcom = PythonOperator(
    task_id = 'push_variable_to_xcom',
    python_callable = push_variable_to_xcom
)

pull_multiple_value_once = PythonOperator(
    task_id = 'pull_multiple_value_once',
    python_callable = pull_multiple_value_once
)

start >> push_variable_to_xcom >> pull_multiple_value_once
```

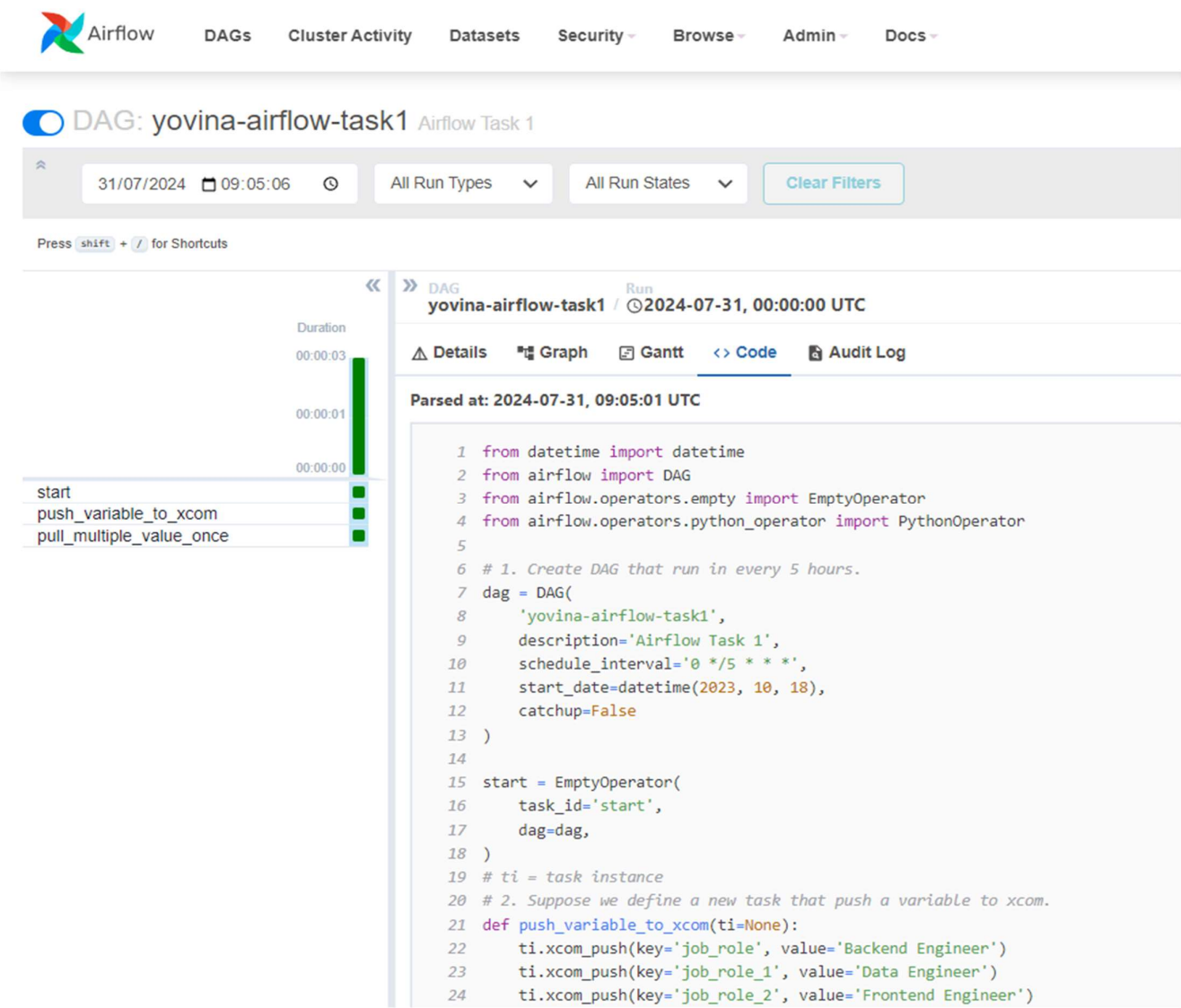
Dalam kode ini, DAG akan dijalankan setiap 5 jam. Tugas pertama push_variable_to_xcom akan mendorong beberapa variabel ke XCom. Setelah itu, tugas pull_multiple_value_once akan menarik variabel-variabel tersebut dari XCom dan mencetaknya. Urutan tugas ini memastikan bahwa variabel didorong ke XCom sebelum ditarik oleh tugas selanjutnya.

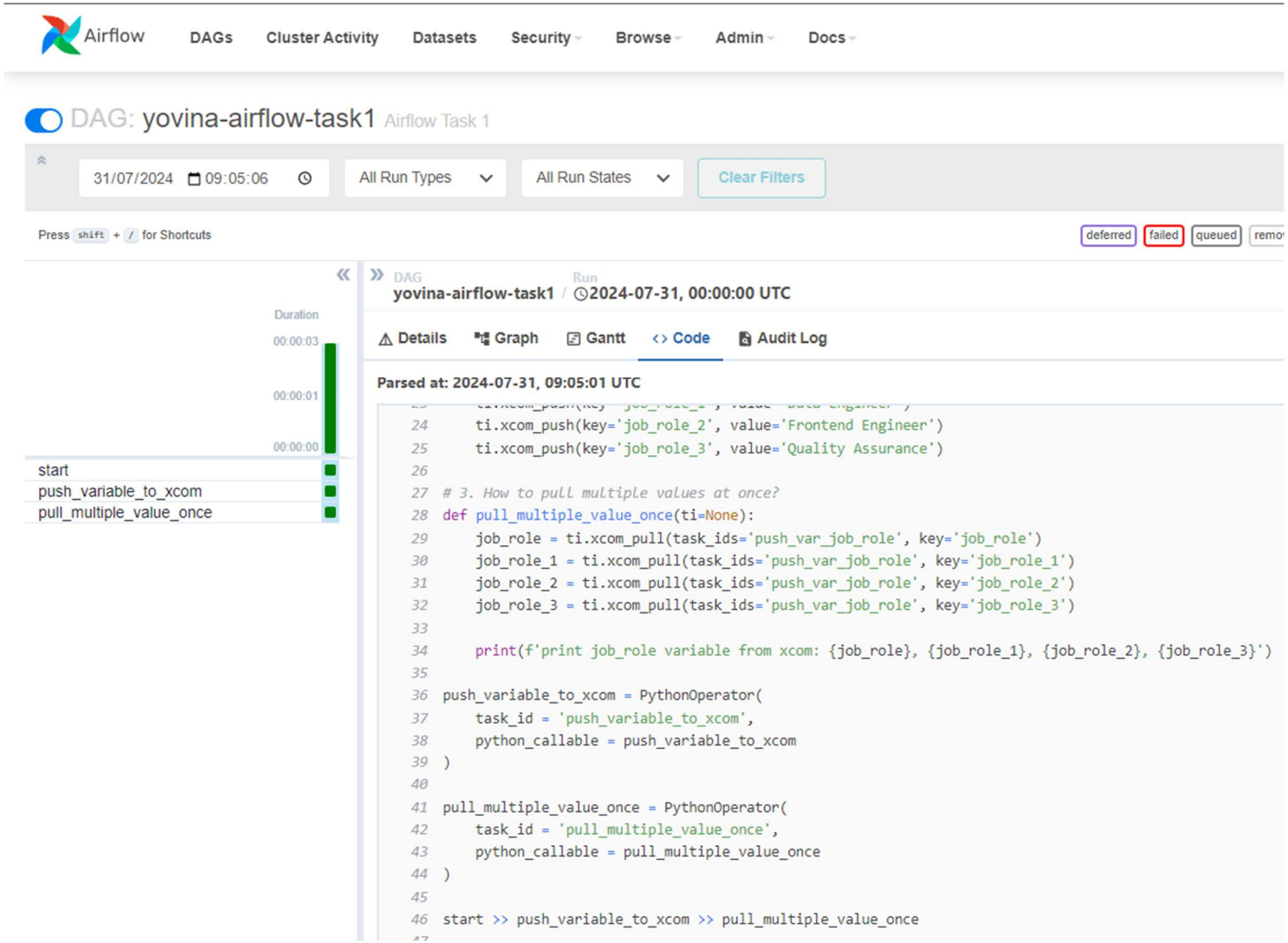
- Tampilan codingan dengan perintah nano di terminal untuk input ke airflownya:

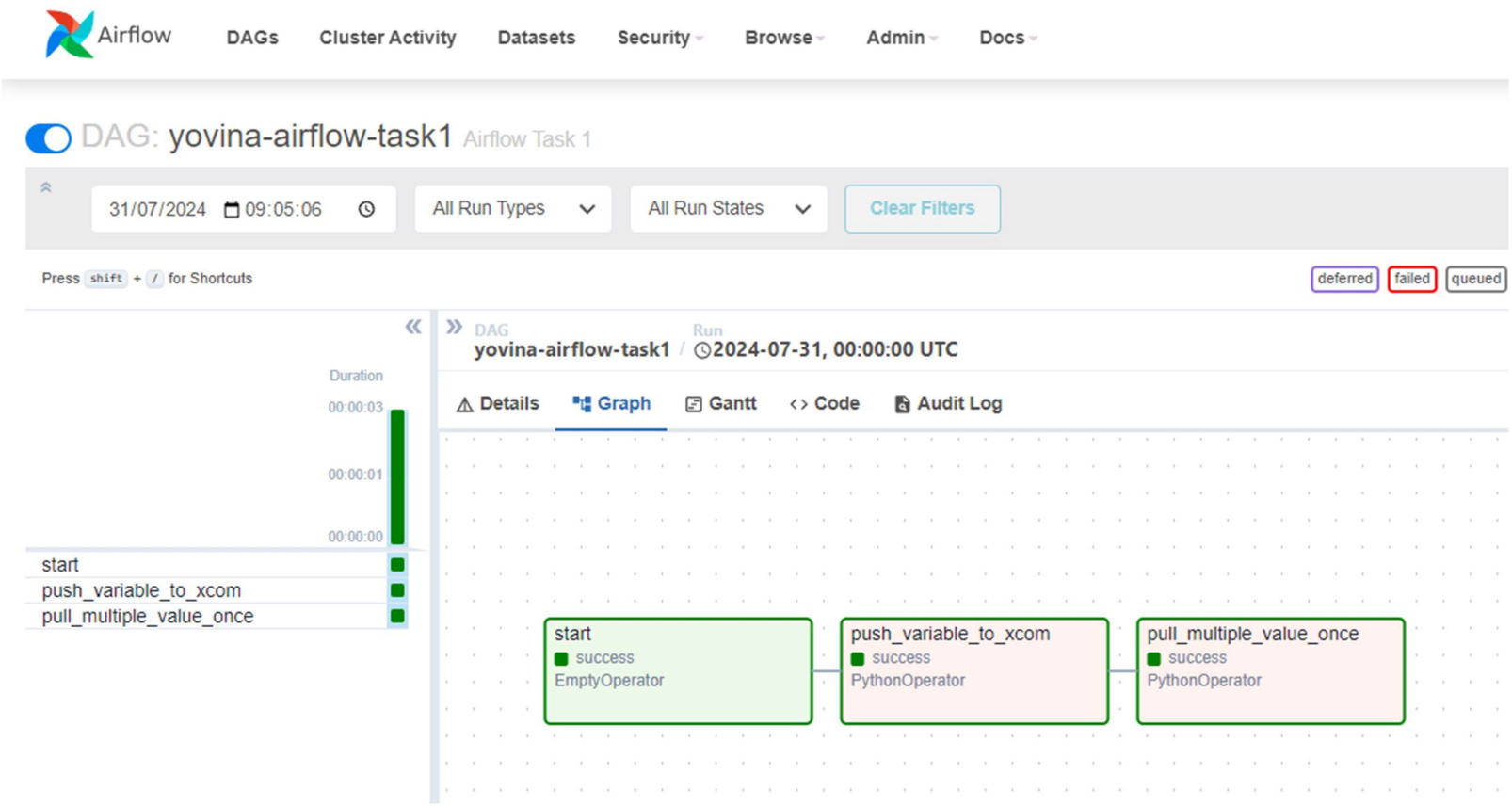
```
  GNU nano 7.2                                                    yovina_airflow_task1.py
from datetime import datetime
from airflow import DAG
from airflow.operators.empty import EmptyOperator
from airflow.operators.python_operator import PythonOperator

# 1. Create DAG that run in every 5 hours.
dag = DAG(
    'yovina-airflow-task1',
    description='Airflow Task 1',
    schedule_interval='0 */5 * * *',
    start_date=datetime(2023, 10, 18),
    catchup=False
)

start = EmptyOperator(
    task_id='start',
    dag=dag,
)
# ti = task instance
# 2. Suppose we define a new task that push a variable to xcom.
def push_variable_to_xcom(ti=None):
    ti.xcom_push(key='job_role', value='Backend Engineer')
    ti.xcom_push(key='job_role_1', value='Data Engineer')
    ti.xcom_push(key='job_role_2', value='Frontend Engineer')
    ti.xcom_push(key='job_role_3', value='Quality Assurance')

# 3. How to pull multiple values at once?
def pull_multiple_value_once(ti=None):
    job_role = ti.xcom_pull(task_ids='push_var_job_role', key='job_role')
    job_role_1 = ti.xcom_pull(task_ids='push_var_job_role', key='job_role_1')
    job_role_2 = ti.xcom_pull(task_ids='push_var_job_role', key='job_role_2')
    job_role_3 = ti.xcom_pull(task_ids='push_var_job_role', key='job_role_3')

    print(f'print job_role variable from xcom: {job_role}, {job_role_1}, {job_role_2}, {job_role_3}')

push_variable_to_xcom = PythonOperator(
    task_id = 'push_variable_to_xcom',
    python_callable = push_variable_to_xcom
)

pull_multiple_value_once = PythonOperator(
    task_id = 'pull_multiple_value_once',
    python_callable = pull_multiple_value_once
)

start >> push_variable_to_xcom >> pull_multiple_value_once
```

- Tampilan pada airflownya:




Airflow
DAGs
Cluster Activity
Datasets
Security
Browse
Admin
Docs
DAG: yovina-airflow-task1 Airflow Task 1
31/07/2024 09:05:06
All Run Types
All Run States
Clear Filters
Press shift + / for Shortcuts
Duration
00:00:03
00:00:01
00:00:00
start
push_variable_to_xcom
pull_multiple_value_once
DAG
yovina-airflow-task1
Run
2024-07-31, 00:00:00 UTC
Details
Graph
Gantt
Code
Audit Log
Parsed at: 2024-07-31, 09:05:01 UTC
from datetime import datetime
from airflow import DAG
from airflow.operators.empty import EmptyOperator
from airflow.operators.python_operator import PythonOperator
# 1. Create DAG that run in every 5 hours.
dag = DAG(
'yovina-airflow-task1',
description='Airflow Task 1',
schedule_interval='0 */5 * * *',
start_date=datetime(2023, 10, 18),
catchup=False
)
start = EmptyOperator(
task_id='start',
dag=dag,
)
# ti = task instance
# 2. Suppose we define a new task that push a variable to xcom.
def push_variable_to_xcom(ti=None):
ti.xcom_push(key='job_role', value='Backend Engineer')
ti.xcom_push(key='job_role_1', value='Data Engineer')
ti.xcom_push(key='job_role_2', value='Frontend Engineer')


Airflow
DAGs
Cluster Activity
Datasets
Security
Browse
Admin
Docs
DAG: yovina-airflow-task1 Airflow Task 1
31/07/2024 09:05:06
All Run Types
All Run States
Clear Filters
Press shift + / for Shortcuts
deferred
failed
queued
DAG
yovina-airflow-task1
Run
2024-07-31, 00:00:00 UTC
Duration
00:00:03
00:00:01
00:00:00
start
push_variable_to_xcom
pull_multiple_value_once
Details
Graph
Gantt
Code
Audit Log
Parsed at: 2024-07-31, 09:05:01 UTC
ti.xcom_push(key='job_role_2', value='Frontend Engineer')
ti.xcom_push(key='job_role_3', value='Quality Assurance')
# 3. How to pull multiple values at once?
def pull_multiple_value_once(ti=None):
job_role = ti.xcom_pull(task_ids='push_var_job_role', key='job_role')
job_role_1 = ti.xcom_pull(task_ids='push_var_job_role', key='job_role_1')
job_role_2 = ti.xcom_pull(task_ids='push_var_job_role', key='job_role_2')
job_role_3 = ti.xcom_pull(task_ids='push_var_job_role', key='job_role_3')
print(f'print job_role variable from xcom: {job_role}, {job_role_1}, {job_role_2}, {job_role_3}')
push_variable_to_xcom = PythonOperator(
task_id = 'push_variable_to_xcom',
python_callable = push_variable_to_xcom
)
pull_multiple_value_once = PythonOperator(
task_id = 'pull_multiple_value_once',
python_callable = pull_multiple_value_once
)
start >> push_variable_to_xcom >> pull_multiple_value_once


Airflow
DAGs
Cluster Activity
Datasets
Security
Browse
Admin
Docs
DAG: yovina-airflow-task1 Airflow Task 1
31/07/2024 09:05:06
All Run Types
All Run States
Clear Filters
Press shift + / for Shortcuts
deferred
failed
queued
DAG
yovina-airflow-task1
Run
2024-07-31, 00:00:00 UTC
Duration
00:00:03
00:00:01
00:00:00
start
push_variable_to_xcom
pull_multiple_value_once
Details
Graph
Gantt
Code
Audit Log
start
success
EmptyOperator
push_variable_to_xcom
success
PythonOperator
pull_multiple_value_once
success
PythonOperator