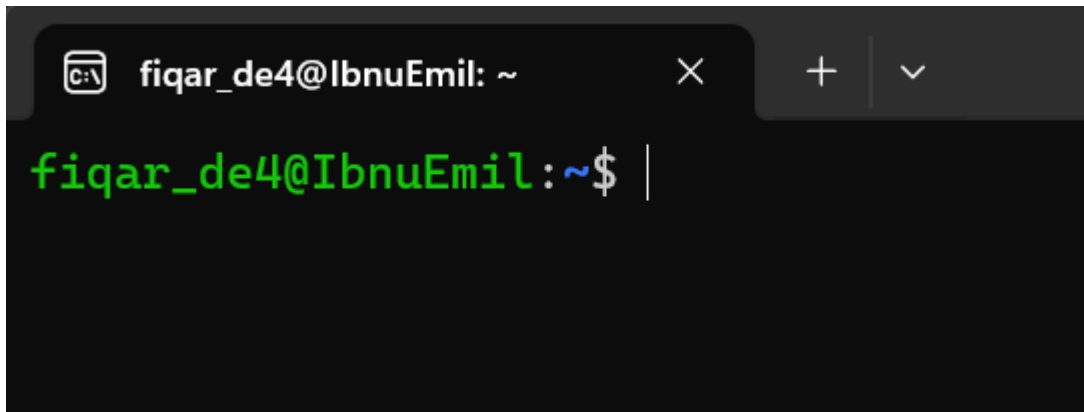


Part 1 – Airflow

1. Buka aplikasi Ubuntu terlebih dahulu.



2. Kemudian masukkan secure shell “ssh raja_rahmanakmaludin@34.101.224.54” dan password “mentoralterra2024”.

```
fiqar_de4@IbnuEmil:~$ 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: Thu Aug  1 12:03:38 2024 from 203.175.125.135
```

Berikut ini tampilan setelah melakukan langkah nomor 2.

3. Lalu ketik ls untuk melihat file atau folder apa saja yang ada.

```
raja_rahmanakmaludin@instance-20240714-035051:~$ ls
2022-01-01-1.json.gz airflow-data dbt-demo extract-load-demo ingestion-data streaming-platform
```

4. Lalu masuk ke folder airflow-data dengan mengetik cd airflow-data dan masuk lagi ke folder docker di dalam folder tersebut dengan mengetik cd docker.

```
raja_rahmanakmaludin@instance-20240714-035051:~$ cd airflow-data
raja_rahmanakmaludin@instance-20240714-035051:~/airflow-data$ ls
TASK-1 airflow.cfg      day-2-airflow.md  day-4-airflow.md  img
TASK-2 day-1-airflow.md  day-3-airflow.md  docker             requirements.txt
raja_rahmanakmaludin@instance-20240714-035051:~/airflow-data$ cd docker
raja_rahmanakmaludin@instance-20240714-035051:~/airflow-data/docker$ |
```

5. Kemudian di dalam folder docker ketik ls untuk mengetahui isi dari folder docker tersebut dan masuk ke folder dags dengan mengetik cd dags.

```
raja_rahmanakmaludin@instance-20240714-035051:~/airflow-data/docker$ ls
Dockerfile config dags docker-compose.yaml farhan-test1.py farhan_test.py logs plugins transformation
raja_rahmanakmaludin@instance-20240714-035051:~/airflow-data/docker$ |
```

6. Setelah itu masuk di folder dags.

```
raja_rahmanakmaludin@instance-20240714-035051:~/airflow-data/docker$ cd dags
raja_rahmanakmaludin@instance-20240714-035051:~/airflow-data/docker/dags$ ls
__pycache__          etl_github_data.py      kharisma-airflow-task2.py
alterra_connection_kharisma.py  farhan_task2.py         kharisma_airflow_task1.py
alterra_connection_yovina.py    farhan_task1.py         kharisma_airflow_task2.py
alterra_hasda_task-1.py        farhan_test.py          loop_print_var_example.py
alterra_hasda_task-2.py        fitri_airflowtask1.py   rais-alltask1-airflow.py
alterra_hasda_task2.py         fitri_airflowtask2.py   rais-alltask2-airflow.py
alterra_hook_kharisma.py       get_var_example.py      sample.csv
alterra_hook_yovina.py         hello_world.py          wartadi_task1_airflow.py
alterra_nurhasanahdarus_task1.py  hello_world_operator.py wartadi_task1_airflow_new.py
alterra_tes_connection_khairullah.py  hook_example.py        wartadi_task2_airflow.py
connection_dimas_task2.py       hook_example_dimas.py   xcom_example_decorator.py
connection_example.py           hook_example_fitri.py   xcom_example_native.py
connection_example_dimas.py     ingestion.py            yovina_airflow_task1.py
connection_example_fitri.py     integrate_all.py        yovina_airflow_task2.py
dag_github_data.py             integrate_all_part2.py   zola_task1_airflow.py
dimas-alltask1-airflow.py       integrate_all_part3.py  zola_task2_airflow.py
dimas-alltask2-airflow.py       khairullah_airflow_task1.py
dimas-task1-airflow.py          kharisma-airflow-task2-new.py
```

Berikut adalah isi dari folder dags. Kemudian buat file ekstensi python dengan mengetik `sudo nano Fiqar_Airflow_Task1.py`.

7. Lalu membuat code dalam file tersebut:

```
GNU nano 7.2 Fiqar_Airflow_Task1.py
from datetime import datetime
from airflow import DAG
from airflow.operators.python_operator import PythonOperator

# 1. Create DAG that run in every 5 hours:

with DAG(
    'Fiqar_Airflow_Task1',
    description='Alterra Airflow Task 1',
    schedule_interval='0 */5 * * *',
    start_date=datetime(2023, 1, 1),
    catchup=False
) as dag:

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

# ti = task instance
def push_variable_to_xcom(ti=None):
    ti.xcom_push(key='job_level1', value='Data Engineer')
    ti.xcom_push(key='job_level2', value='Data Architect')
    ti.xcom_push(key='job_level3', value='Database Administrator')

# 3. How to pull multiple values at once?:

def pull_multiple_value_once(ti=None):
    job_level1 = ti.xcom_pull(task_ids='push_variable_to_xcom', key='job_level1')
    job_level2 = ti.xcom_pull(task_ids='push_variable_to_xcom', key='job_level2')
    job_level3 = ti.xcom_pull(task_ids='push_variable_to_xcom', key='job_level3')

    print(f'print job_level variable from xcom: {job_level1}, {job_level2}, {job_level3}')

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
)

push_variable_to_xcom >> pull_multiple_value_once

^G Help      ^O Write Out  ^W Where Is   ^R Cut        ^T Execute    ^C Location   ^U Undo       ^M Set Mark
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/_ Go To Line ^E Redo       ^-6 Copy
```

Setelah itu tekan tombol kombinasi `Ctrl + X`, kemudian `Y`, kemudian `Enter` untuk keluar dari mode nano.

8. Kemudian ketik ls lagi untuk memastikan apakah file yang kita buat tadi sudah berhasil dibuat.

```
raja_rahmanakmaludin@instance-20240714-035051:~/airflow-data/docker/dags$ ls
Fiqar_Airflow_Task1.py      dimas-task1-airflow.py      kharisma-airflow-task2-new.py
__pycache__                 etl_github_data.py          kharisma-airflow-task2.py
alterra_connection_khari    farhan_task2.py             kharisma_airflow_task1.py
alterra_connection_yovina.py farhan_task_1.py             kharisma_airflow_task2.py
alterra_hasda_task-1.py     farhan_test.py              loop_print_var_example.py
alterra_hasda_task-2.py     fitri_airflowtask1.py        rais-alltask1-airflow.py
alterra_hasda_task2.py      fitri_airflowtask2.py        rais-alltask2-airflow.py
alterra_hook_kharisma.py    get_var_example.py           sample.csv
alterra_hook_yovina.py      hello_world.py               wartadi_task1_airflow.py
alterra_nurhasanahdarus_task1.py hello_world_operator.py      wartadi_task1_airflow_new.py
alterra_tes_connection_khairullah.py hook_example.py              wartadi_task2_airflow.py
connection_dimas_task2.py    hook_example_dimas.py        xcom_example_decorator.py
connection_example.py        hook_example_fitri.py         xcom_example_native.py
connection_example_dimas.py  ingestion.py                  yovina_airflow_task1.py
connection_example_fitri.py  integrate_all.py              yovina_airflow_task2.py
dag_github_data.py          integrate_all_part2.py        zola_task1_airflow.py
dimas-alltask1-airflow.py    integrate_all_part3.py        zola_task2_airflow.py
dimas-alltask2-airflow.py    khairullah_airflow_task1.py
```



File Fiqr_Airflow_Task1.py sudah berhasil dibuat.

9. Kemudian masukkan URL <http://34.101.224.54:8080/> untuk masuk ke server Airflow. Setelah itu pada website tersebut, masukkan username “airflow” dan password “airflow” untuk masuk ke website tersebut.



Sign In


Enter your login and password below:

Username:

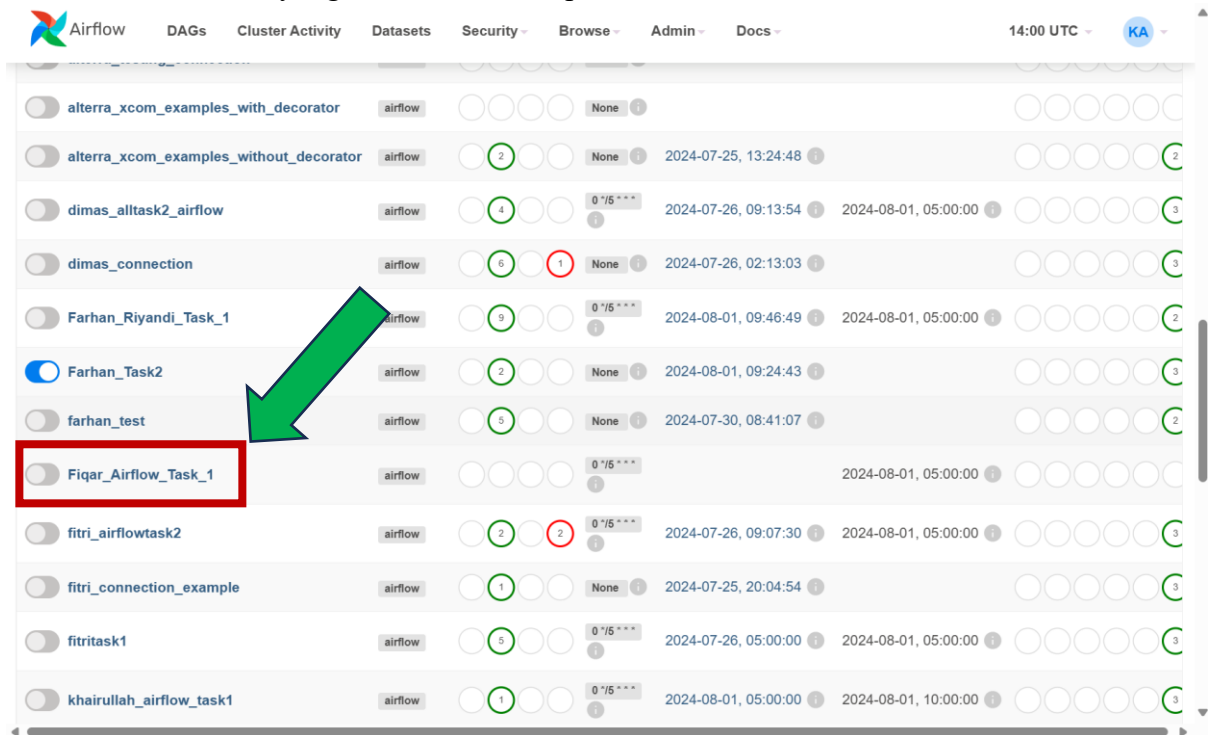
 airflow 

Password:

Sign In 

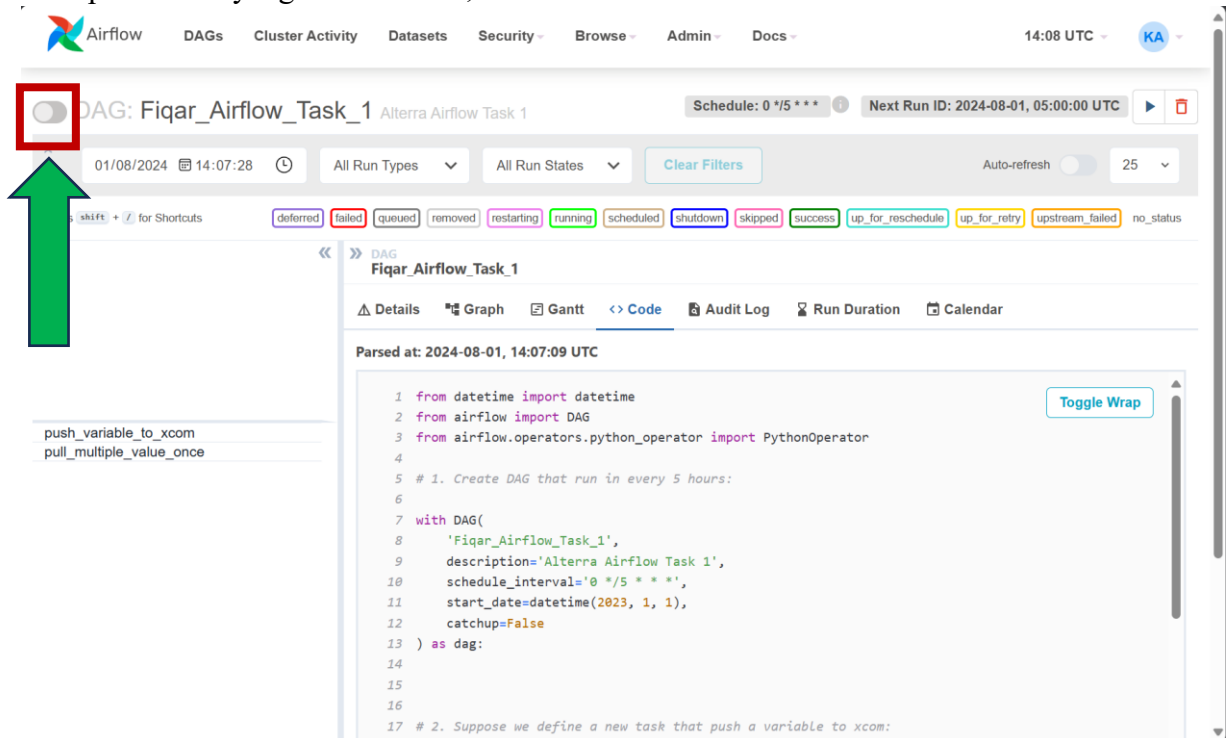
10. Setelah itu cek DAG yang sudah kita buat apakah sudah ada atau belum.



The screenshot shows the Airflow web interface with a list of DAGs. The DAG 'Fiqar_Airflow_Task_1' is highlighted with a red box. A green arrow points to the toggle switch for this DAG, which is currently turned off. The table lists various DAGs with their names, types, last run times, and next run times.

DAG Name	Type	Last Run Time	Next Run Time
alterra_xcom_examples_with_decorator	airflow	None	None
alterra_xcom_examples_without_decorator	airflow	2024-07-25, 13:24:48	2024-07-25, 13:24:48
dimas_alltask2_airflow	airflow	2024-07-26, 09:13:54	2024-08-01, 05:00:00
dimas_connection	airflow	2024-07-26, 02:13:03	2024-07-26, 02:13:03
Farhan_Riyandi_Task_1	airflow	2024-08-01, 09:46:49	2024-08-01, 05:00:00
Farhan_Task2	airflow	2024-08-01, 09:24:43	2024-08-01, 09:24:43
farhan_test	airflow	2024-07-30, 08:41:07	2024-07-30, 08:41:07
Fiqar_Airflow_Task_1	airflow	2024-08-01, 05:00:00	2024-08-01, 05:00:00
fitri_airflowtask2	airflow	2024-07-26, 09:07:30	2024-08-01, 05:00:00
fitri_connection_example	airflow	2024-07-25, 20:04:54	2024-07-25, 20:04:54
fitritask1	airflow	2024-07-26, 05:00:00	2024-08-01, 05:00:00
khairullah_airflow_task1	airflow	2024-08-01, 05:00:00	2024-08-01, 10:00:00

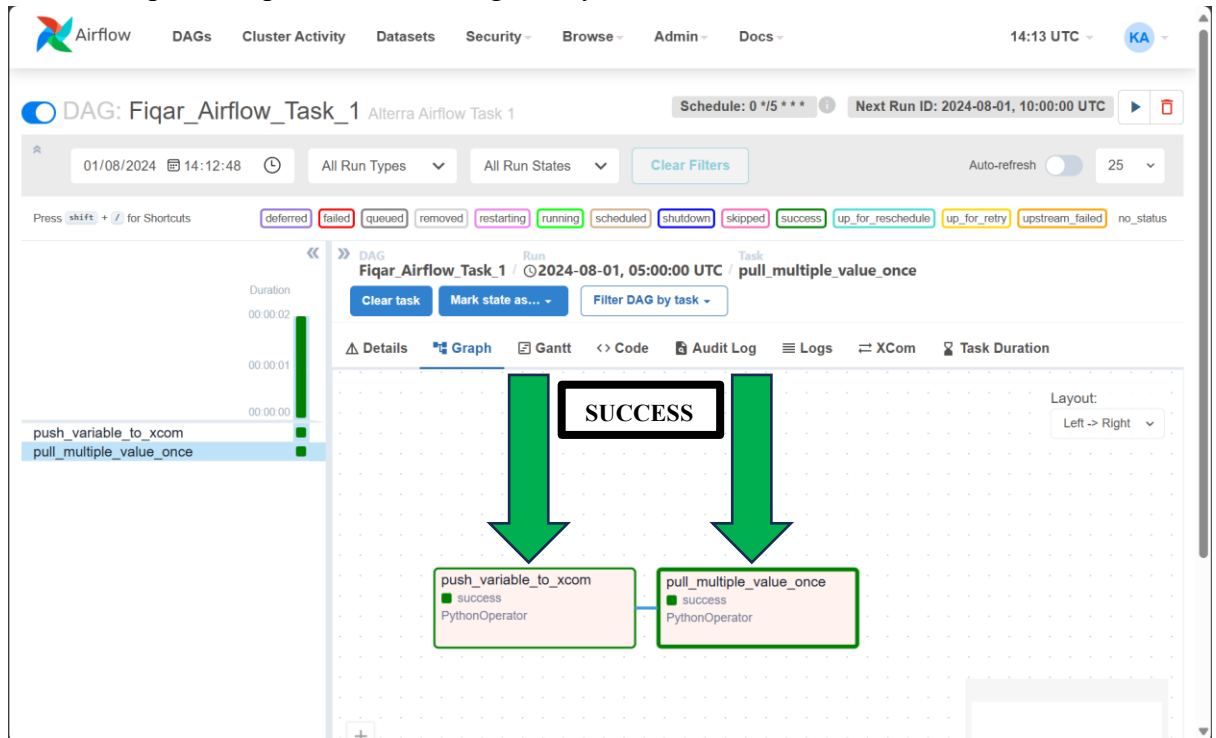
11. Lalu pilih DAG yang sudah dibuat, kemudian aktifkan DAG tersebut.



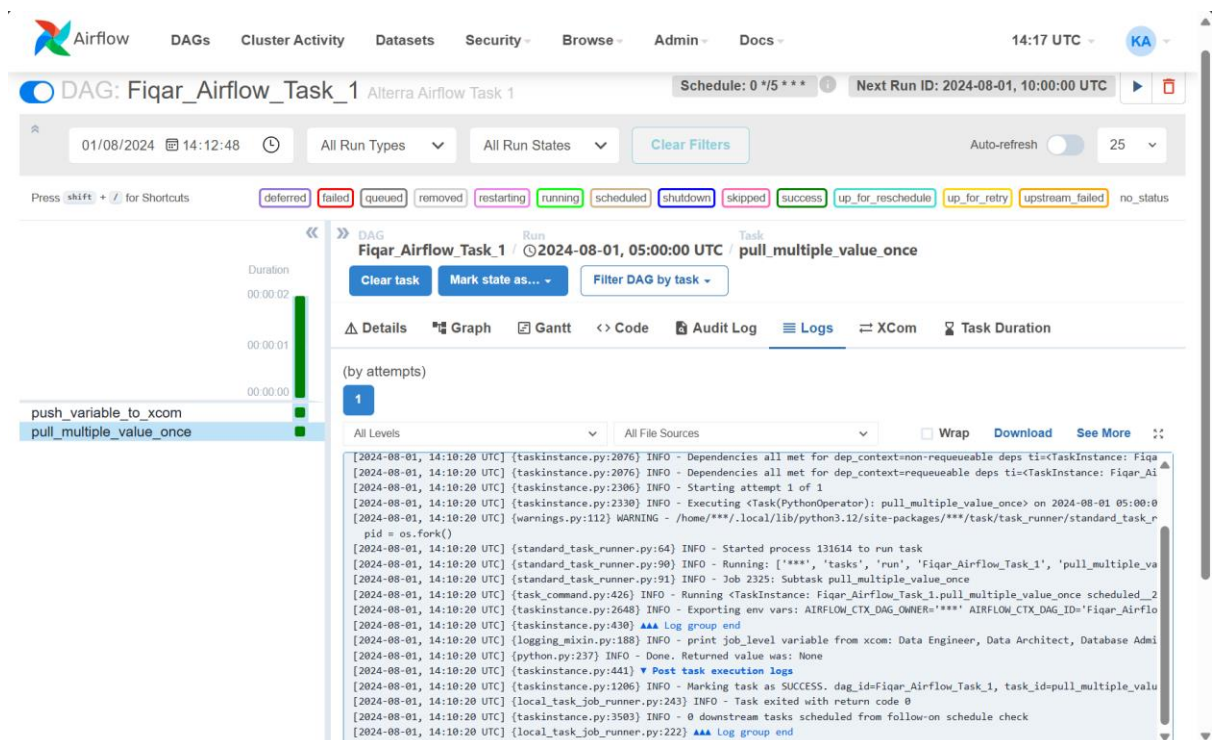
The screenshot shows the Airflow web interface with the DAG 'Fiqar_Airflow_Task_1' selected. The DAG is currently disabled (toggle switch is off). The code for the DAG is displayed in the 'Code' tab, showing the DAG definition and the task definition.

```
1 from datetime import datetime
2 from airflow import DAG
3 from airflow.operators.python_operator import PythonOperator
4
5 # 1. Create DAG that run in every 5 hours:
6
7 with DAG(
8     'Fiqar_Airflow_Task_1',
9     description='Alterra Airflow Task 1',
10    schedule_interval='0 */5 * * *',
11    start_date=datetime(2023, 1, 1),
12    catchup=False
13 ) as dag:
14
15
16
17 # 2. Suppose we define a new task that push a variable to xcom:
```

12. Kemudian pilih Graph untuk melihat grafiknya.



Hasilnya DAG sudah berhasil dengan ditandai tulisan success pada kotak yang ditunjuk panah hijau tersebut.



Dalam logs juga menunjukkan bahwa DAG sudah berhasil berjalan sesuai dengan yang diinginkan.