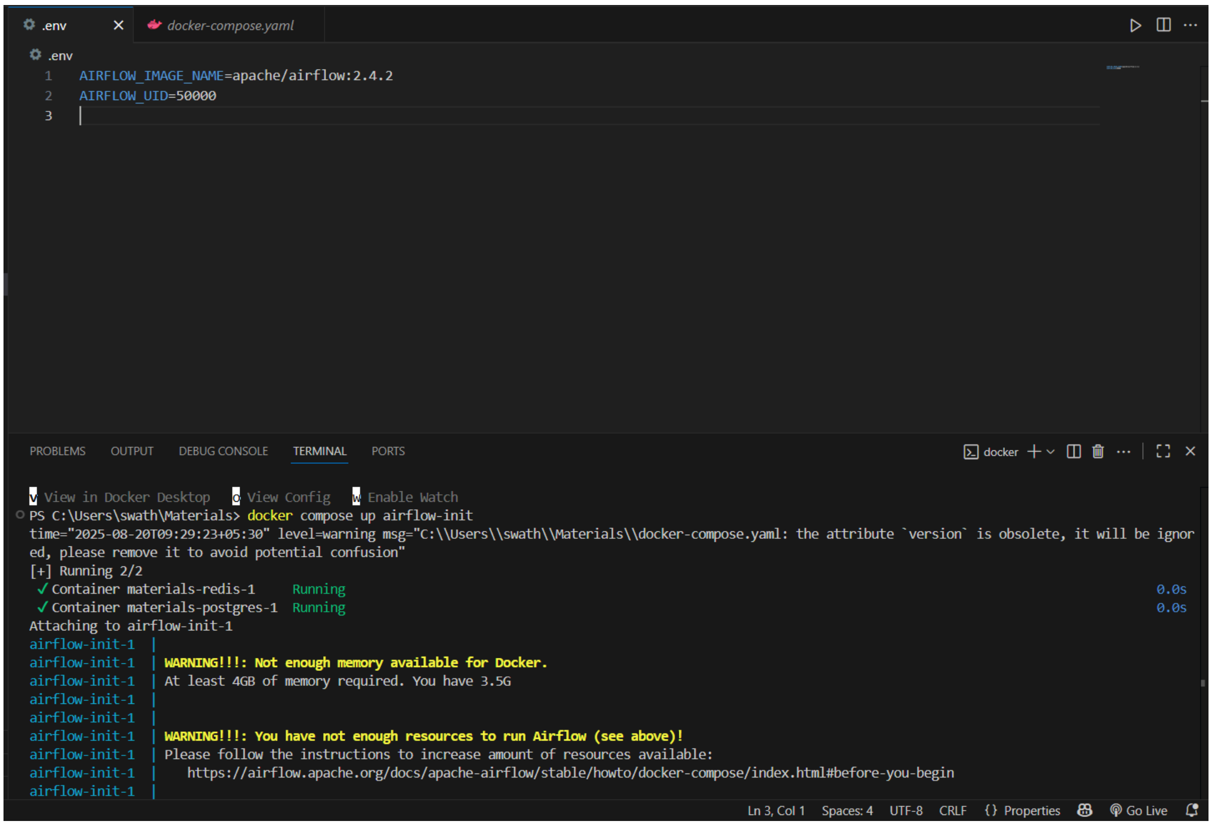
# ASSIGNMENT – 3 (Apache Airflow)

# TRAINEE NAME: Swathi Baskaran

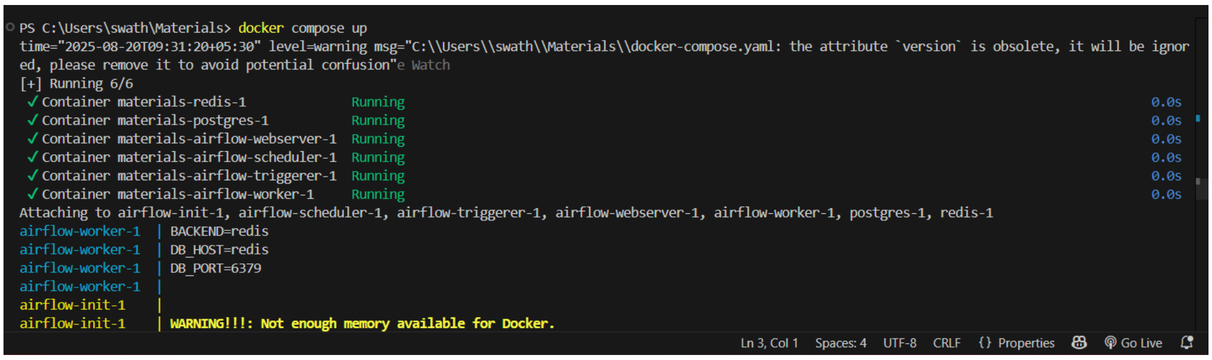
## Setting up airflow-init in VS Code



```
AIRFLOW_IMAGE_NAME=apache/airflow:2.4.2
AIRFLOW_UID=50000
```

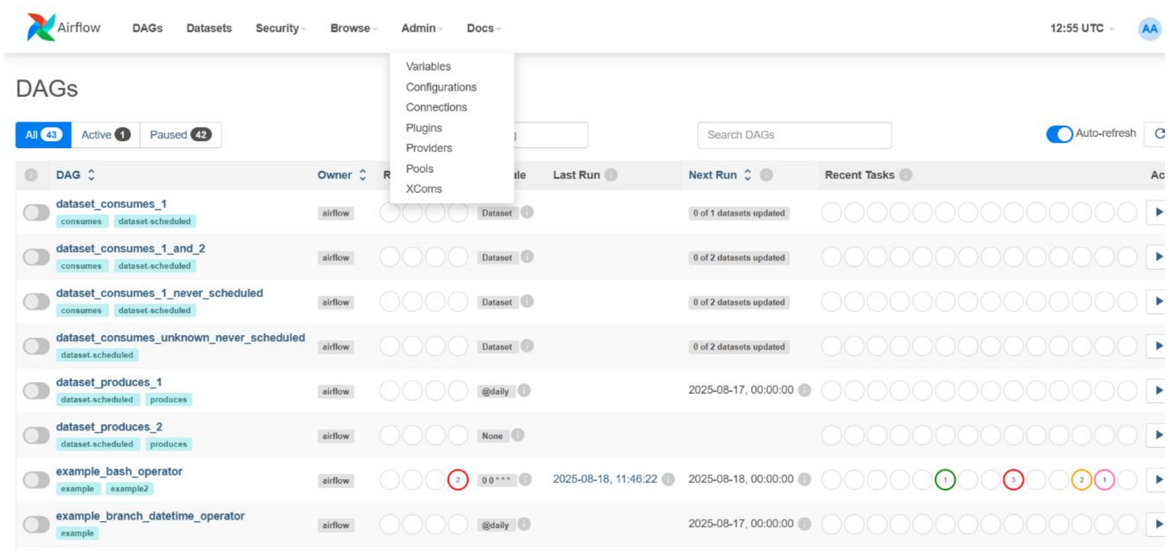
```
v View in Docker Desktop   o View Config   w Enable Watch
PS C:\Users\swath\Materials> docker compose up airflow-init
time="2025-08-20T09:29:23+05:30" level=warning msg="C:\\Users\\swath\\Materials\\docker-compose.yaml: the attribute `version` is obsolete, it will be ignor
ed, please remove it to avoid potential confusion"
[+] Running 2/2
 ✓Container materials-redis-1     Running                                                                                          0.0s
 ✓Container materials-postgres-1  Running                                                                                          0.0s
Attaching to airflow-init-1
airflow-init-1  |
airflow-init-1  | WARNING!!!: Not enough memory available for Docker.
airflow-init-1  | At least 4GB of memory required. You have 3.5G
airflow-init-1  |
airflow-init-1  |
airflow-init-1  | WARNING!!!: You have not enough resources to run Airflow (see above)!
airflow-init-1  | Please follow the instructions to increase amount of resources available:
airflow-init-1  |    https://airflow.apache.org/docs/apache-airflow/stable/howto/docker-compose/index.html#before-you-begin
airflow-init-1  |
```

## Starting up all services



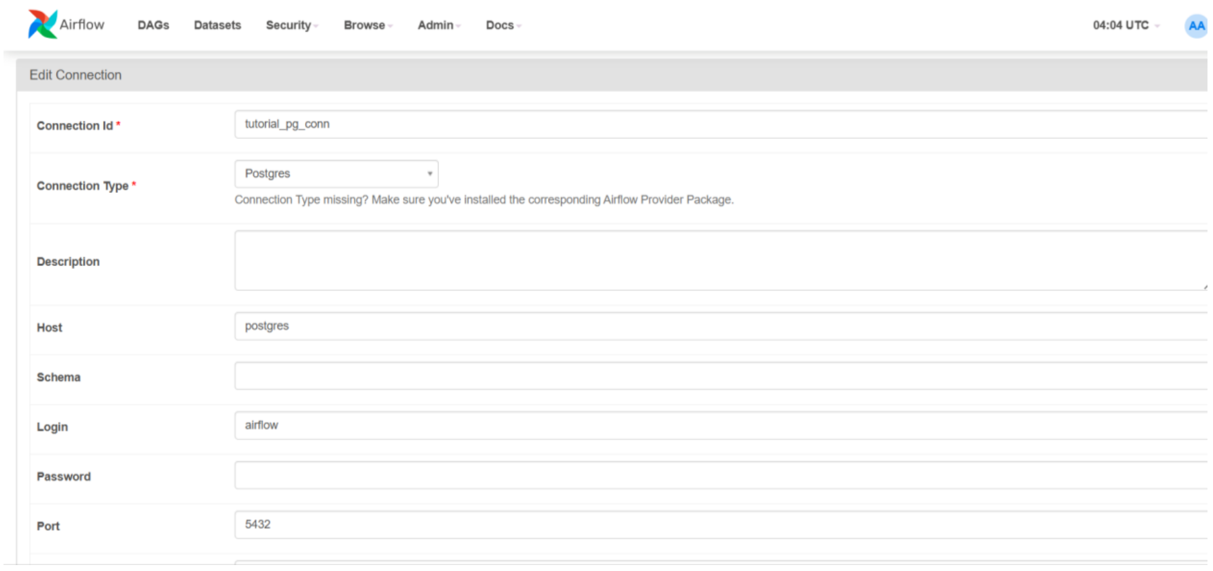
```
PS C:\Users\swath\Materials> docker compose up
time="2025-08-20T09:31:20+05:30" level=warning msg="C:\\Users\\swath\\Materials\\docker-compose.yaml: the attribute `version` is obsolete, it will be ignor
ed, please remove it to avoid potential confusion"e Watch
[+] Running 6/6
 ✓Container materials-redis-1              Running                                                                                 0.0s
 ✓Container materials-postgres-1           Running                                                                                 0.0s
 ✓Container materials-airflow-webserver-1  Running                                                                                 0.0s
 ✓Container materials-airflow-scheduler-1  Running                                                                                 0.0s
 ✓Container materials-airflow-triggerer-1  Running                                                                                 0.0s
 ✓Container materials-airflow-worker-1     Running                                                                                 0.0s
Attaching to airflow-init-1, airflow-scheduler-1, airflow-triggerer-1, airflow-webserver-1, airflow-worker-1, postgres-1, redis-1
airflow-worker-1  | BACKEND=redis
airflow-worker-1  | DB_HOST=redis
airflow-worker-1  | DB_PORT=6379
airflow-worker-1  |
airflow-init-1    |
airflow-init-1    | WARNING!!!: Not enough memory available for Docker.
```

## Setting up a new connection



| DAG | Owner | Schedule | Last Run | Next Run |
|---|---|---|---|---|
| dataset_consumes_1 (consumes, dataset-scheduled) | airflow | Dataset | | 0 of 1 datasets updated |
| dataset_consumes_1_and_2 (consumes, dataset-scheduled) | airflow | Dataset | | 0 of 2 datasets updated |
| dataset_consumes_1_never_scheduled (consumes, dataset-scheduled) | airflow | Dataset | | 0 of 2 datasets updated |
| dataset_consumes_unknown_never_scheduled (dataset-scheduled) | airflow | Dataset | | 0 of 2 datasets updated |
| dataset_produces_1 (dataset-scheduled, produces) | airflow | @daily | | 2025-08-17, 00:00:00 |
| dataset_produces_2 (dataset-scheduled, produces) | airflow | None | | |
| example_bash_operator (example, example2) | airflow | 0 0 * * * | 2025-08-18, 11:46:22 | 2025-08-18, 00:00:00 |
| example_branch_datetime_operator (example) | airflow | @daily | | 2025-08-17, 00:00:00 |

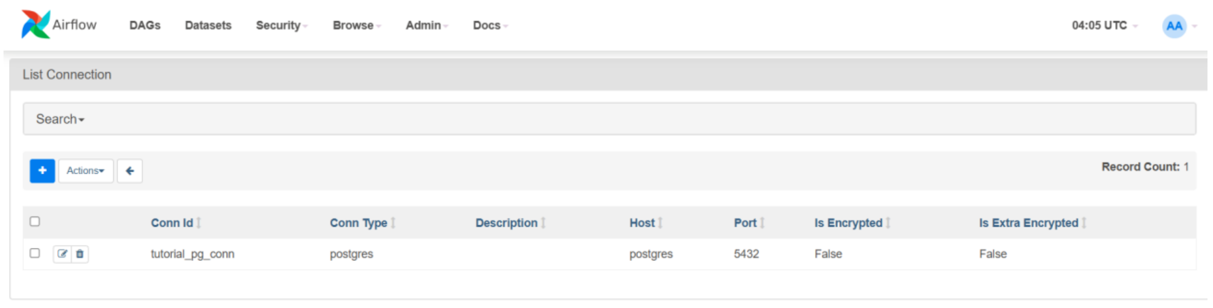
## Entering the connection details



Edit Connection

| | |
|---|---|
| Connection Id * | tutorial_pg_conn |
| Connection Type * | Postgres<br>Connection Type missing? Make sure you've installed the corresponding Airflow Provider Package. |
| Description | |
| Host | postgres |
| Schema | |
| Login | airflow |
| Password | |
| Port | 5432 |

## Connection successfully created

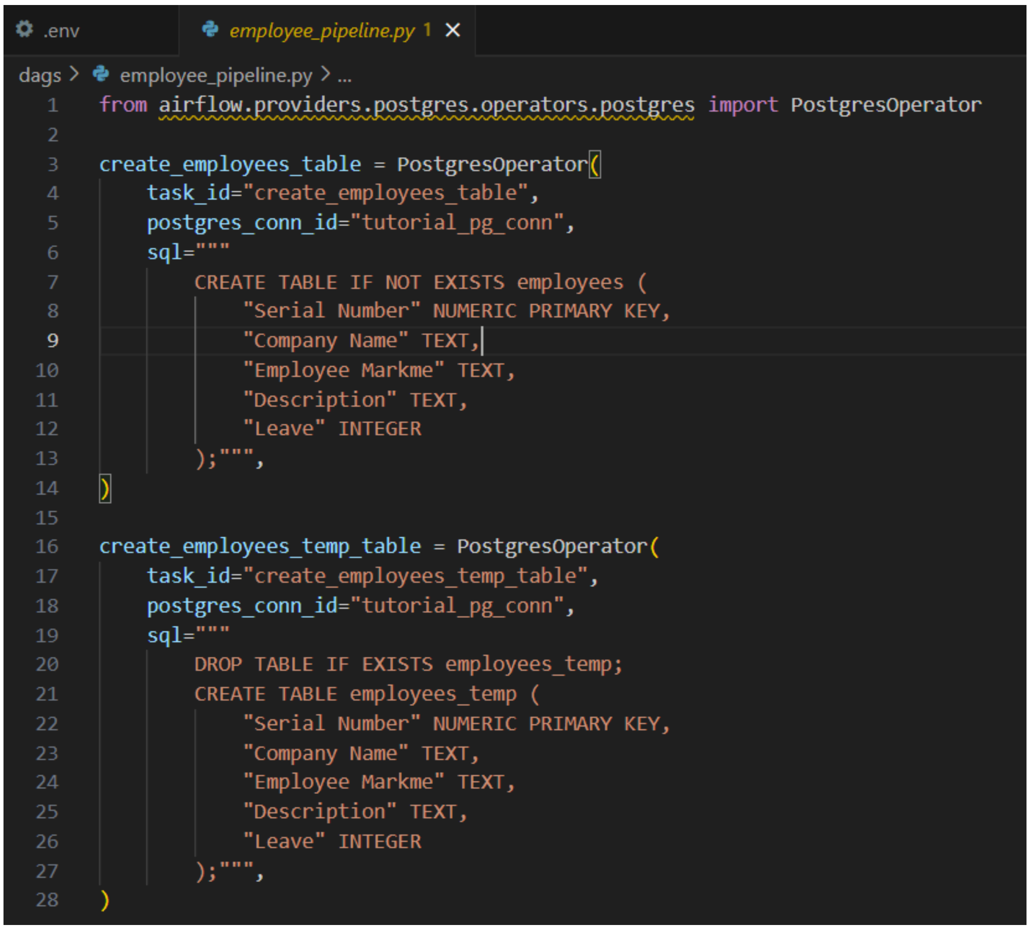


List Connection

Record Count: 1

| Conn Id | Conn Type | Description | Host | Port | Is Encrypted | Is Extra Encrypted |
|---|---|---|---|---|---|---|
| tutorial_pg_conn | postgres | | postgres | 5432 | False | False |

## Create tables for staging and final data

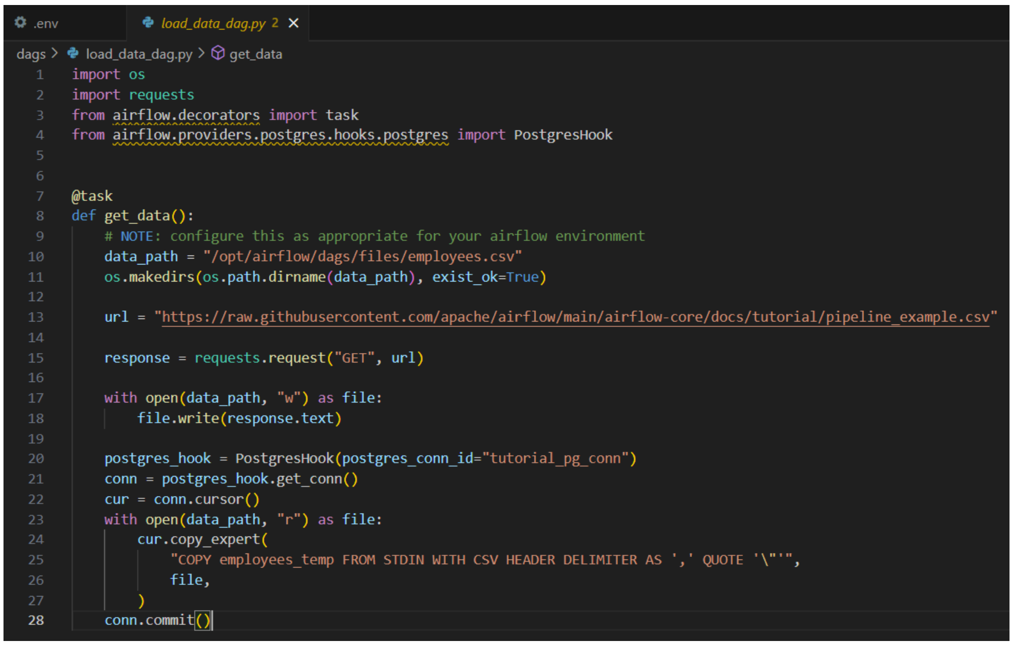


.env | employee_pipeline.py 1

dags > employee_pipeline.py > ...

```python
from airflow.providers.postgres.operators.postgres import PostgresOperator

create_employees_table = PostgresOperator(
    task_id="create_employees_table",
    postgres_conn_id="tutorial_pg_conn",
    sql="""
        CREATE TABLE IF NOT EXISTS employees (
            "Serial Number" NUMERIC PRIMARY KEY,
            "Company Name" TEXT,
            "Employee Markme" TEXT,
            "Description" TEXT,
            "Leave" INTEGER
        );""",
)

create_employees_temp_table = PostgresOperator(
    task_id="create_employees_temp_table",
    postgres_conn_id="tutorial_pg_conn",
    sql="""
        DROP TABLE IF EXISTS employees_temp;
        CREATE TABLE employees_temp (
            "Serial Number" NUMERIC PRIMARY KEY,
            "Company Name" TEXT,
            "Employee Markme" TEXT,
            "Description" TEXT,
            "Leave" INTEGER
        );""",
)
```

## Load data into the staging table

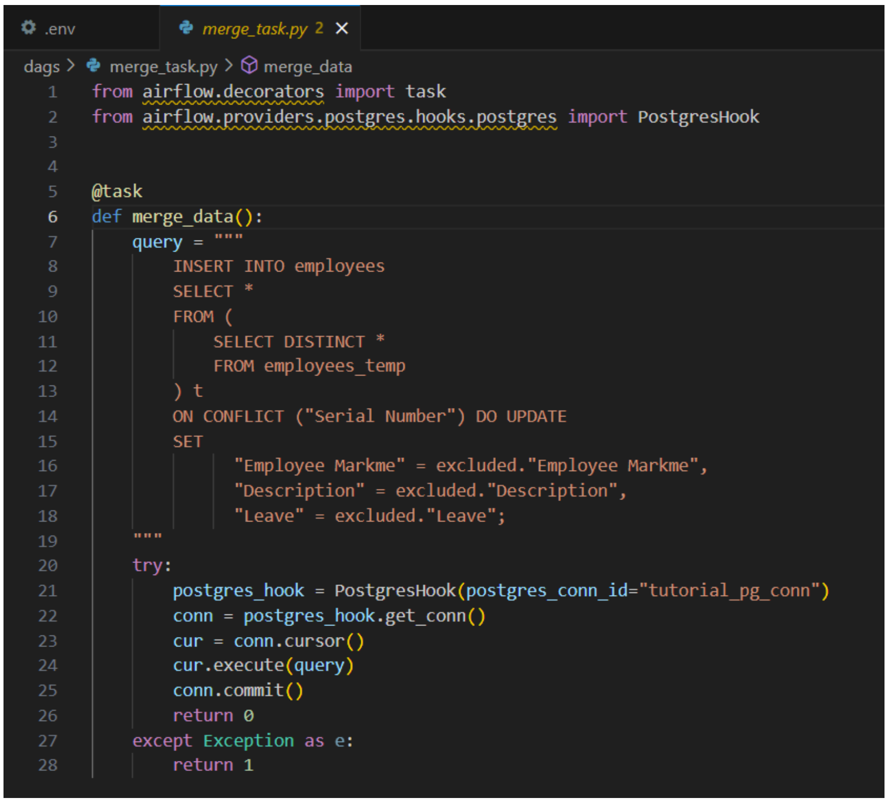


.env | load_data_dag.py 2

dags > load_data_dag.py > get_data

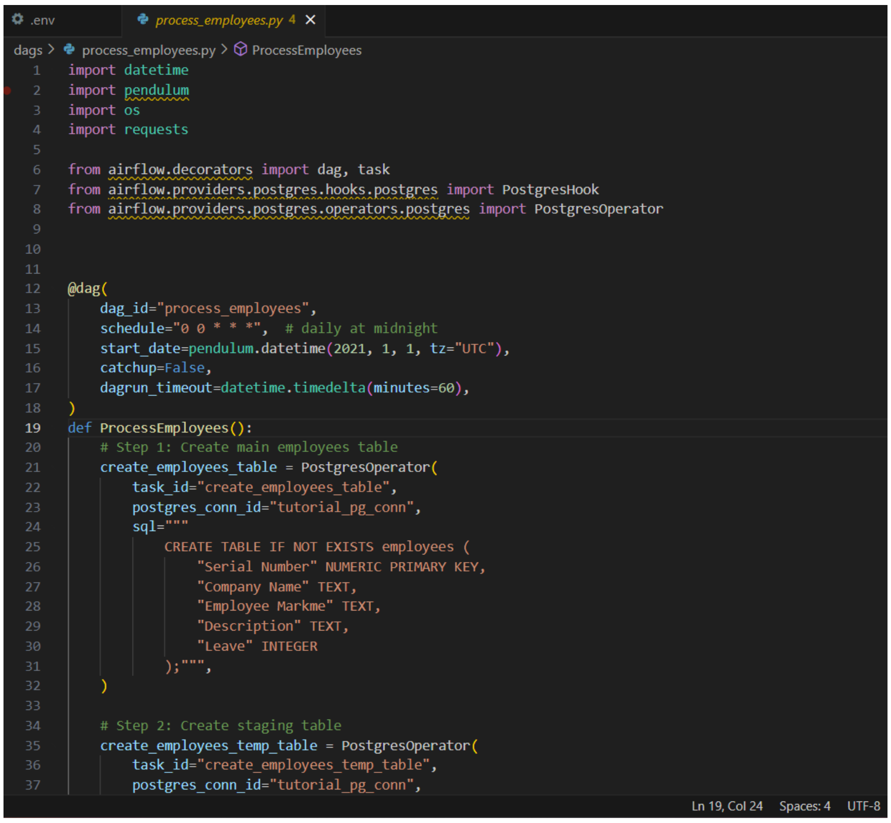
```python
import os
import requests
from airflow.decorators import task
from airflow.providers.postgres.hooks.postgres import PostgresHook


@task
def get_data():
    # NOTE: configure this as appropriate for your airflow environment
    data_path = "/opt/airflow/dags/files/employees.csv"
    os.makedirs(os.path.dirname(data_path), exist_ok=True)

    url = "https://raw.githubusercontent.com/apache/airflow/main/airflow-core/docs/tutorial/pipeline_example.csv"

    response = requests.request("GET", url)

    with open(data_path, "w") as file:
        file.write(response.text)

    postgres_hook = PostgresHook(postgres_conn_id="tutorial_pg_conn")
    conn = postgres_hook.get_conn()
    cur = conn.cursor()
    with open(data_path, "r") as file:
        cur.copy_expert(
            "COPY employees_temp FROM STDIN WITH CSV HEADER DELIMITER AS ',' QUOTE '\"'",
            file,
        )
    conn.commit()
```

## Merge and clean the data

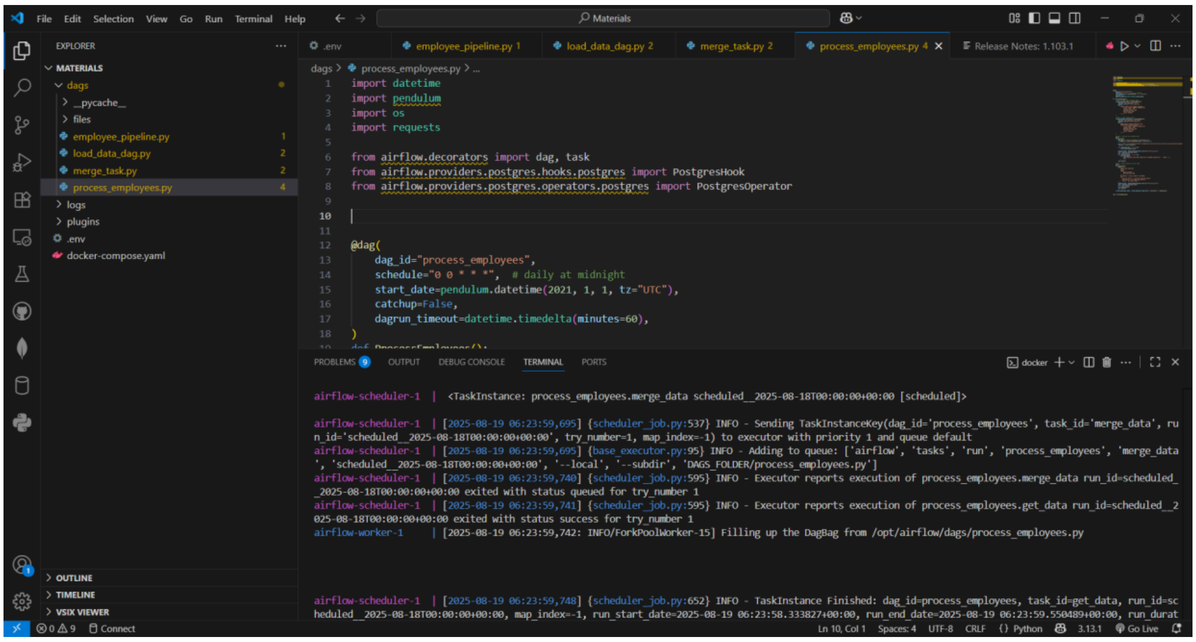


```python
from airflow.decorators import task
from airflow.providers.postgres.hooks.postgres import PostgresHook


@task
def merge_data():
    query = """
        INSERT INTO employees
        SELECT *
        FROM (
            SELECT DISTINCT *
            FROM employees_temp
        ) t
        ON CONFLICT ("Serial Number") DO UPDATE
        SET
              "Employee Markme" = excluded."Employee Markme",
              "Description" = excluded."Description",
              "Leave" = excluded."Leave";
    """
    try:
        postgres_hook = PostgresHook(postgres_conn_id="tutorial_pg_conn")
        conn = postgres_hook.get_conn()
        cur = conn.cursor()
        cur.execute(query)
        conn.commit()
        return 0
    except Exception as e:
        return 1
```

## Defining the DAG



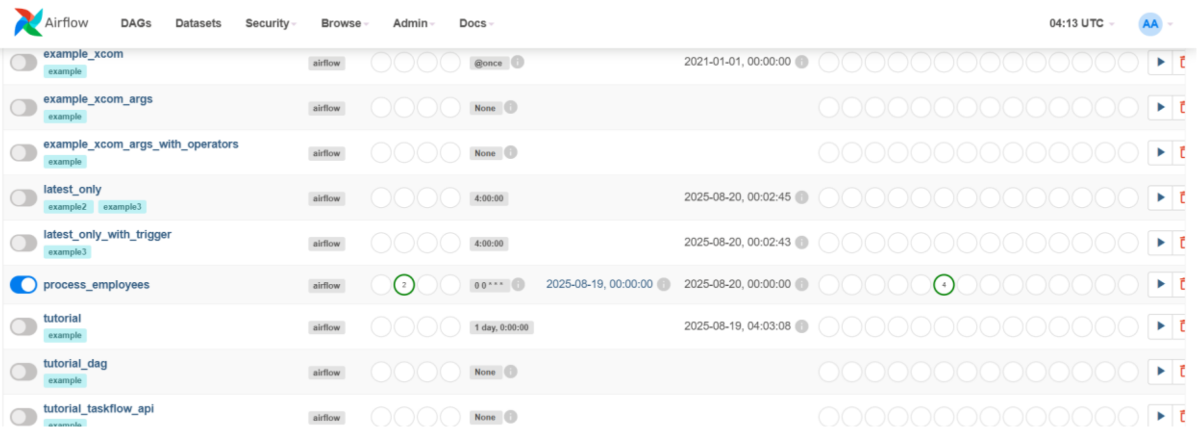
```python
import datetime
import pendulum
import os
import requests

from airflow.decorators import dag, task
from airflow.providers.postgres.hooks.postgres import PostgresHook
from airflow.providers.postgres.operators.postgres import PostgresOperator



@dag(
    dag_id="process_employees",
    schedule="0 0 * * *",  # daily at midnight
    start_date=pendulum.datetime(2021, 1, 1, tz="UTC"),
    catchup=False,
    dagrun_timeout=datetime.timedelta(minutes=60),
)
def ProcessEmployees():
    # Step 1: Create main employees table
    create_employees_table = PostgresOperator(
        task_id="create_employees_table",
        postgres_conn_id="tutorial_pg_conn",
        sql="""
            CREATE TABLE IF NOT EXISTS employees (
                "Serial Number" NUMERIC PRIMARY KEY,
                "Company Name" TEXT,
                "Employee Markme" TEXT,
                "Description" TEXT,
                "Leave" INTEGER
            );""",
    )

    # Step 2: Create staging table
    create_employees_temp_table = PostgresOperator(
        task_id="create_employees_temp_table",
        postgres_conn_id="tutorial_pg_conn",
```



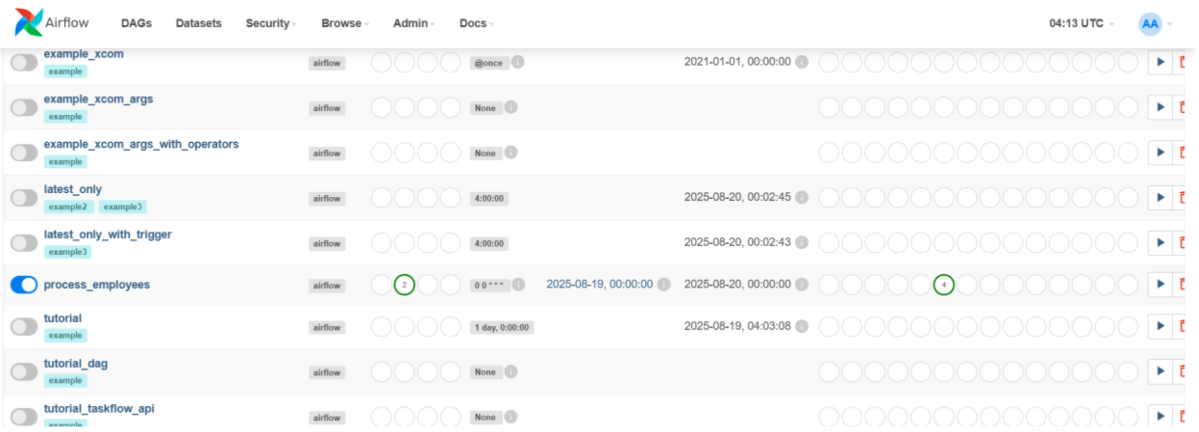
```
import datetime
import pendulum
import os
import requests

from airflow.decorators import dag, task
from airflow.providers.postgres.hooks.postgres import PostgresHook
from airflow.providers.postgres.operators.postgres import PostgresOperator


@dag(
    dag_id="process_employees",
    schedule="0 0 * * *",  # daily at midnight
    start_date=pendulum.datetime(2021, 1, 1, tz="UTC"),
    catchup=False,
    dagrun_timeout=datetime.timedelta(minutes=60),
)
```

```
airflow-scheduler-1  |  <TaskInstance: process_employees.merge_data scheduled__2025-08-18T00:00:00+00:00 [scheduled]>

airflow-scheduler-1  | [2025-08-19 06:23:59,695] {scheduler_job.py:537} INFO - Sending TaskInstanceKey(dag_id='process_employees', task_id='merge_data', ru
n_id='scheduled__2025-08-18T00:00:00+00:00', try_number=1, map_index=-1) to executor with priority 1 and queue default
airflow-scheduler-1  | [2025-08-19 06:23:59,695] {base_executor.py:95} INFO - Adding to queue: ['airflow', 'tasks', 'run', 'process_employees', 'merge_data
', 'scheduled__2025-08-18T00:00:00+00:00', '--local', '--subdir', 'DAGS_FOLDER/process_employees.py']
airflow-scheduler-1  | [2025-08-19 06:23:59,740] {scheduler_job.py:595} INFO - Executor reports execution of process_employees.merge_data run_id=scheduled_
_2025-08-18T00:00:00+00:00 exited with status queued for try_number 1
airflow-scheduler-1  | [2025-08-19 06:23:59,741] {scheduler_job.py:595} INFO - Executor reports execution of process_employees.get_data run_id=scheduled__2
025-08-18T00:00:00+00:00 exited with status success for try_number 1
airflow-worker-1     | [2025-08-19 06:23:59,742: INFO/ForkPoolWorker-15] Filling up the DagBag from /opt/airflow/dags/process_employees.py



airflow-scheduler-1  | [2025-08-19 06:23:59,748] {scheduler_job.py:652} INFO - TaskInstance Finished: dag_id=process_employees, task_id=get_data, run_id=sc
heduled__2025-08-18T00:00:00+00:00, map_index=-1, run_start_date=2025-08-19 06:23:58.333827+00:00, run_end_date=2025-08-19 06:23:59.550489+00:00, run_durat
```

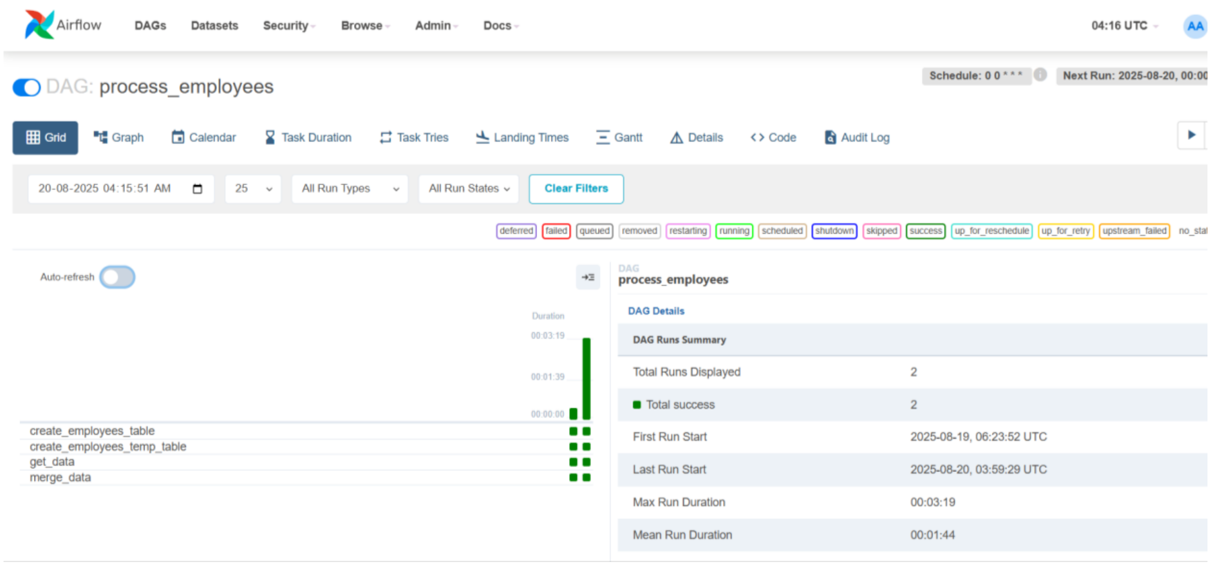
## Finding the DAG in the Apache UI



| | | | | | | |
|---|---|---|---|---|---|---|
| example_xcom | example | airflow | | @once | | 2021-01-01, 00:00:00 |
| example_xcom_args | example | airflow | | None | | |
| example_xcom_args_with_operators | example | airflow | | None | | |
| latest_only | example2 example3 | airflow | | 4:00:00 | | 2025-08-20, 00:02:45 |
| latest_only_with_trigger | example3 | airflow | | 4:00:00 | | 2025-08-20, 00:02:43 |
| process_employees | | airflow | 2 | 0 0 * * * | 2025-08-19, 00:00:00 | 2025-08-20, 00:00:00 |
| tutorial | example | airflow | | 1 day, 0:00:00 | | 2025-08-19, 04:03:08 |
| tutorial_dag | example | airflow | | None | | |
| tutorial_taskflow_api | example | airflow | | None | | |



| | | | | | | |
|---|---|---|---|---|---|---|
| example_xcom | example | airflow | | @once | | 2021-01-01, 00:00:00 |
| example_xcom_args | example | airflow | | None | | |
| example_xcom_args_with_operators | example | airflow | | None | | |
| latest_only | example2 example3 | airflow | | 4:00:00 | | 2025-08-20, 00:02:45 |
| latest_only_with_trigger | example3 | airflow | | 4:00:00 | | 2025-08-20, 00:02:43 |
| process_employees | | airflow | 2 | 0 0 * * * | 2025-08-19, 00:00:00 | 2025-08-20, 00:00:00 |
| tutorial | example | airflow | | 1 day, 0:00:00 | | 2025-08-19, 04:03:08 |
| tutorial_dag | example | airflow | | None | | |
| tutorial_taskflow_api | example | airflow | | None | | |



DAG: process_employees

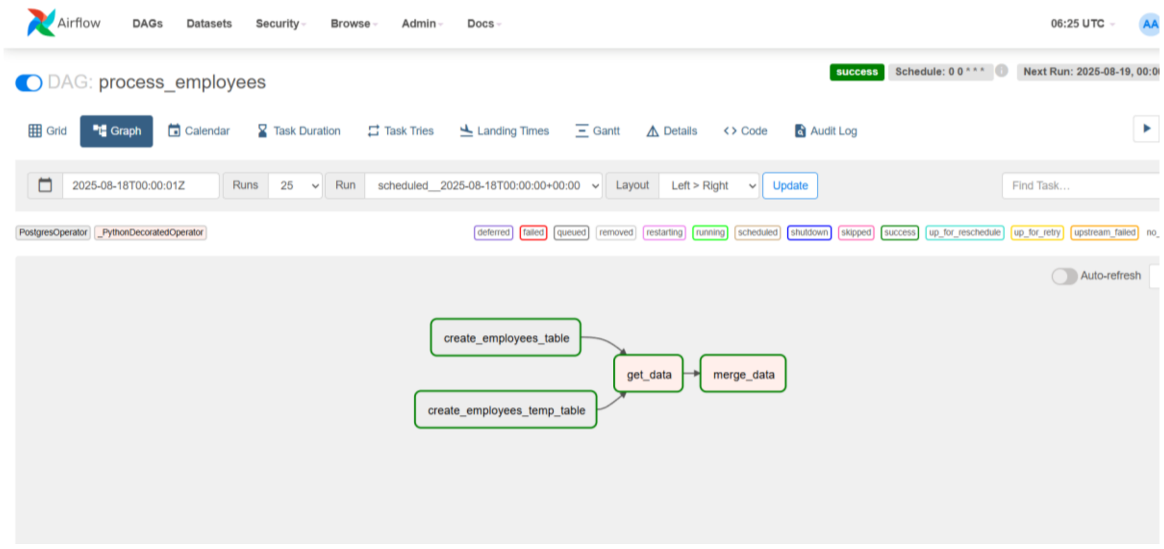
Schedule: 0 0 * * *  Next Run: 2025-08-20, 00:00

20-08-2025 04:15:51 AM | 25 | All Run Types | All Run States

create_employees_table
create_employees_temp_table
get_data
merge_data

DAG
process_employees

| DAG Runs Summary | |
|---|---|
| Total Runs Displayed | 2 |
| 🟩 Total success | 2 |
| First Run Start | 2025-08-19, 06:23:52 UTC |
| Last Run Start | 2025-08-20, 03:59:29 UTC |
| Max Run Duration | 00:03:19 |
| Mean Run Duration | 00:01:44 |



DAG: process_employees

success  Schedule: 0 0 * * *  Next Run: 2025-08-19, 00:00

2025-08-18T00:00:01Z | Runs 25 | Run scheduled__2025-08-18T00:00:00+00:00 | Layout Left > Right