

Name: Meghana G  
USN: 1RVU23CSE264

<b>Ex No: 6</b> <b>Date: 15/10/25</b>	<b>Apache Airflow</b>
--	-----------------------

## Objective:

Apache Airflow is an open-source platform created by Airbnb (now part of the Apache Software Foundation) for programmatically authoring, scheduling, and monitoring workflows. It is widely used in data engineering and ETL (Extract, Transform, Load) pipelines

## Outcomes:

1. Connecting to Airflow using Docker compose.
2. Installing Apache Airflow.
3. Running the DAGs(postgres) in Airflow.

## Materials:

- ENV files (.env)
- Csv file(users.csv)
- Yaml source file (docker-compose) and dockerfile.
- Python source file like dagselt\_pipeline\_postgres

## Lab Procedure:

### Stage 1: Install Apache Airflow for windows PC

1. Open the command prompt and navigate to your lab folder (cd "c:\Users\< >\lab6)
2. Install the docker and docker compose by commands :

```
sudo apt-get update
```

```
sudo apt-get install docker.io docker-compose -y
```

3. This builds a fresh image using your Dockerfile.

```
docker build -t airflowsqlserver -f Dockerfile --no-cache .
```

```
docker-compose up
```

Name: Meghana G  
USN: 1RVU23CSE264

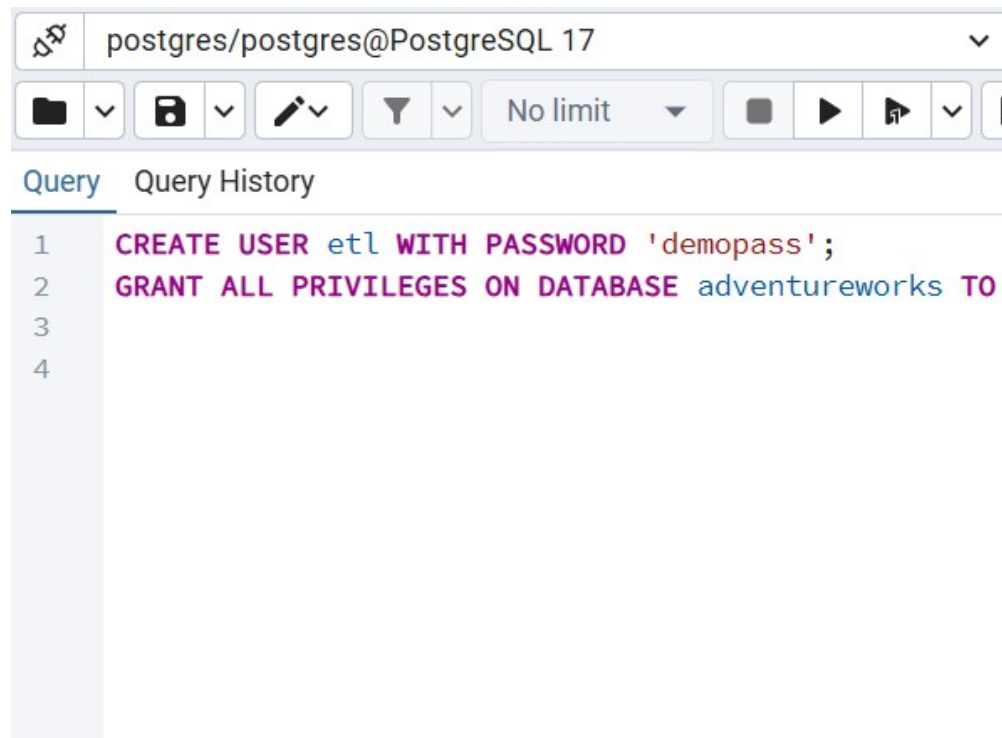
docker ps

- You should see containers for Airflow, Postgres, etc.
- Open a browser → Go to <http://localhost:8080> (usually Airflow UI).

4. Add this schema to pgadmin4 then save and run

```
CREATE SCHEMA IF NOT EXISTS etl_staging;
```

```
GRANT ALL PRIVILEGES ON SCHEMA etl_staging TO etl;
```



## Stage 2: Access to the Airflow

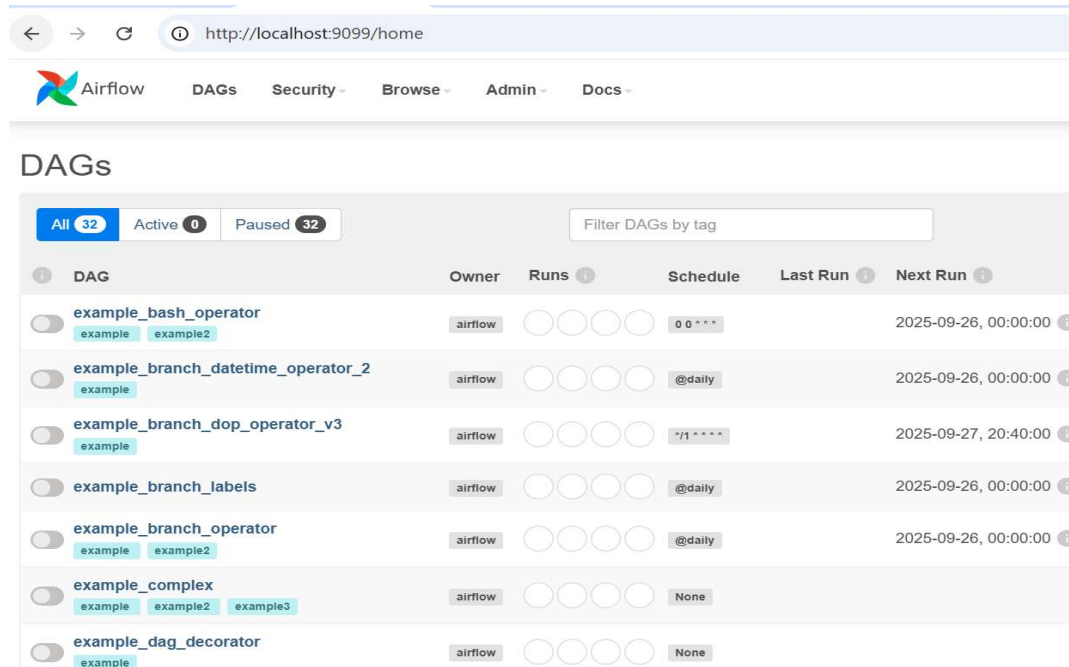
1. Open your browser:  
<http://localhost:8080>
2. Default Airflow login (unless changed in `.env`):

username: airflow

password: airflow

3. Verify the DAGs

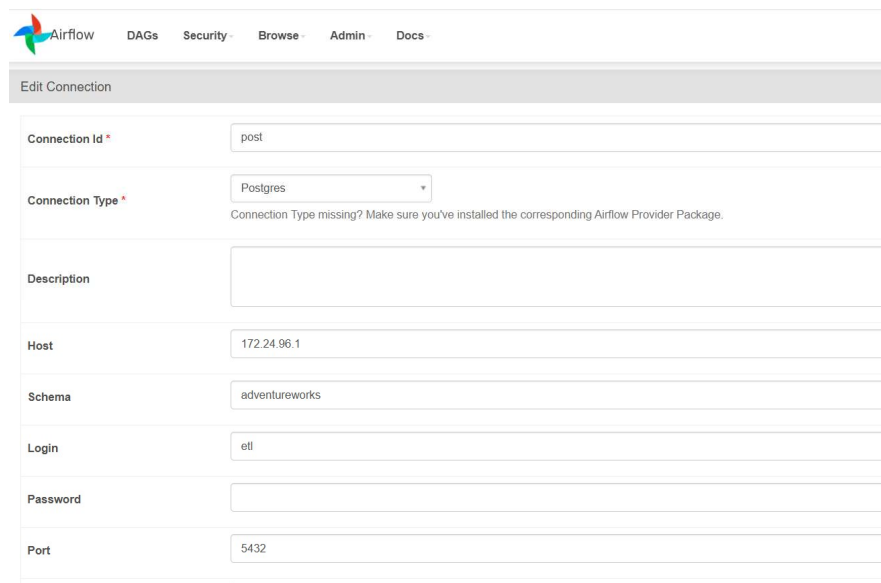
Name: Meghana G  
USN: 1RVU23CSE264



The screenshot shows the Airflow web interface at <http://localhost:9099/home>. The 'DAGs' tab is selected. At the top, there are filters for 'All' (32), 'Active' (0), and 'Paused' (32). A search bar 'Filter DAGs by tag' is also present. Below the filters is a table of DAGs:

DAG	Owner	Runs	Schedule	Last Run	Next Run
<input type="checkbox"/> example_bash_operator example example2	airflow	<div><div></div><div></div><div></div><div></div><div></div></div>	0 0 * * *		2025-09-26, 00:00:00
<input type="checkbox"/> example_branch_datetime_operator_2 example	airflow	<div><div></div><div></div><div></div><div></div><div></div></div>	@daily		2025-09-26, 00:00:00
<input type="checkbox"/> example_branch_dop_operator_v3 example	airflow	<div><div></div><div></div><div></div><div></div><div></div></div>	* * * * *		2025-09-27, 20:40:00
<input type="checkbox"/> example_branch_labels example	airflow	<div><div></div><div></div><div></div><div></div><div></div></div>	@daily		2025-09-26, 00:00:00
<input type="checkbox"/> example_branch_operator example example2	airflow	<div><div></div><div></div><div></div><div></div><div></div></div>	@daily		2025-09-26, 00:00:00
<input type="checkbox"/> example_complex example example2 example3	airflow	<div><div></div><div></div><div></div><div></div><div></div></div>	None		
<input type="checkbox"/> example_dag_decorator example	airflow	<div><div></div><div></div><div></div><div></div><div></div></div>	None		

- In the Airflow UI, go to the **DAGs** tab.
  - You should see your pipeline DAG from `dagselt_pipeline_postgres.py`.
  - Turn the toggle **ON** and trigger it.
4. Go to Edit connection and give host as your pc ip address , login and password save it

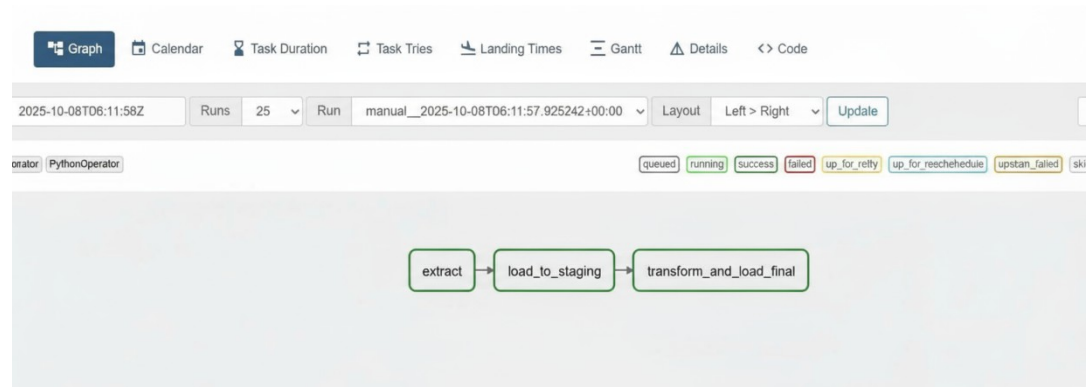


The screenshot shows the 'Edit Connection' form in the Airflow UI. The form has the following fields:

- Connection Id: `post`
- Connection Type: `Postgres` (dropdown menu)
- Description: (empty text area)
- Host: `172.24.96.1`
- Schema: `adventureworks`
- Login: `etl`
- Password: (empty password field)
- Port: `5432`

5. Run the Dag

Name: Meghana G  
USN: 1RVU23CSE264



It shows all green which means all stages run successfully.

Github link : <https://github.com/meghana1653/Data-Engineering/blob/main/Lab-6.zip>