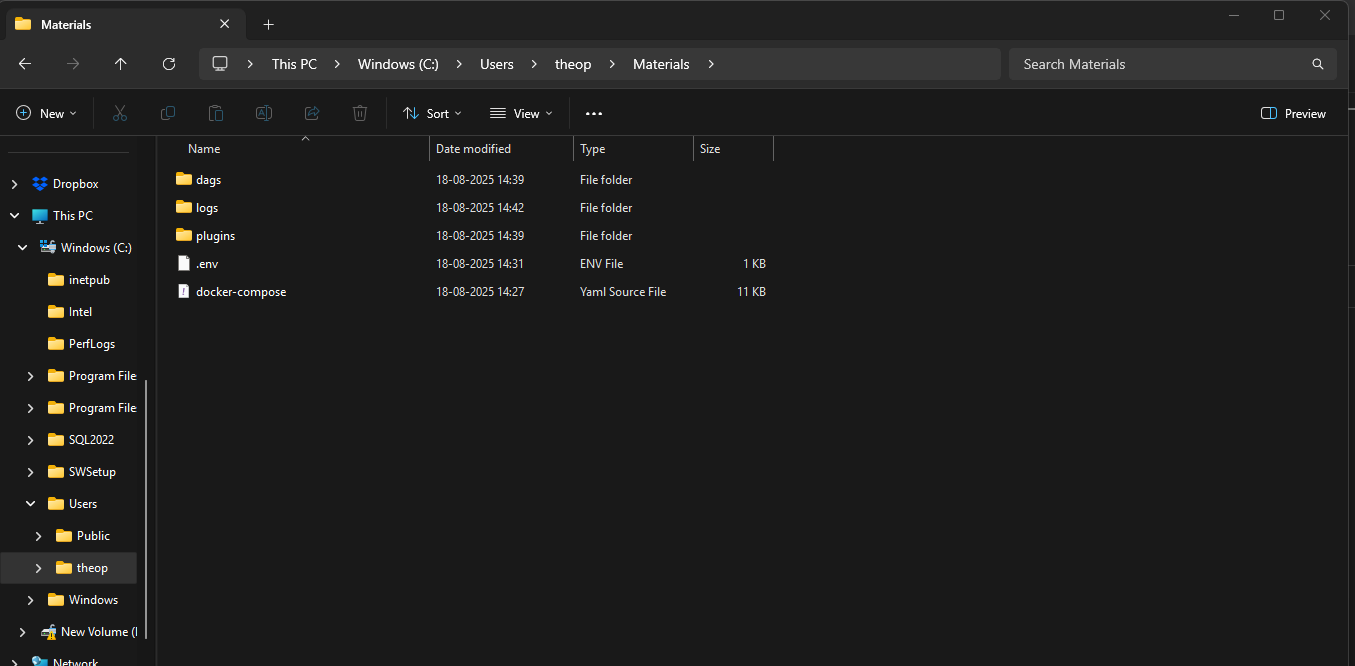
APACHE AIRFLOW SETUP

Theophila Murphy S

DE-Batch 6

**Step 1: Open your project folder**

* Go to the folder where your docker-compose.yaml will be (e.g. airflow-docker/).
* You can open it in Visual Studio Code or any text editor.

****

**Step 2: Create a new file**

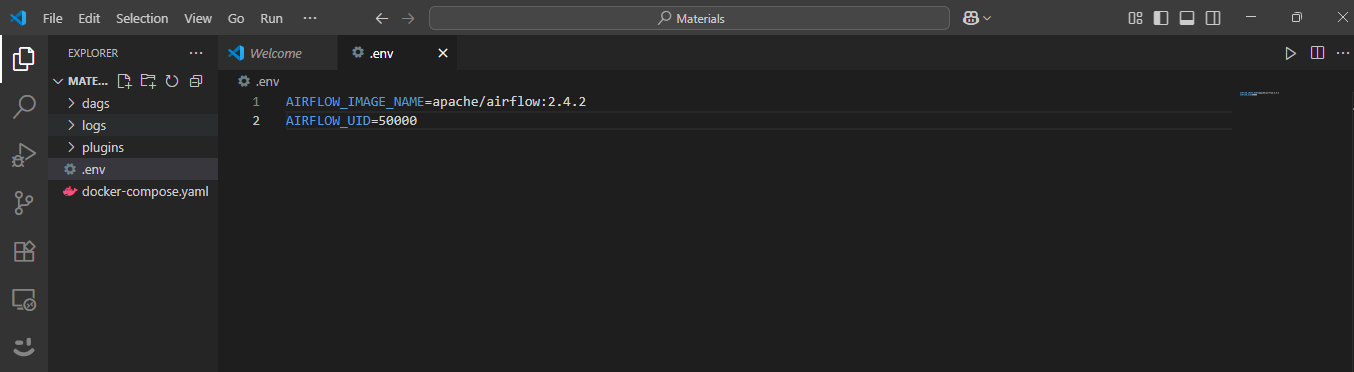
* In VS Code:
  + Right-click → New File → name it: **.env**
  + Open .env and paste this:

**AIRFLOW\_IMAGE\_NAME=apache/airflow:2.4.2**

**AIRFLOW\_UID=50000**

**Step 3: Save the file**

* Press **Ctrl+S** in VS Code (or File → Save).
* Ensure the file is in the **same folder as docker-compose.yaml**.

****

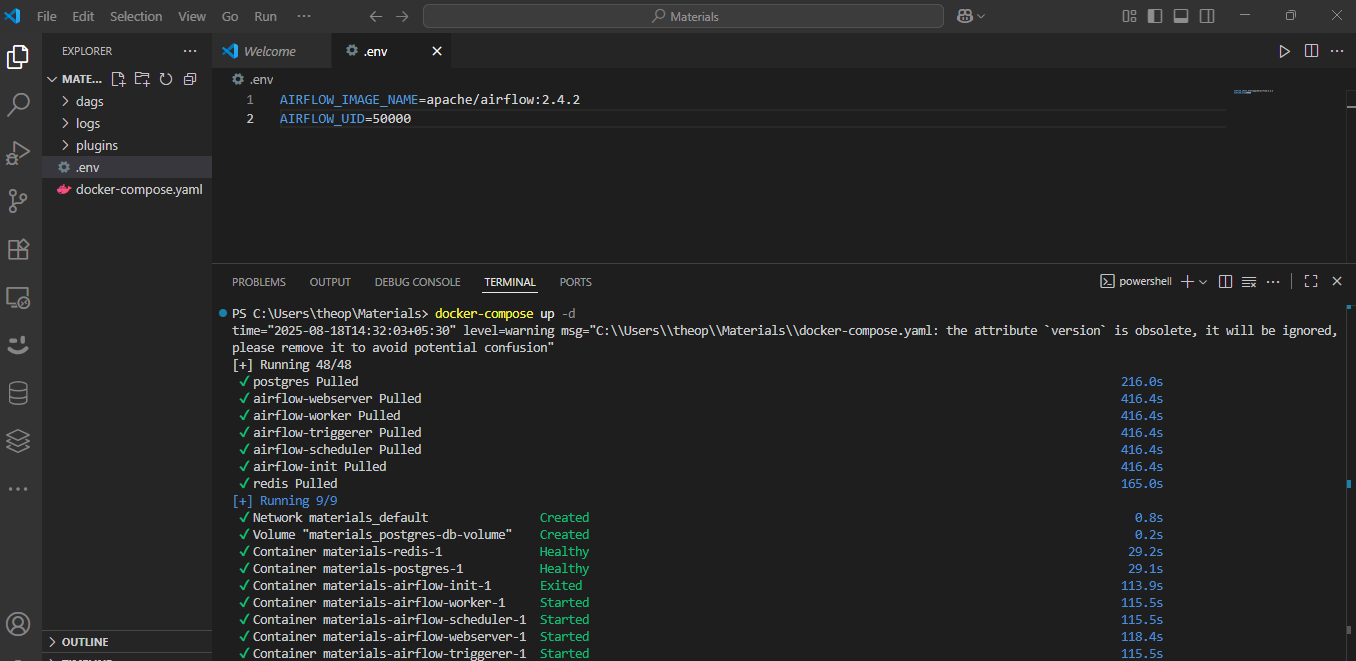
**Step 4: Start Airflow Services**

Bring up the entire Airflow stack (webserver, scheduler, worker, triggerer, redis, postgres).

🔹 What happens here:

* -d means **detached mode** (runs in background)
* Starts these containers:
  + **postgres** → stores metadata
  + **redis** → message broker (Celery needs this)
  + **airflow-webserver** → web UI (port 8080)
  + **airflow-scheduler** → schedules tasks
  + **airflow-worker** → executes tasks
  + **airflow-triggerer** → manages async triggers

✅ After this, Airflow is “running.”

****

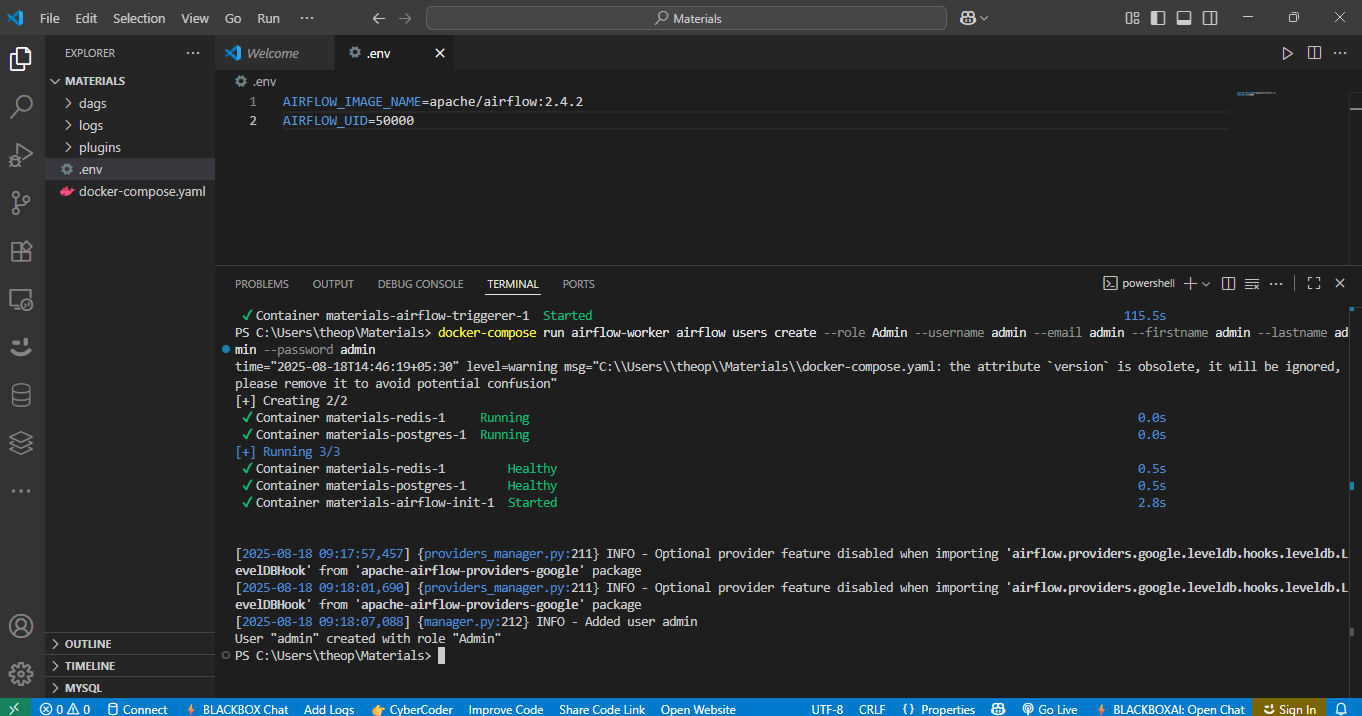
**Step 5: Create Admin User**

🔹 Explanation:

* docker compose run → runs a command inside the airflow-worker container
* --rm → remove container after it finishes
* airflow users create → Airflow CLI command
* --role Admin → gives full access in UI
* Username/password: admin/admin (you can change later)

✅ You’ll get a success message:

User admin created with role Admin

****

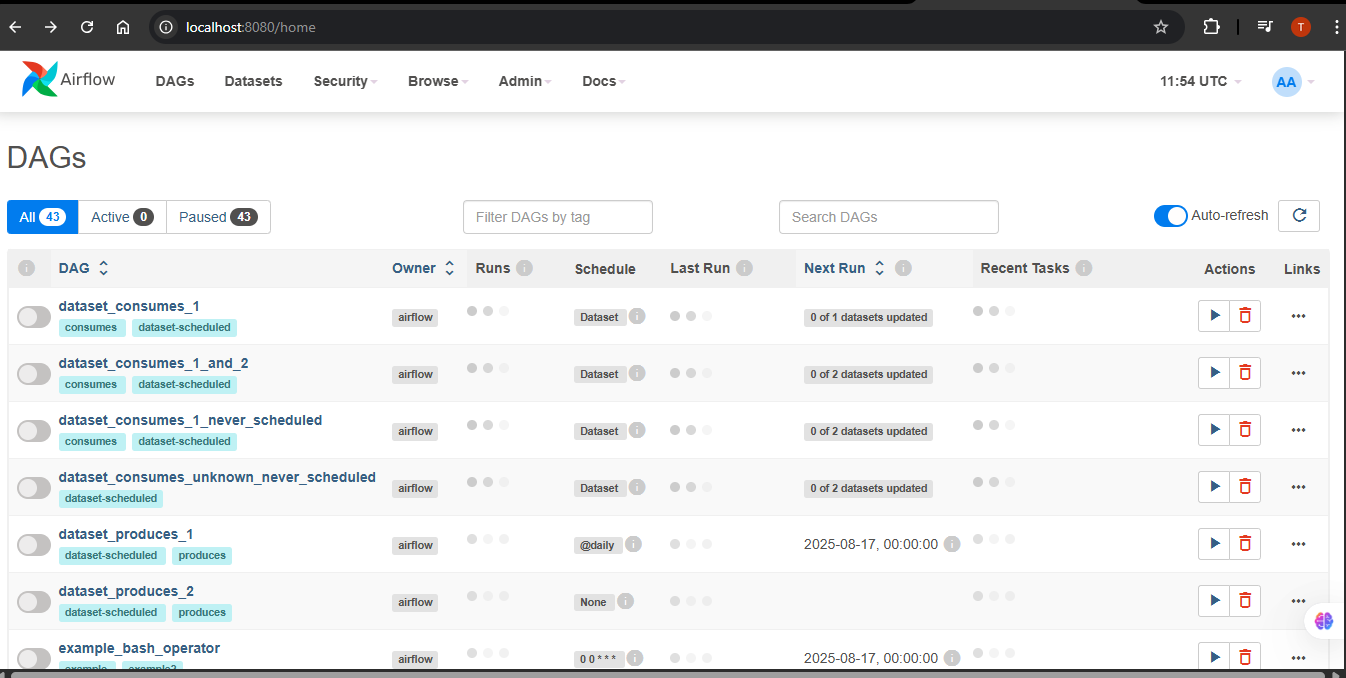
**Step 6: Open Airflow Web UI**

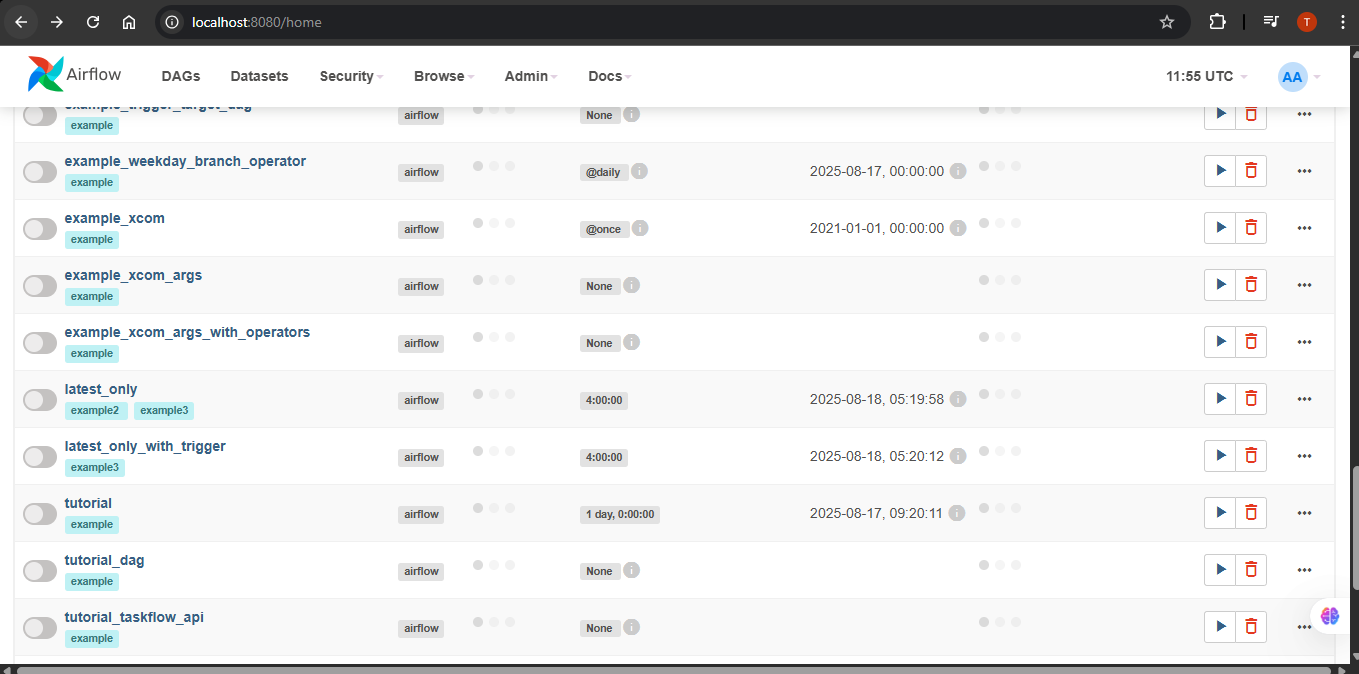
Open your browser and go to:

👉 <http://localhost:8080>

* Username: admin
* Password: admin

✅ You’ll see the Airflow dashboard.

****

****