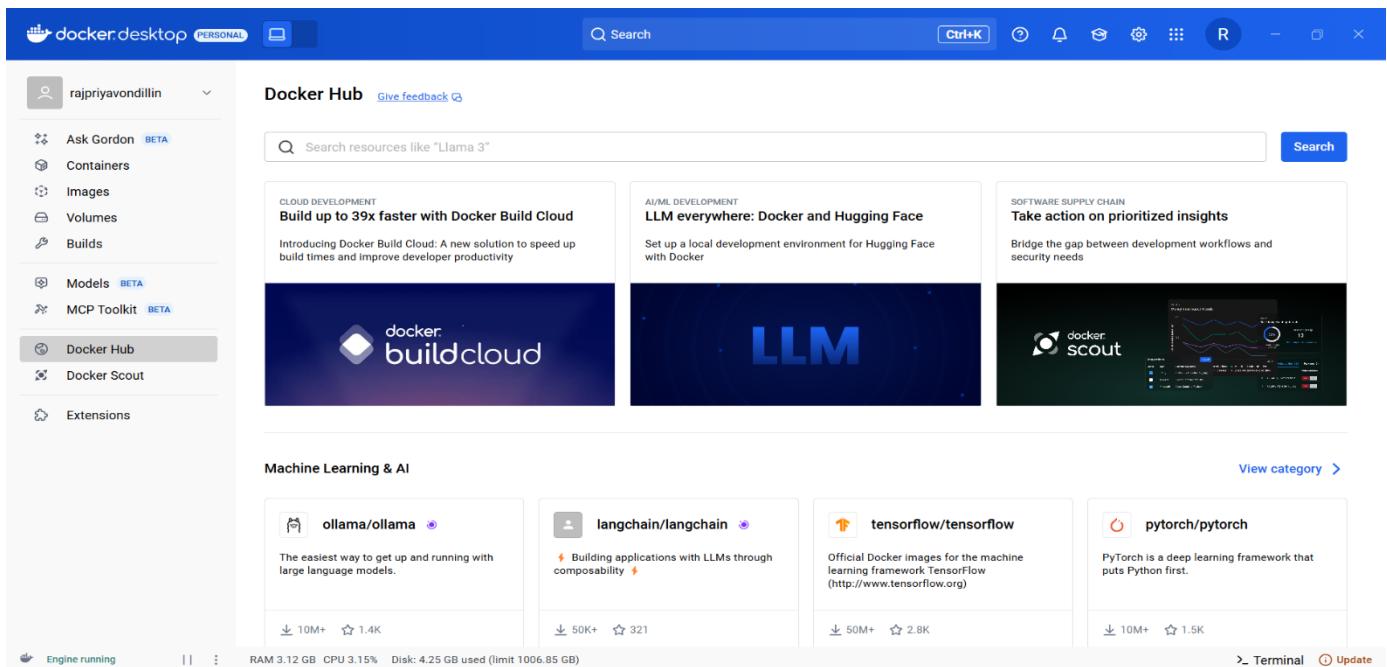


Install Docker - <https://youtu.be/JBEUKrbWqg?si=Xv4k-04gUxFFF84H> (Reference link)



CREATING DOCKER HUB ACCOUNT

- Create a docker hub account using your email Id (It is just like signing up)
- Then restart your PC (optional – but for safer side)
- Once done, you have now created the docker hub account

Creating the Workspace Manually with the help of yaml file,

- I have attached the “docker-compose.yaml” file
 - Download it in your local folder
 - Now at any path in your file manager, create a folder called “airflow-docker”
 - **Note down the path**
 - Inside the “airflow-docker” folder, put your “docker-compose.yaml” file
 - In the same “airflow-docker” folder, create 3 subfolders say,
 - dags
 - logs
 - plugins
 - Now open your cmd (Command prompt) and type the following,
 - **Open CMD (not PyCharm terminal for this step, unless you've mounted Docker inside PyCharm).**
- Navigate into your folder: (NOTE: Paste your noted “airflow-docker” folder path)

```
cd C:\Users\rajpr\OneDrive\Documents\HEXAWARE\airflow-docker
```

- Run docker-compose to start Airflow in the background:

```
docker-compose up -d
```

Wait a few minutes (first time it will pull several Docker images, so it may take a while).

- Once running, check containers:

```
docker ps
```

- You should see services like airflow-webserver, airflow-scheduler, airflow-worker, etc.
- You must see your cmd something like this,

```
Command Prompt
Microsoft Windows [Version 10.0.26100.4904]
(c) Microsoft Corporation. All rights reserved.

C:\Users\rajpr\OneDrive\Documents\HEXAWARE\airflow-docker>docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
31166dd52f70        apache/apache:2.9.1   "/usr/bin/dumb-init --"   About a minute ago   Up 46 seconds (healthy)   8088/tcp            airflow-docker-airflow-triggerer-1
95d9cfc07cda        apache/apache:2.9.1   "/usr/bin/dumb-init --"   About a minute ago   Up 46 seconds (health: starting)  8088/tcp            airflow-docker-airflow-scheduler-1
23bf28d43694        apache/apache:2.9.1   "/usr/bin/dumb-init --"   About a minute ago   Up 45 seconds (health: starting)  0.0.0.0:8080->8080/tcp, [::]:8080->8080/tcp  airflow-docker-airflow-webserver-1
6b03eada82cb        apache/apache:2.9.1   "/usr/bin/dumb-init --"   About a minute ago   Up 46 seconds (health: starting)  8088/tcp            airflow-docker-airflow-worker-1
56a643ba89bd        postgres:13            "docker-entrypoint.s--" About a minute ago   Up About a minute (healthy)    5432/tcp            airflow-docker-postgres-1
6ccceded4d801       redis:7.2-bookworm   "docker-entrypoint.s--" About a minute ago   Up About a minute (healthy)    6379/tcp            airflow-docker-redis-1

C:\Users\rajpr\OneDrive\Documents\HEXAWARE\airflow-docker>docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
31166dd52f70        apache/apache:2.9.1   "/usr/bin/dumb-init --"   About a minute ago   Up 46 seconds (healthy)   8088/tcp            airflow-docker-airflow-triggerer-1
95d9cfc07cda        apache/apache:2.9.1   "/usr/bin/dumb-init --"   About a minute ago   Up 46 seconds (health: starting)  8088/tcp            airflow-docker-airflow-scheduler-1
23bf28d43694        apache/apache:2.9.1   "/usr/bin/dumb-init --"   About a minute ago   Up 45 seconds (health: starting)  0.0.0.0:8080->8080/tcp, [::]:8080->8080/tcp  airflow-docker-airflow-webserver-1
6b03eada82cb        apache/apache:2.9.1   "/usr/bin/dumb-init --"   About a minute ago   Up 46 seconds (health: starting)  8088/tcp            airflow-docker-airflow-worker-1
56a643ba89bd        postgres:13            "docker-entrypoint.s--" About a minute ago   Up About a minute (healthy)    5432/tcp            airflow-docker-postgres-1
6ccceded4d801       redis:7.2-bookworm   "docker-entrypoint.s--" About a minute ago   Up About a minute (healthy)    6379/tcp            airflow-docker-redis-1

C:\Users\rajpr\OneDrive\Documents\HEXAWARE\airflow-docker>
```

- Open your browser and go to: https://localhost:8080/#google_vignette (Airflow UI will be available there).
- Now click on to the **Go to localhost: 8080**, a green colour button
- Refresh and wait for sometimes it will surely open
- Once done, type the username and password,
- By default, in the **official Airflow Docker Compose setup**, the login is:
Username: airflow
Password: airflow
First try with airflow / airflow.
- By doing the above steps, 3 steps have been completed
 - Step 1: Install Airflow Locally (Using Virtualenv)
 - Create and activate a virtual environment
 - Install Airflow (with local executor + SQLite for dev)
 - Step 2: Initialize Airflow
 - Step 3: Start Airflow UI and Scheduler

TO PERFORM STEP 4, WHICH IS TO CREATE A SIMPLE DAG

- Open PyCharm and Navigate to File -> Open “airflow-docker” folder
- In the left side of contents in PyCharm, you will find all the subfolders inside this “airflow-docker” folder
- Set up your python interpreter
- Inside the dags subfolder in PyCharm Right Click -> new -> python file -> name it as “hello_dag.py”
- And enter this sample code,

```

● from airflow import DAG
from airflow.operators.python import PythonOperator
from datetime import datetime

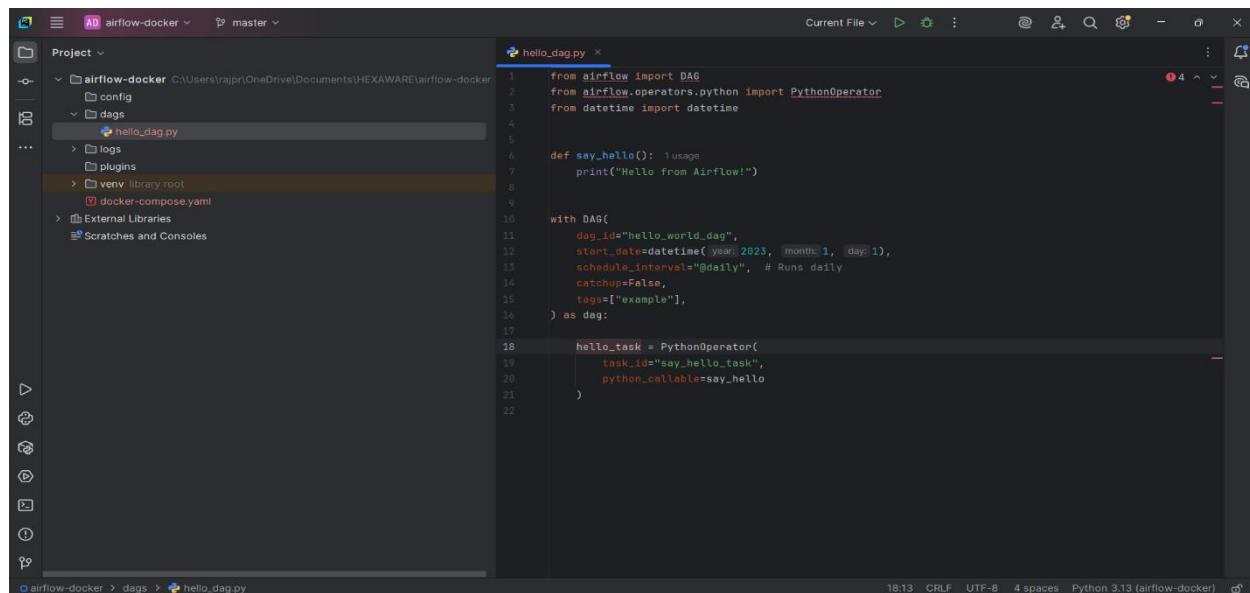
def say_hello():
    print("Hello from Airflow!")

with DAG(
    dag_id="hello_world_dag",
    start_date=datetime(2023, 1, 1),
    schedule_interval="@daily", # Runs daily
    catchup=False,
    tags=["example"],
) as dag:

    hello_task = PythonOperator(
        task_id="say_hello_task",
        python_callable=say_hello
    )

```

- Finally, you will see something like this



The screenshot shows the PyCharm interface with the 'airflow-docker' project open. The 'Project' tool window on the left displays the directory structure: 'airflow-docker' containing 'config', 'dags' (which contains 'hello_dag.py'), 'logs', 'plugins', 'venv library root', and 'docker-compose.yaml'. The 'External Libraries' and 'Scratches and Consoles' sections are also visible. The 'Current File' tab at the top shows 'hello_dag.py' is the active file. The code editor window on the right contains the Python code for the DAG, which prints "Hello from Airflow!" and runs daily.

```

1  from airflow import DAG
2  from airflow.operators.python import PythonOperator
3  from datetime import datetime
4
5
6  def say_hello():
7      print("Hello from Airflow!")
8
9
10 with DAG(
11     dag_id="hello_world_dag",
12     start_date=datetime(year=2023, month=1, day=1),
13     schedule_interval="@daily", # Runs daily
14     catchup=False,
15     tags=["example"],
16 ) as dag:
17
18     hello_task = PythonOperator(
19         task_id="say_hello_task",
20         python_callable=say_hello
21     )
22

```

IMPORTANT: WHEN USING DOCKER, YOU DON'T RUN DAGS LOCALLY. AIRFLOW INSIDE THE CONTAINER RUNS THEM AUTOMATICALLY.

- Go to your browser → https://localhost:8080/#google_vignette
- Refresh the UI → you will see “hello_world_dag”
- Click **Trigger DAG**  Airflow will execute the DAG inside the container.

Key takeaway:

- **Do not run DAG Python files directly in PyCharm** if you’re using Docker.
- PyCharm is only for **editing** the DAG code.
- Dockerized Airflow is the one that **executes** the DAGs.

Here’s what happens when you Trigger DAG and what the output looks like:

1. DAG Status in Airflow UI

- In the **DAGs** list, the hello_world_dag row will change:
 - Status will show **running** (green arrow spinning).
 - Once finished, it will turn **green** → indicates successful execution.

2. Task Instance Status

- Click on the DAG name → you’ll see the **graph view / tree view**.
- Your single task say_hello_task will appear as a **node**.
- Node colours indicate status:
 - **Green** → success
 - **Yellow** → running
 - **Red** → failed

3. Logs / Output

- Click on the task → **View Log**.
- You will see something like:

[2025-08-18 12:35:00,123] {python.py:123} INFO - Executing <PythonOperator: say_hello_task> on ...

[2025-08-18 12:35:00,125] {python.py:126} INFO - Hello from Airflow!

[2025-08-18 12:35:00,130] {taskinstance.py:1450} INFO - Marking task as SUCCESS

- The line Hello from Airflow! is exactly from your print() in the DAG. 

4. Summary

- DAG triggered → task runs inside Docker container → output goes to **Airflow logs**.
- You **don’t see the output in PyCharm** because the Python process runs inside Docker.

localhost:8080/home?status=active

12:06 UTC AA

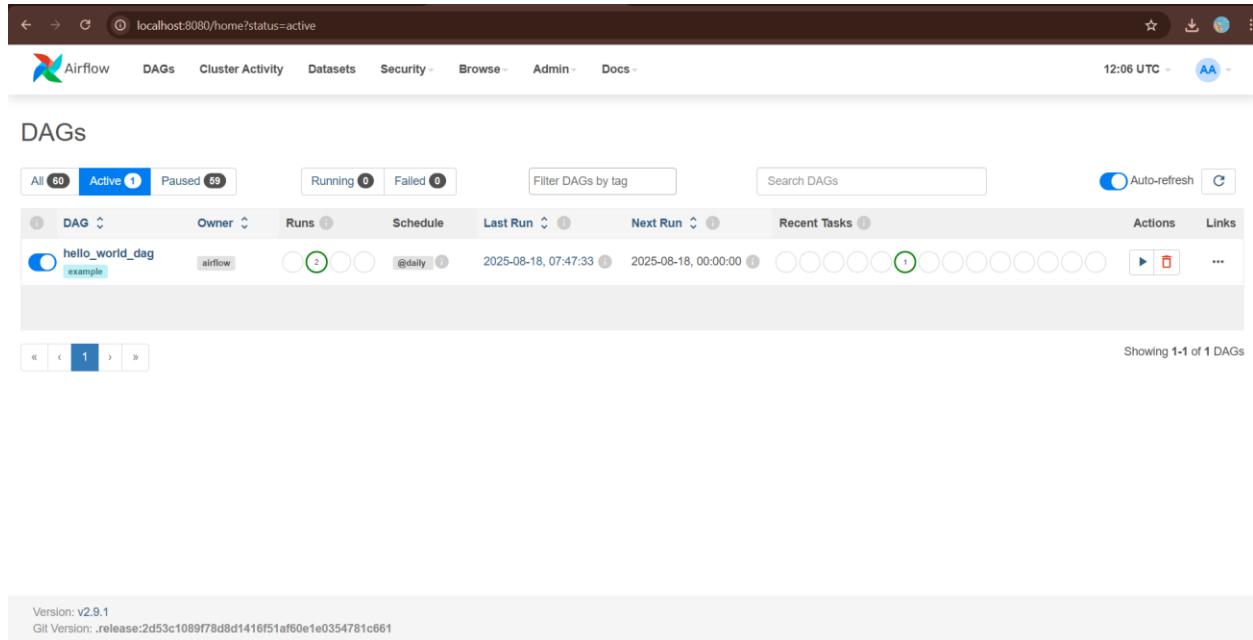
DAGs

All 60 Active 1 Paused 59 Running 0 Failed 0 Filter DAGs by tag Search DAGs Auto-refresh C

DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks	Actions	Links
hello_world_dag	airflow	2	@daily	2025-08-18, 07:47:33	2025-08-18, 00:00:00	1	...	

Showing 1-1 of 1 DAGs

Version: v2.9.1
Git Version: .release:2d53c1089f78d8d1416f51af60e1e0354781c661



12:29 UTC AU

DAG: hello_world_dag

18-08-2025 00:00 Clear Filters Auto-refresh 25

Schedule: @daily Next Run ID: 2025-08-17, 00:00:00 UTC

Press shift + / for Shortcuts

Duration: 00:00:05

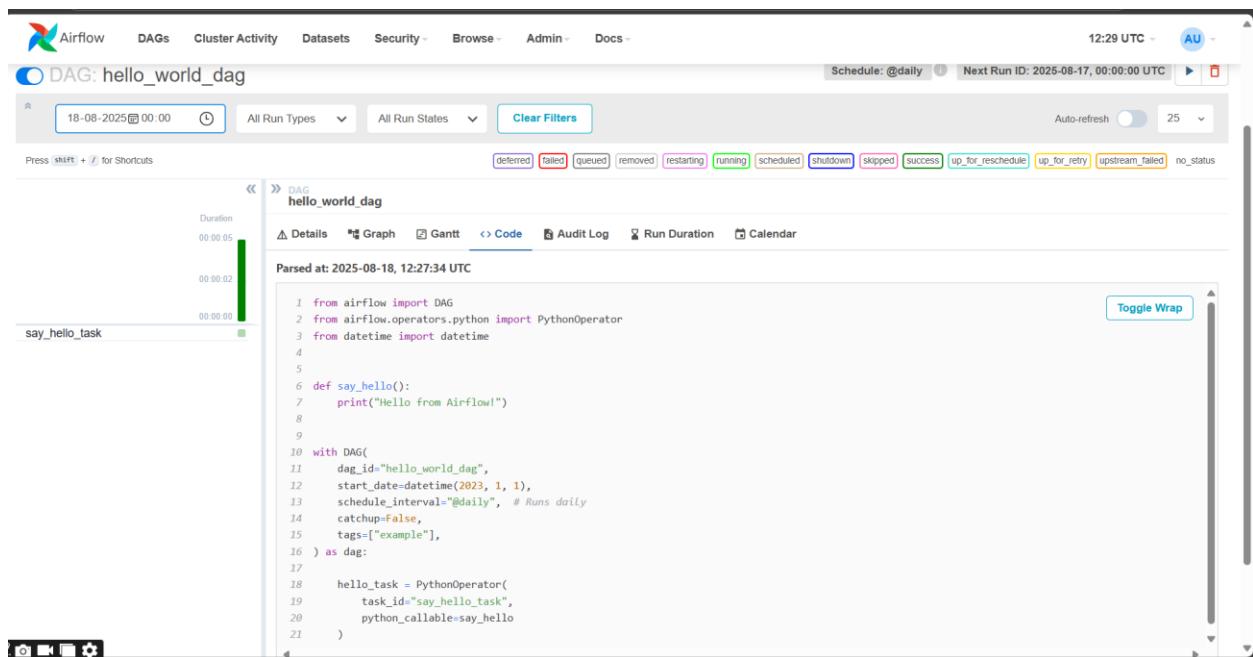
say_hello_task

Deferred Failed Queued Removed Restarting Running Scheduled Shutdown Skipped Success Up_for_reschedule Up_for_retry Upstream_failed No_status

Details Graph Gantt Audit Log Run Duration Calendar

Parsed at: 2025-08-18, 12:27:34 UTC

```
1 from airflow import DAG
2 from airflow.operators.python import PythonOperator
3 from datetime import datetime
4
5
6 def say_hello():
7     print("Hello from Airflow!")
8
9
10 with DAG(
11     dag_id="hello_world_dag",
12     start_date=datetime(2023, 1, 1),
13     schedule_interval="@daily", # Runs daily
14     catchup=False,
15     tags=["example"],
16 ) as dag:
17
18     hello_task = PythonOperator(
19         task_id="say_hello_task",
20         python_callable=say_hello
21     )
```



Airflow DAG: hello_world_dag

Schedule: @daily | Next Run ID: 2025-08-18, 00:00:00 UTC | Trigger DAG

18-08-2025 12:06:52 | All Run Types | All Run States | Clear Filters | Auto-refresh 25

Press shift + / for Shortcuts

Duration: 00:00:23

DAG hello_world_dag

Details Graph Gantt Code Audit Log Run Duration Calendar

DAG Runs Summary

Total Runs Displayed	2
Total success	2
First Run Start	2025-08-18, 07:47:34 UTC
Last Run Start	2025-08-18, 07:47:34 UTC
Max Run Duration	00:00:23
Mean Run Duration	00:00:23
Min Run Duration	00:00:23

DAG Summary

Total Tasks: 1

localhost:8080/dags/hello_world_dag/trigger?origin=%2Fdags%2FHello_world_dag%2Fgrid

Triggered hello_world_dag, it should start any moment now.

DAG: hello_world_dag

Schedule: @daily | Next Run ID: 2025-08-17, 00:00:00 UTC | Trigger DAG

18-08-2025 00:00 | All Run Types | All Run States | Clear Filters | Auto-refresh 25

Press shift + / for Shortcuts

Duration: 00:00:05

DAG hello_world_dag

Details Graph Gantt Code Audit Log Run Duration Calendar

% Success

