

Steps Followed to Setup Apache Airflow with Docker

Below are the steps that were followed to set up Apache Airflow using Docker:

1. Installed Docker

Docker Desktop was installed on the system to allow containerized applications to run. This is a prerequisite for running Apache Airflow with Docker Compose.

2. Created Project Directory

A new directory was created for setting up Apache Airflow (e.g., airflow-docker). Inside this directory, all required files for Airflow were kept.

3. Downloaded Official Airflow Docker Compose Setup

The official Apache Airflow Docker Compose file was downloaded using the following command:

```
curl -LfO 'https://airflow.apache.org/docs/apache-airflow/2.9.1/docker-compose.yaml'
```

4. Set Environment Variable for Airflow UID

To ensure proper permissions, the following environment variable was set:

```
$env:AIRFLOW_UID=50000 (for PowerShell)
```

or

```
export AIRFLOW_UID=50000 (for Linux/macOS)
```

5. Initialized Airflow Database

The Airflow database was initialized using:

```
docker-compose up airflow-init
```

This step prepares the metadata database for Airflow.

6. Started Airflow Services

All Airflow services (webserver, scheduler, worker, etc.) were started with:

```
docker-compose up
```

or in detached mode:

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

7. Accessed Airflow Web UI

The Airflow web interface was accessed at <http://localhost:8080>.

Default credentials: Username = airflow, Password = airflow.

8. Verified DAGs

The Airflow UI showed the default example DAGs in a paused state. These DAGs can be enabled and triggered for testing.

Conclusion

So far, Docker has been successfully set up, Airflow services have been started, and the web UI is accessible with default DAGs visible. The next steps would include creating custom DAGs for workflows.