USN:1RVU23CSE325
NAME: PANCHAMI DINESH

Ex No: 6 **Date:15-10-25**	**APACHE AIRFLOW**

Objective:

- To understand Apache Airflow as a platform for authoring, scheduling, and monitoring workflows.
- To explore key components of Airflow including DAGs, operators, executors, and the metadata database.
- To install and configure Apache Airflow in a Windows environment.
- To create and run workflows/programs defined as DAGs in Airflow for ETL pipelines.

Outcomes:

1. Knowledge of Apache Airflow architecture and how workflows are represented as DAGs.
2. Ability to define tasks using different operators like BashOperator, PythonOperator, Transfer Operators, and Sensor Operators.
3. Understanding of execution models via Executors such as Sequential, Local, Celery, and Kubernetes.
4. Skills to monitor workflows using the Airflow web dashboard.
5. Hands-on experience installing Airflow with Docker on Windows.
6. Capability to create, schedule, and execute DAGs with dependency management.

Materials:

- dagselt_pipeline_postgres - python source file
- docker-compose - yaml source file
- Users.csv
- .env - ENV file
- Dockerfile

Lab Procedure:

Stage 1: Installation and Setup

1. Install **Apache Airflow** using **Docker**.

USN:1RVU23CSE325
NAME: PANCHAMI DINESH

2. Run the following commands to build and start the setup:
   - `docker build -t airflowsqlserver -f Dockerfile --no-cache .`
   - `docker-compose up`
3. Once Airflow is running, open the **Airflow Web UI** in your browser at http://localhost:9099/home.

Stage 2: Understanding Airflow Components

1. **DAG (Directed Acyclic Graph):** Defines the workflow structure as a sequence of dependent tasks.
2. **Task:** Represents a single unit of work, executed by an operator.
3. **Workers:** Responsible for running the actual tasks or scripts.
4. **Web Server (UI):** Provides a graphical dashboard to monitor DAGs, task runs, and logs.
5. **Metadata Database:** Stores details about DAG runs, task states, and configurations.

Stage 3: Creating and Running a Sample DAG

1. Open the **Airflow DAGs** folder and create a new Python file, for example, `etl_dag.py`.
2. Define the DAG with tasks such as **extract**, **transform**, and **load**.
3. Set task dependencies using:
   - `task1 >> task2 >> task3`
4. Save and deploy the DAG.

USN:1RVU23CSE325
NAME: PANCHAMI DINESH

5. Monitor the DAG execution and task progress in the **Airflow Web UI**.

Stage 4: Database Schema Setup

1. Connect to the **Airflow-linked database**.
2. Execute the following SQL commands:
   - `CREATE SCHEMA IF NOT EXISTS etl_staging;`
   - `GRANT ALL PRIVILEGES ON SCHEMA etl_staging TO etl;`
3. These commands create the ETL staging schema (if not already present) and grant necessary privileges to the ETL user.

GitHub Link: https://github.com/panchamidinesh/FDE6