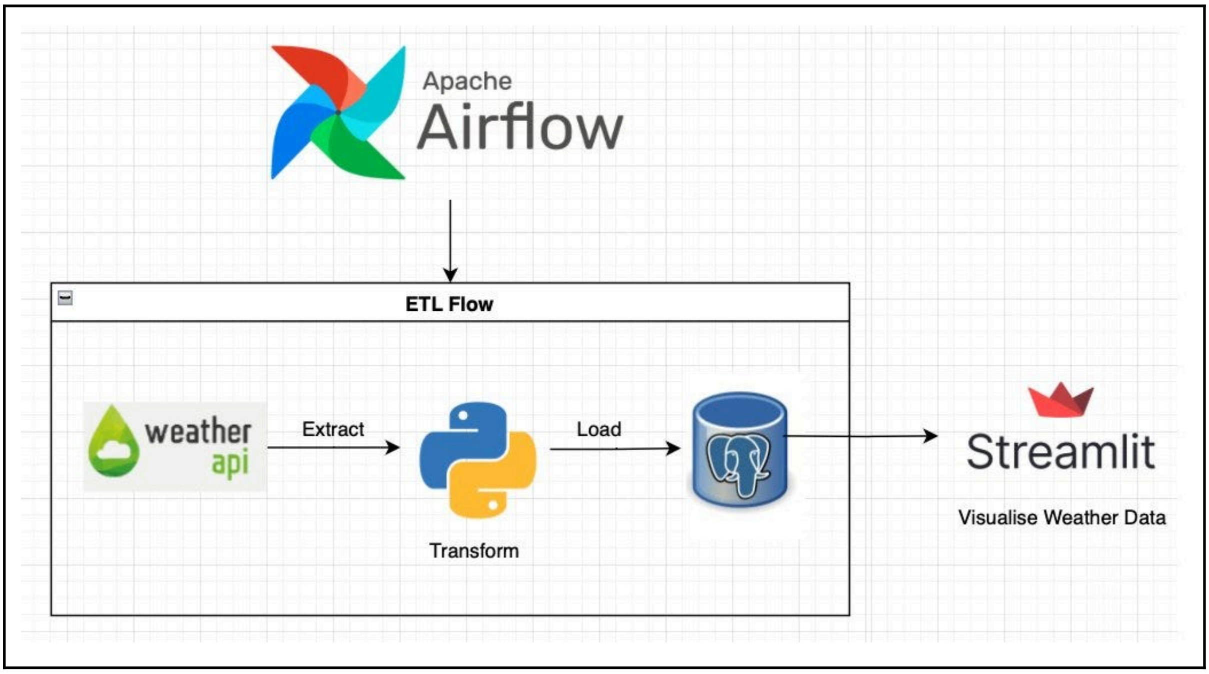
Apache Airflow

Apache Airflow is an open-source platform used to programmatically author, schedule, and monitor workflows. It allows you to define complex workflows as code and manage their execution. Airflow is commonly used for data pipelines, where tasks like data extraction, transformation, and loading (ETL) are orchestrated across multiple systems.



Why Airflow For MLOps ?

In MLOps (Machine Learning Operations), orchestrating ML workflows efficiently is crucial for ensuring that data pipelines, model training, and deployment tasks happen smoothly and in an automated manner. Airflow is well-suited for this purpose because it allows you to define, automate, and monitor every step in an ML pipeline.

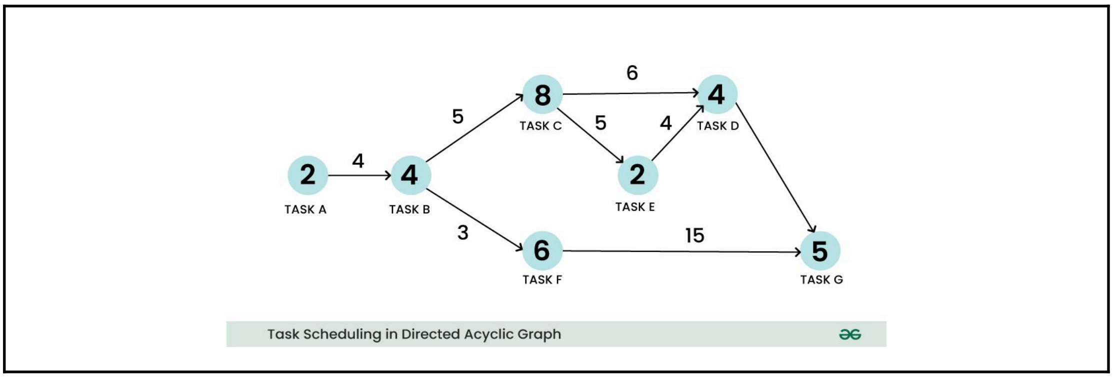
I. Orchestrating ML and ETL Pipelines : Defines end-to-end workflows (ETL → feature engineering → training → evaluation → deployment) as DAGs, ensuring correct order, dependencies, and reproducibility.

II. Task Automation : Automates recurring ML tasks such as data ingestion, retraining models, batch inference, and model validation on schedules or triggers.

III. Monitoring & Alerts : Provides built-in (ui) for monitoring logs, retries, and alerting (email/slack) in real-time to quickly detect and handle pipeline failures.

Key Concepts In Apache Airflow

1. DAG (Directed Acyclic Graph) : It is the collection of tasks that we want to schedule and run. It is also a directed graph which gives some direction for this particular graph.



Task Scheduling in Directed Acyclic Graph

This graph shows the flow in which the task needs to be executed. There are two main properties of this graph:

I. Directed : Tasks must have a specific sequence.
II. Acyclic : A task shouldn't be dependent on itself.

2. Tasks : It represents the individual unit of work in a DAG. These tasks can be python functions, querying a database, sending HTTP requests, etc.

3. Dependecies : Tasks in a DAG have dependencies, meaning one task might need to finish before another task can start. These dependencies allow you to control the order in which tasks are executed. Airflow provides mechanisms like set_upstream and set_downstream to define these dependencies between tasks.