APACHE AIRFLOW

Apache Airflow, or Airflow, is an open-source tool and framework for running your data pipelines in production. As an industry-leading data workflow management tool, Apache Airflow leverages Python to allow data practitioners to define their data pipelines as code. Airflow adds the ability to schedule pipeline execution and observe performance, making it a centralized hub for all of your data workflows.

FEATURES OF AIRFLOW'S FRAMEWORK:

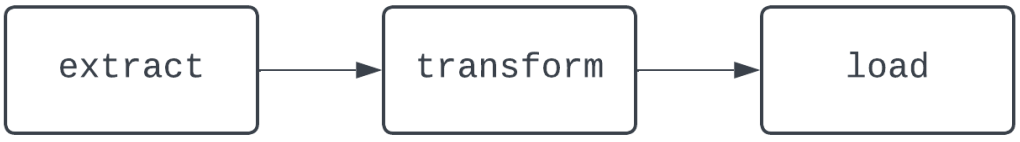
The simplest unit of the Airflow framework are tasks. *Tasks can be thought of as operations or, for most data teams, operations in a data pipeline.*

A traditional ETL workflow has three tasks:

- Extracting
- Transforming
- Loading data.

Dependencies define the relationships between tasks. Going back to our ETL example, the "load" task depends on the "transform" task, which, in turn, depends on the "extract" task. The combination of tasks and dependencies create **DAG**s, or directed-acyclic graphs.

DAGs represent data pipelines in Airflow.



The DAG above has three tasks, with two dependencies. It's considered a DAG because there are no loops (or cycles) between tasks. Here, the arrows show the directed nature of the process; first, the extract task is run, followed by the transform and load tasks. With DAGs, it's easy to see a distinct start and end to the process.

To schedule DAGs, execute tasks, and provide visibility into data pipeline execution details, Airflow leverages a Python-based architecture made up of the components below:

- **Scheduler:** Monitors all DAG definitions and decides when to schedule and trigger tasks based on the DAG's schedule interval and task dependencies.
- **Executor:** Executes the actual task instances that the scheduler queues.
- **Metadata database:** Stores all Airflow metadata including:
    - DAG definitions
    - Task instance states
    - Logs, execution history, variables, connections
- **Webserver (UI):** Provides a web-based UI for users to:
    - View DAGs and their status.
    - Trigger DAGs manually
    - Pause/resume DAGs
    - View logs for task runs

COMPONENTS OF AIRFLOW:

Airflow has 4 important components that are very important in order to understand how Airflow works.

- **Dynamic:** Airflow allows dynamic pipeline generation and configures using Python programming.
- **Extensible:** Airflow is very extensible. User can easily define their own operators as the requirement and suits the environment.
- **Elegant:** Airflow pipelines are lean and explicit.
- **Scalable:** Airflow uses a message queue for communication. It has a modular architecture.