APACHE AIRFLOW

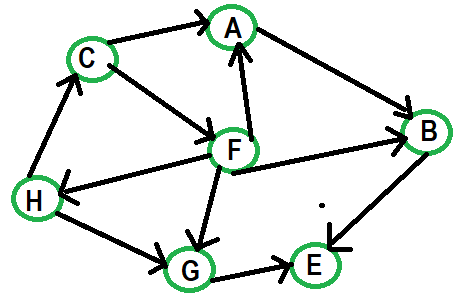
[Jobaoushadancse2021@jersalemengg.ac.in](mailto:Jobaoushadancse2021@jersalemengg.ac.in)

Job Aoushadan N

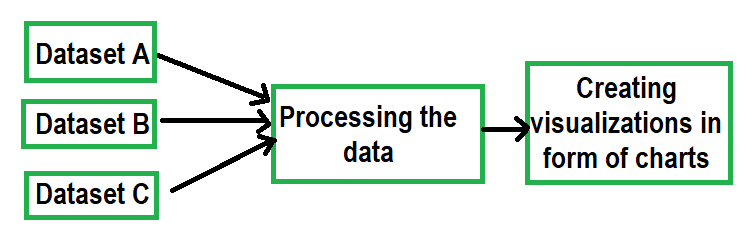
Apache Airflow is an open-source tool to programmatically author, schedule, and monitor workflows. It is used by Data Engineers for orchestrating workflows or pipelines. One can easily visualize your data pipelines' dependencies, progress, logs, code, trigger tasks, and success status. Complex data pipelines are managed using it. These data pipelines are used to deliver datasets that are easily used by business intelligence applications or machine learning models where a huge amount of data is required. It is one of the most robust platforms for data engineers. Batch-oriented workflows are developed, scheduled, and monitored efficiently. Apache Airflow is a workflow engine that  easily schedules and runs  complex data pipelines

**Working of Airflow and DAG:**

Workflow refers to the process of achieving some goal. They always have an end goal which could be something like creating visualizations for some data as given here. Directed Acyclic Graphs (abbreviated as DAG) are used to represent the workflow.



In the above-directed graph, if we traverse along the direction of the edges, and find no closed loop, we can conclude that no directed cycles are present. This type of graph is called a directed acyclic graph.



This is a workflow that shows that in order to create visualizations, various datasets are needed to be loaded independently and then processed. Loading datasets can be performed in parallel since they're independent of each other.

**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.

**Benefits of using Apache Airflow:**

* The Airflow community is very large and is still growing. It was started back in 2015 by Airbnb. So, there's a lot of support available.
* Apache Airflow is highly extensible which allows it to suit any environment. Custom cases are implemented very easily.
* The pipelines are generated dynamically and are configured as code using Python programming language.
* Rich scheduling and execution semantics are used to easily define complex pipelines and keep them running at regular intervals.
* With a little bit of Python knowledge, one can go about deploying on Airflow.
* It is free and open-source and has a lot of active users.
* Users can monitor and manage their workflows.

Since workflows are defined as Python codes they can be stored in version control so that they can be rolled back to previous versions. Workflows can be developed by multiple people simultaneously. A vast collection of existing components can be built since workflow components are extensible.

Apache Airflow is an open-source platform for authoring, scheduling and monitoring data and computing workflows. First developed by Airbnb, it is now under the Apache Software Foundation. Airflow uses Python to create workflows that can be easily scheduled and monitored. Airflow can run anything—it is completely agnostic to what you are running.

Benefits of Apache Airflow include:

* **Ease of use**—you only need a little python knowledge to get started.
* **Open-source community**—Airflow is free and has a large community of active users.
* **Integrations**—ready-to-use operators allow you to integrate Airflow with cloud platforms (Google, AWS, Azure, etc).
* **Coding with standard Python—**you can create flexible workflows using Python with no knowledge of additional technologies or frameworks.
* **Graphical UI**—monitor and manage workflows, check the status of ongoing and completed tasks.
* Airflow Use Cases
* Workloads
* Airflow Architecture
* Control Flow
* Airflow Components
* Airflow Best Practices
* Keep Your Workflow Files Up to Date
* Define the Clear Purpose of your DAG
* Use Variables for More Flexibility
* Set Priorities
* Define Service Level Agreements (SLAs)
* First, make sure you have installed Docker Desktop and Visual Studio. If not, lets do it now!
* Get Docker:
* Get Visual Studio Code
* Download the following file:
* Open Visual Studio Code
* create a new file .env and add the following lines
* AIRFLOW\_IMAGE\_NAME=apache/airflow:2.4.2
* AIRFLOW\_UID=50000
* docker-compose up -d
* create Admin user using below command:
* docker-compose run airflow-worker airflow users create --role Admin --username admin --email admin --firstname admin --lastname admin --password admin
* -----------------------------------------------------------------------------------------------------------------
* Apache Airflow allows you to schedule, monitor, and manage workflows with ease, and having it on your Windows PC can be a game-changer for your data engineering and automation projects.