What is airflow and components of airflow

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

Dags

A DAG is a model that encapsulates everything needed to execute a workflow. Some DAG attributes include the following:

- Schedule: When the workflow should run.
- Tasks: tasks are discrete units of work that are run on workers.
- Task Dependencies: The order and conditions under which tasks execute.
- Callbacks: Actions to take when the entire workflow completes.
- Additional Parameters: And many other operational details.

Core components

The following are the core components of Airflow:

- Scheduler: The scheduler is the heart of Airflow. It monitors all tasks and DAGs and schedules task instances to run as soon as their dependencies are fulfilled. When creating a new DAG run, the scheduler always picks the latest version of that DAG. When a task is ready to run, the scheduler uses its configured executor to run the task on a worker.
- API server: A FastAPI server that serves the Airflow UI, as well as three APIs:
    - An API for workers to interact with when running task instances.
    - An internal API for the Airflow UI that provides updates on dynamic UI components such as the state of task instances and DAG runs.
    - The public Airflow REST API that users can interact with.
- DAG processor: The DAG processor is responsible for retrieving and parsing the files from the location determined by the configured DAG bundle(s).
- Metadata database: The Airflow metadata database stores information vital to Airflow's functioning, such as Airflow connections, serialized DAGs, and XCom information. It also contains the history of previous DAG runs and task instances alongside metadata about their states. The most common backend used for the Airflow metadata database is PostgreSQL. See the Airflow documentation for supported versions.

- Triggerer: A separate process which supports running asynchronous Python functions as part of trigger classes. The triggerer is needed to use deferrable operators and event-driven scheduling.
- Operators: Operators define the specific action that each task performs. Airflow comes with a rich set of built-in operators that you can use to execute a variety of tasks. Some of the common operators include:

- PythonOperator: Executes Python functions.
- BashOperator: Runs bash commands or scripts.
- EmailOperator: Sends emails.
- DummyOperator: A no-op operator that is useful for defining task dependencies.
- MySqlOperator: Executes SQL commands on a MySQL database.
- PostgresOperator: Executes SQL commands on a PostgreSQL database.
- In addition to these built-in operators, you can create custom operators to fit specific use cases, making Airflow highly extensible.