Apache Airflow

Last Updated June 11, 2023

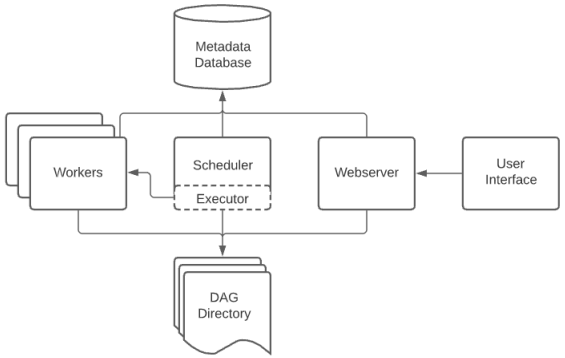
Airflow

Apache Airflow is an open-source platform for developing, scheduling, and monitoring batch-oriented workflows.

Architecture

An Airflow installation generally consists of the following components:

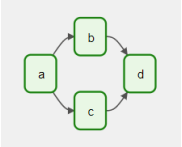
- A scheduler, which handles both triggering scheduled workflows, and submitting Tasks to the executor to run.
- An executor, which handles running tasks. In the default Airflow installation, this runs everything inside the scheduler, but most production-suitable executors actually push task execution out to workers.
- A webserver, which presents a handy user interface to inspect, trigger and debug the behaviour of DAGs and tasks.
- A folder of DAG files, read by the scheduler and executor (and any workers the executor has)
- A metadata database, used by the scheduler, executor and webserver to store state.



Concepts

DAGs

- A DAG (Directed Acyclic Graph) is the core concept of Airflow, collecting Tasks together, organized with dependencies and relationships to say how they should run. Directed: Data flows only forwards and not backwards. Acyclic: It avoids infinite loops. Here's a basic example DAG with four Tasks - A, B, C, and D:



Tasks

- A Task is the basic unit of execution in Airflow. There are three basic kinds of Task:

- **Operators**: Predefined task templates that you can string together quickly to build most parts of your DAGs.
  - **BashOperator** - executes a bash command
  - **PythonOperator** - calls an arbitrary Python function
- **Sensors**: A special subclass of Operators which are entirely about waiting for an external event to happen.
- **TaskFlow-decorated @task**, which is a custom Python function packaged up as a Task.

References

Architecture Overview
UI / Screenshots

Frank Facundo