

Getting started with Apache Airflow

Shyarnis Ghising

Dec 12, 2023



Contents

- 1 Introduction
- 2 Data Pipelines
 - Data pipelines as graphs
- 3 DAG in Python Code
- 4 Demonstration
 - Installation
 - Live Demonstration
- 5 References



Introduction

- Apache Airflow is a **batch-oriented** workflow for building data pipelines.
- It enables engineers to easily build **scheduled data pipelines** using a flexible Python framework.
- It **orchestrates** the different components responsible for processing data in data pipelines[1].

Introduction

- Apache Airflow is a **batch-oriented** workflow for building data pipelines.
- It enables engineers to easily build **scheduled data pipelines** using a flexible Python framework.
- It **orchestrates** the different components responsible for processing data in data pipelines[1].

Introduction

- Apache Airflow is a **batch-oriented** workflow for building data pipelines.
- It enables engineers to easily build **scheduled data pipelines** using a flexible Python framework.
- It **orchestrates** the different components responsible for processing data in data pipelines[1].

Data Pipelines

- It consists of **several tasks** that needed to be executed.
- Tasks need to be executed in a **specific order**.

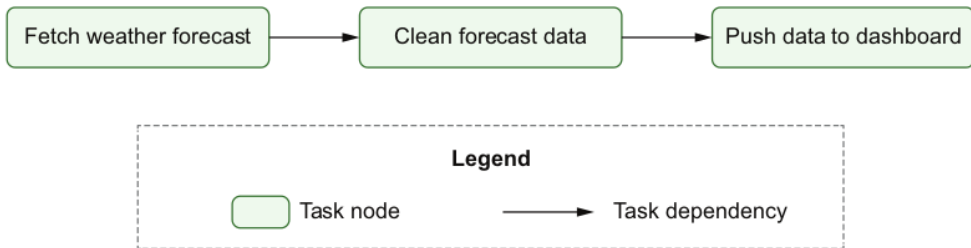


Figure: Data Pipeline for the weather dashboard

Data pipelines as graphs

- **Tasks** are represented by nodes/ vertices.
- **Dependencies between tasks** are represented by directed edges.

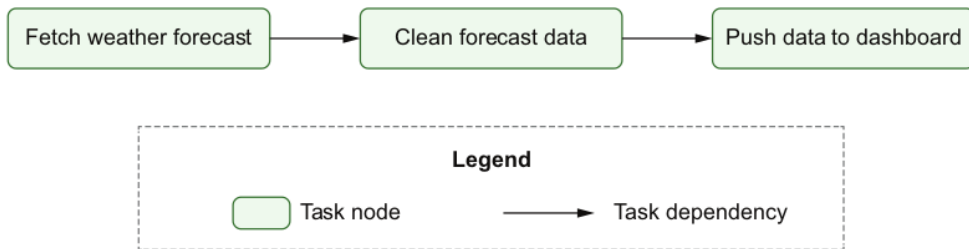
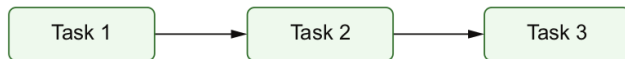


Figure: Data Pipeline represented as DAG

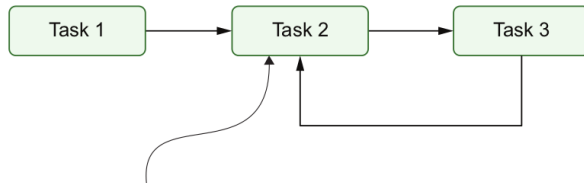
- Such graphs are called *directed acyclic graph* **DAG**.

A directed *acyclic* graph (DAG) of tasks



- Directed cyclic graph leads to **deadlock** situation.

A directed *cyclic* graph of tasks



Task 2 will never be able to execute,
due to its dependency on task 3,
which in turn depends on task 2.

DAG in Python Code

- Python provide flexibility for building DAGs.

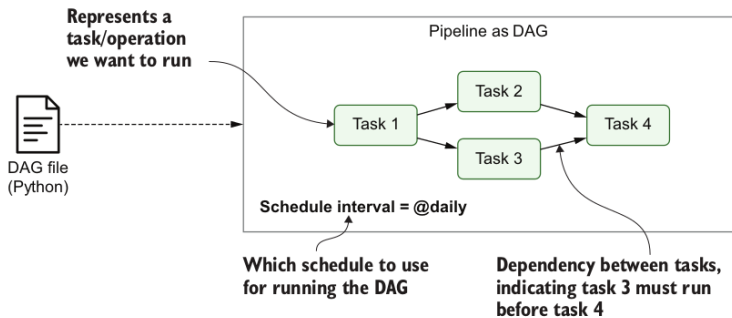


Figure: Pipelines are defined as DAGs using Python code

Installation

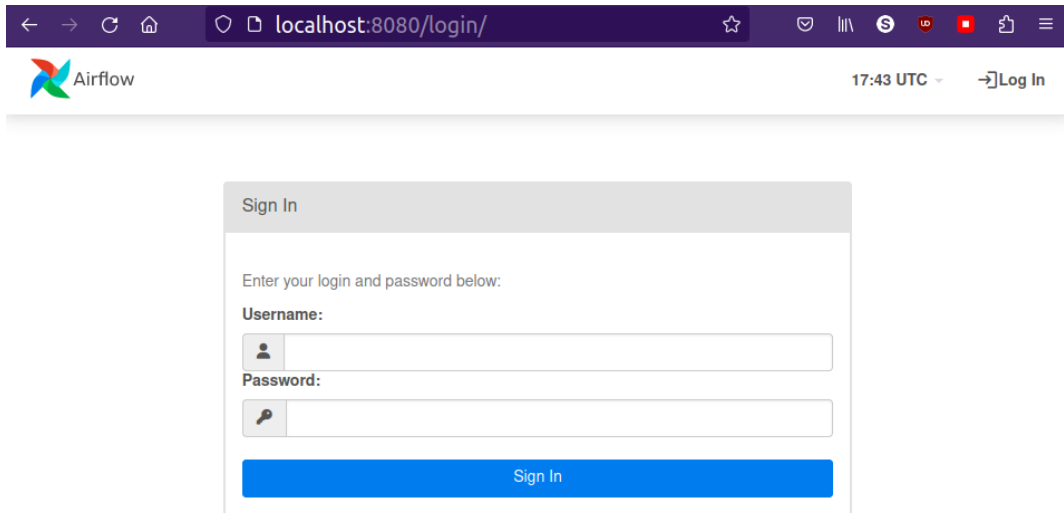
`Install Airflow` on your machine.

Example

- `AIRFLOW_VERSION==2.7.2`
- `PYTHON_VERSION==3.8`
- `pip install "apache-airflow==2.7.2" --constraint "https://raw.githubusercontent.com/apache/airflow/constraints-2.7.2/constraints-no-providers-3.8.txt"`

`constraints`

- airflow db migrate
- airflow users create --username <username>
--password <password> --firstname <fname> --lastname
<lname> --role Admin --email <email>
- airflow scheduler
- airflow webserver



The screenshot shows a web browser window with the address bar displaying `localhost:8080/login/`. The browser's navigation bar includes back, forward, and refresh buttons, as well as a home icon. The page title is "Airflow". In the top right corner, the time is "17:43 UTC" and there is a "Log In" button. The main content area features a "Sign In" form with the following elements:

- A header "Sign In" in a grey box.
- Text: "Enter your login and password below:"
- Label: "Username:"
- Input field for the username, preceded by a user icon.
- Label: "Password:"
- Input field for the password, preceded by a key icon.
- A blue "Sign In" button at the bottom of the form.

Figure: Airflow login view

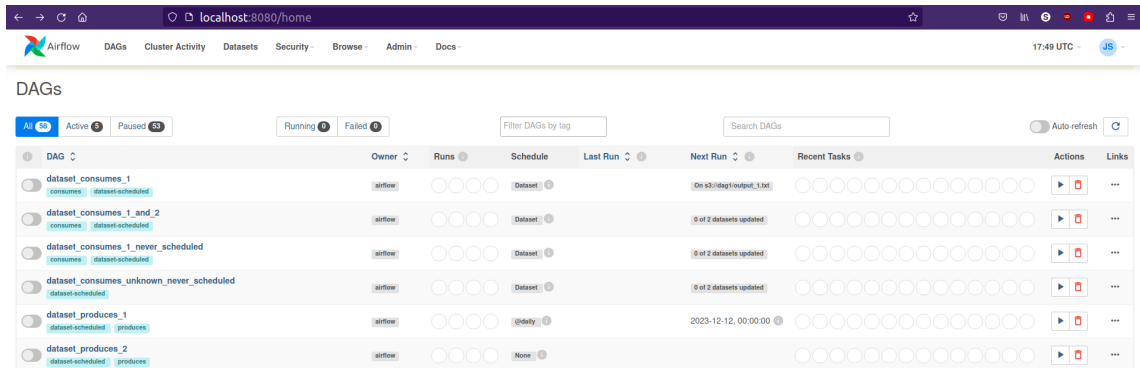


Figure: List of Airflow DAG

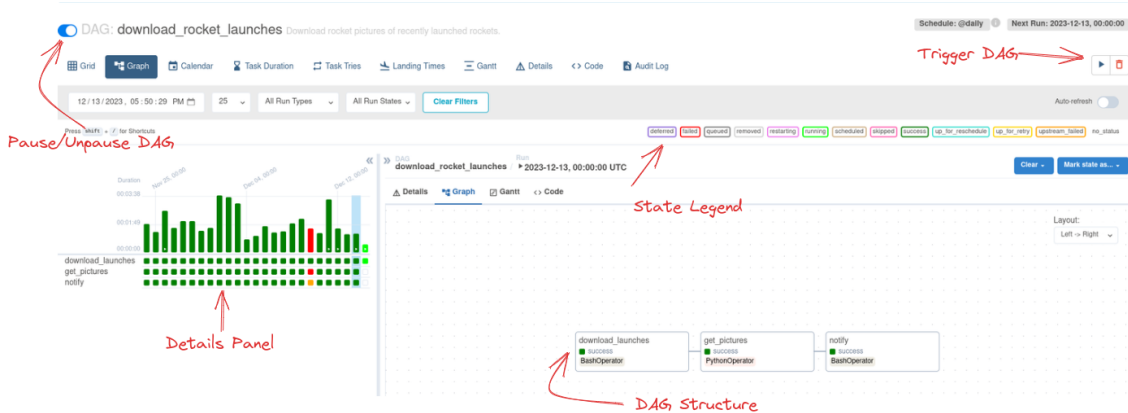


Figure: Airflow DAG in Action

Live Demonstration

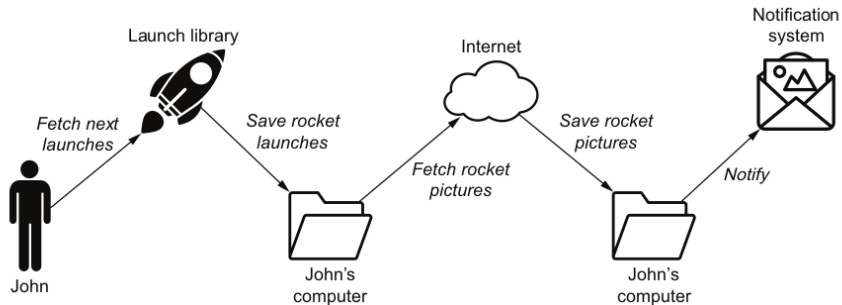


Figure: John's mental model of downloading rocket pictures

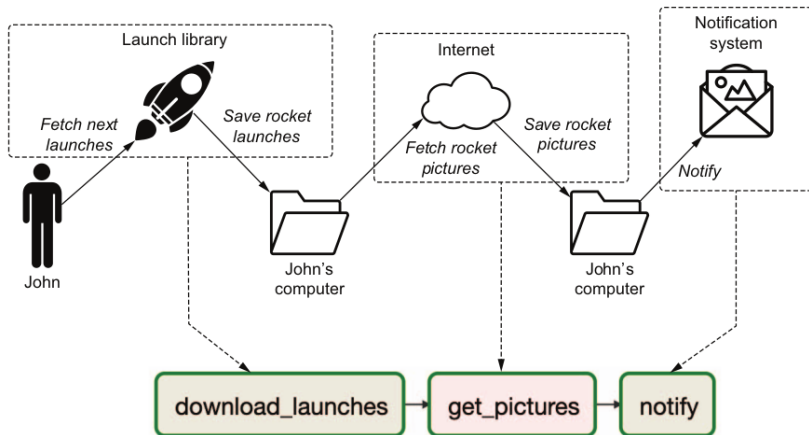



Figure: John's mental model mapped to tasks in Airflow

References

 Ruiter Harenslak.
Data Pipelines with Apache Airflow.
Manning, 2020.