**APACHE AIRFLOW:**

1. **Introduction:**

Apache Airflow is an **open-source workflow automation and scheduling platform** used to programmatically author, schedule, and monitor workflows. It allows users to define workflows as **Directed Acyclic Graphs (DAGs)** using Python code, making it flexible and scalable.

Airflow is widely used for **data engineering, ETL pipelines, machine learning pipelines, and automation tasks** across industries. Its modular architecture allows easy integration with various databases, cloud services, and tools.

Key Features:

* **Dynamic workflows:** Workflows are defined as Python code.
* **Scalability:** Can scale horizontally with Celery or Kubernetes executors.
* **Monitoring:** Provides a rich web interface for monitoring tasks and workflows.
* **Scheduling:** Supports cron-like scheduling and external triggers.
* **Extensibility:** Integrates with databases, cloud storage, and APIs.

1. **System Requirements:**

To run Apache Airflow, your system must meet the following requirements:

* **Operating System:** Linux, macOS, or Windows (WSL recommended)
* **Python Version:** 3.9, 3.10, or 3.11
* **Database:** SQLite (default), MySQL, or PostgreSQL for production
* **Disk Space:** Minimum 2 GB
* **Internet Connection:** Required for package installations

1. **Installation Procedure**

**Step 1: Create a Virtual Environment**

# Create virtual environment

python -m venv airflow\_venv

# Activate virtual environment

# Windows:

.\airflow\_venv\Scripts\activate

**Step 2: Install Apache Airflow**

# Export Airflow version and Python version

export AIRFLOW\_VERSION=2.7.1

export PYTHON\_VERSION=3.10

export CONSTRAINT\_URL="https://raw.githubusercontent.com/apache/airflow/constraints-${AIRFLOW\_VERSION}/constraints-${PYTHON\_VERSION}.txt"

# Install Airflow

pip install "apache-airflow==${AIRFLOW\_VERSION}" --constraint "${CONSTRAINT\_URL}"

On Windows PowerShell, replace export with set:

set AIRFLOW\_VERSION=2.7.1

set PYTHON\_VERSION=3.10

set CONSTRAINT\_URL=https://raw.githubusercontent.com/apache/airflow/constraints-%AIRFLOW\_VERSION%/constraints-%PYTHON\_VERSION%.txt

pip install "apache-airflow==%AIRFLOW\_VERSION%" --constraint "%CONSTRAINT\_URL%"

**Step 3: Initialize Airflow Database**

airflow db init

**Step 4: Create Admin User**

airflow users create \

--username admin \

--firstname First \

--lastname Last \

--role Admin \

--email [admin@example.com](mailto:admin@example.com)

**Step 5: Start Airflow Services**

# Start web server (default port 8080)

airflow webserver --port 8080

# Start scheduler in a new terminal

airflow scheduler

**Step 6: Creating a DAG**

# example\_dag.py

from airflow import DAG

from airflow.operators.bash import BashOperator

from datetime import datetime

with DAG('example\_dag', start\_date=datetime(2025, 8, 18schedule\_interval='@daily', catchup=False) as dag:

t1 = BashOperator(

task\_id='print\_hello',

bash\_command='echo "Hello, Airflow!"'

)

1. **Conclusion**

Apache Airflow is a powerful tool for **orchestrating workflows and automating tasks** in a scalable and maintainable way. By following the installation steps above, you can quickly set up Airflow for development, testing, or production purposes. The combination of DAGs, operators, and the web interface allows both developers and data engineers to monitor workflows efficiently.