مــكيــن
Makeen

سلطنة عُمان
وزارة النقل والاتصالات وتقنية المعلومات
Sultanate of Oman
Ministry of Transport, Communications and Information Technology

Code Academy

Introduction to Apache Airflow with Docker and MySQL Integration

Trainees:

- Iman Abdullah Al-Battashi
- Hebatallah Al-Jabri
- Ibreez Al-Aufi
- Zamzam Al-Salti

Instructor: Kulsoom Shoukat

Table of Contents

Introduction:

In the era of data-driven decision making, it is crucial to have robust tools that automate and manage data workflows. Apache Airflow is one of the most powerful open-source tools for designing, scheduling, and monitoring complex workflows. This project demonstrates how to create a fully automated ETL (Extract, Transform, Load) pipeline using Apache Airflow to extract data from a MySQL-based Online Bookstore database and export it as structured CSV files.

What is Apache Airflow?

Apache Airflow is an open-source platform to programmatically author, schedule, and monitor workflows. It allows users to define workflows as Directed Acyclic Graphs (DAGs), which are executed by the Airflow scheduler.

Key Idea: Automating and managing data pipelines in a simple, scalable, and reliable way.

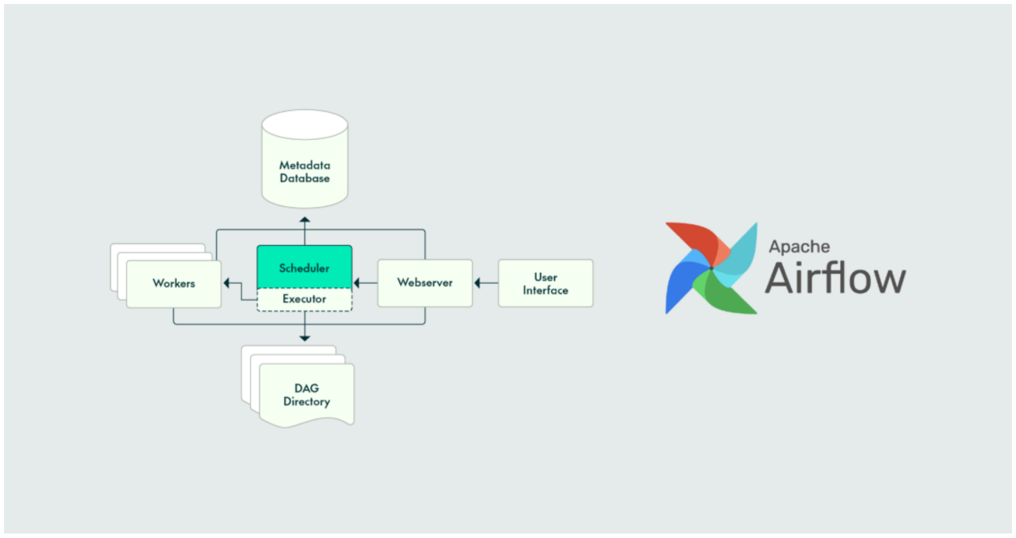

Figure 1: Airflow

Using Apache Airflow with Docker

Apache Airflow can be deployed using Docker, which simplifies the setup and makes it easier to manage dependencies. Docker is an important and efficient way to run Airflow because it ensures consistent environments across different systems, simplifies installation, and makes scaling easier.

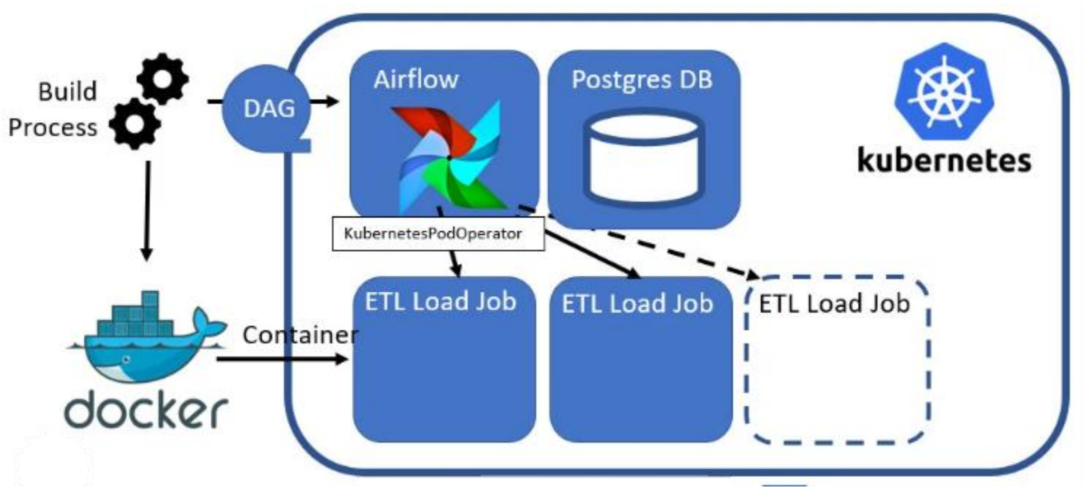

Figure 2: Docker

ETL VS ELT

Feature	ETL (Extract, Transform, Load)	ELT (Extract, Load, Transform)
Transformation	Happens before loading	Happens after loading
Storage	Requires staging area	Uses data warehouse directly
Speed	Slower for large data	Faster using modern engines
Use Case	Traditional systems	Big Data and Cloud-based systems

Figure 3: ETL VS ELT

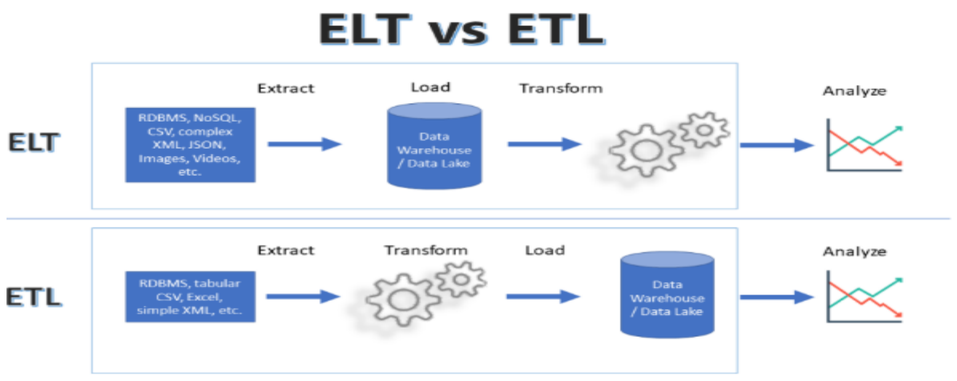
ELT vs ETL
ELT
Extract
Load
Transform
Analyze
RDBMS, NoSQL, CSV, complex XML, JSON, Images, Videos, etc.
Data Warehouse / Data Lake
ETL
Extract
Transform
Load
Analyze
RDBMS, tabular CSV, Excel, simple XML, etc.
Data Warehouse / Data Lake

Figure 4: ELT VS ETL

Simplified View of Apache Airflow

Apache Airflow has 3 main components:

- **Scheduler**: Triggers tasks based on time or conditions

- **Executor**: Runs the tasks (can be local, Celery, Kubernetes, etc.)

- **Web UI**: Monitor and manage your workflows visually

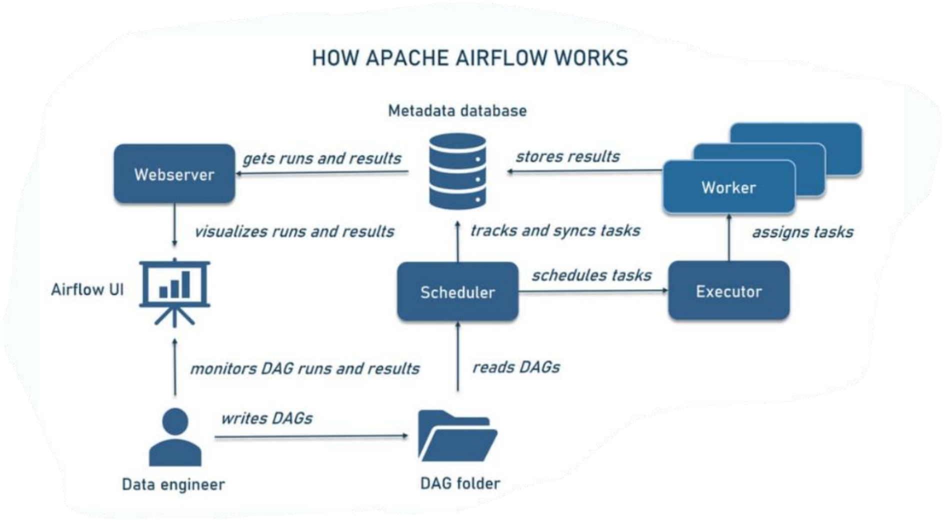

Figure 5: Apache

Features of Apache Airflow

- Dynamic pipeline generation using Python
- Web UI for managing workflows
- Scheduler and monitoring tools
- Plug-in support for various services (e.g., MySQL, AWS, etc.)
- Task dependencies and retries

Principles of Apache Airflow

1. **Dynamic**: Workflows are defined as Python code
2. **Scalable**: Supports distributed execution
3. **Extensible**: Easy to write plugins and operators
4. **Elegant**: Simple yet powerful user interface

What is a DAG?

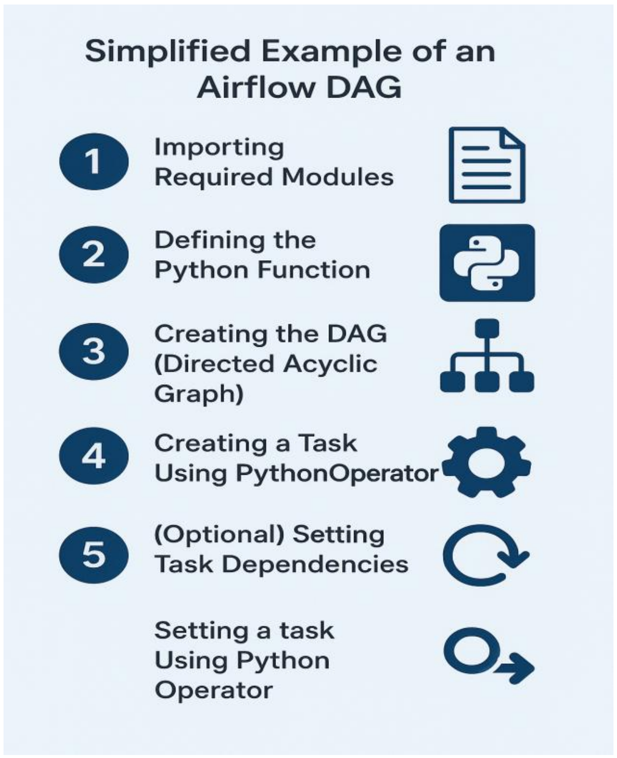
Simplified Example of an Airflow DAG
1 Importing Required Modules
2 Defining the Python Function
3 Creating the DAG (Directed Acyclic Graph)
4 Creating a Task Using PythonOperator
5 (Optional) Setting Task Dependencies
Setting a task Using Python Operator

Figure 6: Dag

Linking Apache Airflow with MySQL

To connect Airflow with MySQL:

1. Install MySQL client inside the Docker container:

   pip install apache-airflow-providers-mysql

2. Set up MySQL connection in the Airflow UI:
   - Go to Admin > Connections > Create
   - Connection ID: mysql_conn
   - Connection Type: MySQL
   - Host, Schema, User, Password, Port
3. Use the connection in a task:

```
from airflow.providers.mysql.operators.mysql import MySqlOperator


mysql_task = MySqlOperator(
   task_id='create_table',
   mysql_conn_id='mysql_conn',
   sql="""
      CREATE TABLE IF NOT EXISTS users (
         id INT AUTO_INCREMENT PRIMARY KEY,
         name VARCHAR(100)
      )
   """,
   dag=dag
)
```

Conclusion:

Apache Airflow is a powerful tool for coordinating and automating activities, especially in data engineering. Directed Acyclic Graphs (DAGs) enable a structured method of defining and monitoring tasks. Its flexibility and adaptability make it suitable for managing complicated pipelines with ease.

Airflow's scalability and portability are improved by integrating with Docker, allowing for consistent deployment across several environments. Combining it with MySQL expands its possibilities by providing a dependable backend for storing information and process states.

Throughout this paper, we looked at the differences between ETL and ELT, the features and concepts that distinguish Airflow, and presented a simplified overview of its architecture. Each of these features highlights how Airflow enables effective, dependable, and scalable data pipeline management.

In conclusion, Apache Airflow, especially when used alongside Docker and MySQL, offers a modern, adaptable solution for managing data workflows, making it a key tool in today's data engineering landscape.