**Apache Airflow**

Apache Airflow — это платформа с открытым исходным кодом, созданная для **программного создания, планирования и мониторинга рабочих процессов** (workflows). Она особенно полезна в задачах **обработки данных**, **ETL-пайплайнах**, **машинном обучении**, и других автоматизированных процессах.

**💡 Основные понятия Airflow:**

1. **DAG (Directed Acyclic Graph)** — направленный ацикличный граф, который описывает порядок выполнения задач. Это основа всего рабочего процесса в Airflow.
   * Каждая вершина графа — это задача.
   * Ребра — зависимости между задачами (что за чем выполняется).
2. **Task** — отдельная задача, например:
   * запуск Python-скрипта;
   * выполнение SQL-запроса;
   * перемещение файлов;
   * вызов API и др.
3. **Operator** — это шаблон или обёртка, определяющая, как выполняется задача. Есть разные типы операторов:
   * PythonOperator — выполняет Python-функцию;
   * BashOperator — запускает bash-команду;
   * PostgresOperator, BigQueryOperator и т. д. — выполняют запросы к базам данных.
4. **Scheduler** — компонент, который планирует запуск DAG'ов в соответствии с расписанием (например, каждый день в 8 утра).
5. **Executor** — отвечает за фактическое выполнение задач (может работать локально, параллельно или распределённо — например, с Celery или Kubernetes).
6. **Web UI** — веб-интерфейс для мониторинга DAG'ов, отслеживания их состояния, перезапуска задач и отладки.

### Как работает Airflow:

1. **Вы пишете DAG** — это Python-файл, в котором описывается, что и когда делать.
2. **Scheduler** регулярно проверяет DAG-и и определяет, какие задачи нужно запустить.
3. **Executor** запускает эти задачи (локально или в распределённой среде).
4. **Metadata Database** — хранит информацию о запусках DAG'ов, состоянии задач, логах и др.
5. **Web UI** показывает всё это в виде удобного графического интерфейса.

Настройка **Apache Airflow** в **Docker**

**🔧 Что мы будем делать?**

1. Установим нужные папки и файлы
2. Создадим docker-compose.yaml
3. Запустим контейнеры
4. Зайдём в Airflow и проверим, что всё работает

## Что нужно ДО начала

* Установлен **Docker**
* Установлен **Docker Compose**
* Любой терминал: PowerShell (Windows), Git Bash, или Terminal (Mac/Linux)

**Шаг 1**. В Docker в нужной папке в терминале введи:

mkdir airflow-docker

cd airflow-docker

mkdir dags, logs, plugins

**Шаг 2.** Создай файл docker-compose.yaml

Открой папку airflow-docker в Проводнике

Создай файл с именем docker-compose.yaml

Открой его и вставь туда следующий код:

version: '3.7'

services:

postgres:

image: postgres:13

environment:

POSTGRES\_USER: airflow

POSTGRES\_PASSWORD: airflow

POSTGRES\_DB: airflow

ports:

- "5432:5432"

volumes:

- postgres-db-volume:/var/lib/postgresql/data

airflow-webserver:

image: apache/airflow:2.9.1

restart: always

depends\_on:

- postgres

environment:

AIRFLOW\_\_CORE\_\_EXECUTOR: LocalExecutor

AIRFLOW\_\_CORE\_\_SQL\_ALCHEMY\_CONN: postgresql+psycopg2://airflow:airflow@postgres/airflow

AIRFLOW\_\_WEBSERVER\_\_EXPOSE\_CONFIG: "True"

volumes:

- ./dags:/opt/airflow/dags

- ./logs:/opt/airflow/logs

- ./plugins:/opt/airflow/plugins

ports:

- "8080:8080"

command: >

bash -c "

airflow db upgrade &&

airflow users create --username admin --password admin --firstname Admin --lastname User --role Admin --email admin@example.com &&

airflow webserver

"

airflow-scheduler:

image: apache/airflow:2.9.1

restart: always

depends\_on:

- airflow-webserver

environment:

AIRFLOW\_\_CORE\_\_EXECUTOR: LocalExecutor

AIRFLOW\_\_CORE\_\_SQL\_ALCHEMY\_CONN: postgresql+psycopg2://airflow:airflow@postgres/airflow

volumes:

- ./dags:/opt/airflow/dags

- ./logs:/opt/airflow/logs

- ./plugins:/opt/airflow/plugins

command: >

bash -c "

airflow scheduler

"

volumes:

postgres-db-volume:

### Шаг 3: Запусти контейнеры

1. В терминале убедись, что ты находишься в папке airflow-docker.
2. Выполни команду:

docker-compose up -d

Это запустит:

* PostgreSQL (база данных)
* Web-сервер Airflow
* Планировщик задач Airflow (scheduler)

**Шаг 5: Добавь пример DAG**

Создай файл dags/hello\_world\_dag.py со следующим содержимым:

from airflow import DAG

from airflow.operators.python import PythonOperator

from datetime import datetime

def hello():

print("Привет от Airflow!")

with DAG(

'hello\_world',

start\_date=datetime(2024, 1, 1),

schedule\_interval='@daily',

catchup=False

) as dag:

task = PythonOperator(

task\_id='say\_hello',

python\_callable=hello

)

**Как остановить**

Если хочешь остановить контейнеры:

docker-compose down