

Ejercicio Entendiendo el Docker-compose (contenido relacionado: Docker-compose)

En esta oportunidad trabajaremos con el siguiente [archivo](#) nuestro objetivo será entender cómo utilizar docker-compose para levantar el servicio de Airflow a través de Docker

Solución

1. Ir a la terminal
2. `mkdir airflow_prueba`
3. `cd airflow_prueba`
4. `mkdir dags`
5. `mkdir logs`
6. `mkdir plugins`
7. Posicionar nuestro [archivo](#) docker compose en la carpeta `airflow_prueba`
8. Iniciar docker
9. `docker-compose up -d`
10. `docker-compose ps`

Explicación de las secciones del archivo docker-compose

Servicios usados

- **postgres:** base de datos para almacenar los metadatos y registros de Apache Airflow.
- **airflow-webserver:** Este servicio utiliza la imagen de Apache Airflow y ejecuta el componente webserver de Airflow. El puerto 8080 del contenedor se mapea al puerto 8080 del host, lo que permite acceder a la interfaz web de Airflow desde el navegador.
- **airflow-scheduler:** Este servicio también utiliza la imagen de Apache Airflow y ejecuta el componente scheduler de Airflow. El planificador se encarga de programar y ejecutar las tareas definidas en los DAGs de Airflow.
- **airflow-init:** inicializar el entorno de Apache Airflow y establecer las configuraciones y bases de datos necesarias para su correcto funcionamiento

En cuanto a la configuración común compartida por varios servicios:

Se define una variable **x-airflow-common** que contiene algunas configuraciones comunes para los servicios de Airflow. Por ejemplo, la imagen de Docker que se utilizará, la configuración de la base de datos (conexión con PostgreSQL), ubicaciones de volúmenes compartidos (dags, logs y plugins), y otras configuraciones específicas del entorno de Airflow.

Las definiciones de servicios como **airflow-webserver**, **airflow-scheduler**, y **airflow-init** utilizan <<: *airflow-common para heredar las configuraciones definidas en **x-airflow-common**.