

**Airflow**

**Airflow is a platform to programmatically author, schedule and monitor workflows.**

Objetivo:

Reporte en el que muestres un ejemplo utilizando Airflow.

Desarrollo:

¿Qué es Apache Airflow?

Apache Airflow es un orquestador de código abierto que le permite crear, administrar y programar canalizaciones de datos, todo en Python. Es una tecnología imprescindible para los ingenieros de datos, pero también para los científicos de datos. La mayoría de los trabajos de ciencia de datos, al menos en las empresas más pequeñas, implican cierto grado de ingeniería de datos. Si tiene una cantidad decente de tuberías para administrar, no busque más allá de Airflow

1. crear un nuevo entorno virtual

Para empezar nuestra práctica de una manera sencilla lo primero será crear un entorno virtual de python, Al cual en este caso lo llame airflow:env, puede tener cualquier nombre pero será mas fácil ponerle un nombre con el que podamos identificar rápidamente este entorno lo base en python 3.9, pero se pueden cambiar ambos campos tanto el nombre como la version de python dependiendo cual tengas en el ordenador.

conda create --name airflow\_env python=3.9 -y

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Para activar el entorno que acabamos de crear se hace de una manera muy sencilla solamente insertamos en nuestra consola:

conda activate airflow\_env



Despues de activar nuestro entorno virtual, se nos despliega un mensaje en nuestra consola de la siguiente manera:



Este processo de creacion del entorno virtual fue creado con Miniforge, existen diferente maneras de crear entornos. Ahora veamos cómo instalar Airflow en nuesto ordenador de la siguiente manera.

1. Instalar Apache Airflow

La última versión de Airflow es la 2.4.1 pero en este cao instalaremos una version mas antigua para no tener ningun problema. El comando de instalación depende tanto de las versiones de Airflow como de Python, ya que tenemos que especificar una ruta al archivo de restricciones .

Creé un entorno basado en Python 3.9, por lo que la ruta del archivo de restricciones se ve así:

[https://raw.githubusercontent.com/apache/airflow/constraints-2.2.3 /constraints-3.9.txt](https://raw.githubusercontent.com/apache/airflow/constraints-2.4.1/constraints-3.9.txt)

Texto, Carta

Descripción generada automáticamente

En esta parte depende mucho la version que quieras utilizar de Airflow o Python, si se quiere utiliar diferentes versiones con tan solo cambiar 2.2-3 y/o 3.9 en la siguiente URL, es mas que suficiente pero para asefurar su funcionamiento correctamente, antes de instalrlo ponga la URL en su navegador para ver si existe dicha version:

pip install "apache-airflow==2.2.3" --constraint "https://raw.githubusercontent.com/apache/airflow/constraints-2.2.3/constraints-3.9.txt"

Esperaremos unos minutos en lo que se termina de instalar, dependera mucho de la velocidad de tu internet, ya que hay muchas dependencias por descargar:

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

1. Configurar la base de datos y el usuario de Airflow

Una vez que hayamos terminado de instalar Airflow, inicializaremos la base de datos con el siguiente comando en nuestra consola:

airflow db init

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Ahora crearemos la carpeta airflow en la raiz del direcctorio, asi que entramos a la carpeta con el siguiente comando:

cd ~/airflow



A continuacion creemos un usuario de Airflow, debera de tener los siguientes campos:

airflow users create \

--username admin \

--password admin \

--firstname <FirstName> \

--lastname <LastName> \

--role Admin \

--email <YourEmail>

La creación del usuario tomará un par de segundos. Una vez hecho esto, debería ver que el usuario con el rol de administrador se creó correctamente:

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

1. Inicie el servidor web y el programador de Airflow

Apache Airflow consta de dos partes principales: servidor web y programador . Tendrá que ejecutar ambos para inspeccionar y ejecutar sus DAG.

Primero, inicie el servidor web en el modo daemon (como un proceso en segundo plano):

airflow webserver -D

Texto

Descripción generada automáticamente

Una vez que se esté ejecutando, use un comando similar para ejecutar el Programador:

airflow scheduler -D

Texto

Descripción generada automáticamente

Airflow se ejecuta en el puerto 8080 de forma predeterminada, así que abra la siguiente URL en su navegador:

<http://localhost:8080>

Nos mostrara la pagina de inciar sesión, Rellenamos los datos con el usuario admin de Airflow que registramos anteriormente:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Despues de iniciar seesion, nos mandara a la siguiente pagina, Airflow de manera predeterminada nos da 30 DAG de ejemplo que no puede servir para crear uno propio.

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

1. Ejemplo

En nuestro apache airflow, en nuestra carpeta DAGs ya existente, crearemos un archivo en dicha carpeta, le puse de nombre first, cree un algoritmo en python en donde importaremos algunos modelos que necesitaremos para llevar a cabo la practica utilizando Airflow de una manera sencilla.

Texto

Descripción generada automáticamente

Despues de importar las librerias, declaramos un diccionario que nos servira para definir todos los parametros por default de cada una de nuestras tareas se van a ejecutar.

Texto

Descripción generada automáticamente

Posteriormente en nuestro algoritmo necesitaremos 3 funciones para loggiar la informacion, la primera funcion llamada “scrape” nos servira para descarfar la informacion desde el internet, la segunda funcion llamada “process” como su nombre lo dice es la que va a procesar toda la informacion que descargaremos de internet anteriormente y la tercera funcion que se llama “save” nos ayudara para que despudes de haber procesado la infmacion se almacenara en nuestra base de datos o en un archivo.

Texto

Descripción generada automáticamente

Depues de defininir nuestras funciones, mandaremos el DAG, primero especificaremos nuestro with Dag, en donde recibira el nombre del diccionario, una descripción de nuestro trabajo, una agenda de cada cuando queremos que se ejecute la actividad, inicio de nuestro trabajo y un conjunto de etiquetas para marcar los diferentes tipos de trabajos.

Texto

Descripción generada automáticamente

Para terminar, definirmeros la primera tarea que será para llamar la función screpe. Y así con las otras dos, de procesar y guardar la información.

Texto

Descripción generada automáticamente

En el apache Airflow podemos observar que ya tenemos nuestro algoritmo creado y subido al servidor.



Airflow es una gran herramienta de gran utilidad debido a que nos permite ejecutar varias tareas a la vez, tiene una gran interfaz que nos permite interactuar de manera muy dinámica, para mi gusto funciona muy bien y más por la opción de poder automatizar tareas programáticamente.scr

1. Codigo github