Universidad de Guadalajara
Sistema de Educación Media Superior
Centro Universitario de Ciencias Exactas e Ingenierías

Airflow

Materia: Computación Tolerante a Fallas

D06 2023 B

Alumno: Esquivel Barbosa Diego Humberto

Código: 211401635

Carrera: INCO

Fecha: 16/10/2023

Introducción

La computación tolerante a fallos se centra en el diseño, desarrollo y despliegue de sistemas que tienen la capacidad de resistir, mitigar y recuperarse de fallos y errores, manteniendo su funcionalidad esencial en situaciones adversas. Estos sistemas no solo aspiran a evitar la interrupción total, sino que también buscan ofrecer una operación confiable en escenarios donde componentes individuales pueden experimentar problemas. Ya sea en aplicaciones críticas para la seguridad, sistemas médicos, redes de comunicación o dispositivos de consumo, la tolerancia a fallos se ha convertido en un aspecto esencial para garantizar la integridad y la confianza en la tecnología que utilizamos diariamente.

Apache Airflow es una plataforma de código abierto diseñada para la automatización y programación de flujos de trabajo (workflows) complejos. Se utiliza para orquestar tareas y procesos en una variedad de aplicaciones, desde la administración de flujos de datos hasta la programación de tareas de ETL (Extract, Transform, Load). Airflow permite a los usuarios definir, programar y monitorear flujos de trabajo, lo que lo convierte en una herramienta poderosa para la automatización de procesos en entornos empresariales y de desarrollo. Las tareas individuales en un flujo de trabajo se representan como "DAGs" (Directed Acyclic Graphs), que describen las dependencias y el orden en que deben ejecutarse.

Código

```
from datetime import datetime, timedelta
from airflow import DAG
from airflow.operators.python_operator import PythonOperator
import pandas as pd

default_args = {
    'owner': 'tú',
    'depends_on_past': False,
    'start_date': datetime(2023, 10, 16),
    'retries': 1,
    'retry_delay': timedelta(minutes=5),
}

dag = DAG(
    'procesamiento_csv',
    default_args=default_args,
    schedule_interval=timedelta(days=1),  )

def descargar_csv():
    print("Descargando archivo CSV...")

def procesar_datos():
    data = {'Columna1': [1, 2, 3], 'Columna2': ['A', 'B', 'C']}
    df = pd.DataFrame(data)
    print("Procesando datos:")
    print(df)

def almacenar_resultado():
```

```
    print("Almacenando resultado...")

descarga_task = PythonOperator(
    task_id='descargar_csv',
    python_callable=descargar_csv,
    dag=dag,
)

procesamiento_task = PythonOperator(
    task_id='procesar_datos',
    python_callable=procesar_datos,
    dag=dag,
)

almacenamiento_task = PythonOperator(
    task_id='almacenar_resultado',
    python_callable=almacenar_resultado,
    dag=dag,
)

descarga_task >> procesamiento_task >> almacenamiento_task
```

Desarrollo del Código

Para instalar Apache Airflow en una computadora con Windows 11, puedes utilizar la versión oficial de Apache Airflow que es compatible con Windows. A partir de la versión 2.0.0, Apache Airflow es compatible con Windows y se puede instalar utilizando Python y pip. Aquí tienes los pasos para instalar Apache Airflow en Windows 11:

Nota: Asegúrate de que tengas Python instalado en tu sistema antes de comenzar.

Abre una ventana de línea de comandos (Command Prompt) o una terminal de PowerShell en tu computadora con Windows 11.

Crea un entorno virtual para Apache Airflow. Puedes utilizar la herramienta venv para esto. Ejecuta el siguiente comando para crear un nuevo entorno virtual en un directorio de tu elección. Reemplaza <directorio> con la ubicación deseada para el entorno virtual:

python -m venv <directorio>

```
C:\Windows\System32>python -m venv documentos
```

Activa el entorno virtual. Dependiendo de tu terminal, el comando puede variar:

<directorio>\Scripts\activate

```
C:\Windows\System32>documentos\Scripts\activate
```

Una vez que el entorno virtual esté activado, puedes usar pip para instalar Apache Airflow. Ejecuta el siguiente comando:

pip install apache-airflow

```
(documentos) C:\Windows\System32>pip install apache-airflow
Collecting apache-airflow
  Downloading apache_airflow-2.7.2-py3-none-any.whl (12.9 MB)
     ---------------------------------------- 12.9/12.9 MB 13.6 MB/s eta 0:00:00
Collecting alembic<2.0,>=1.6.3
  Using cached alembic-1.12.0-py3-none-any.whl (226 kB)
Collecting argcomplete>=1.10
  Downloading argcomplete-3.1.2-py3-none-any.whl (41 kB)
     ---------------------------------------- 41.5/41.5 kB 2.1 MB/s eta 0:00:00
Collecting asgiref
  Downloading asgiref-3.7.2-py3-none-any.whl (24 kB)
Collecting attrs>=22.1.0
  Using cached attrs-23.1.0-py3-none-any.whl (61 kB)
Collecting blinker
  Downloading blinker-1.6.3-py3-none-any.whl (13 kB)
Collecting cattrs>=22.1.0
  Downloading cattrs-23.1.2-py3-none-any.whl (50 kB)
     ---------------------------------------- 50.8/50.8 kB 2.7 MB/s eta 0:00:00
Collecting colorlog<5.0,>=4.0.2
  Downloading colorlog-4.8.0-py2.py3-none-any.whl (10 kB)
Collecting configupdater>=3.1.1
  Downloading ConfigUpdater-3.1.1-py2.py3-none-any.whl (34 kB)
Collecting connexion[flask]>=2.10.0
  Downloading connexion-2.14.2-py2.py3-none-any.whl (95 kB)
     ---------------------------------------- 95.1/95.1 kB ? eta 0:00:00
Collecting cron-descriptor>=1.2.24
  Downloading cron_descriptor-1.4.0.tar.gz (29 kB)
  Preparing metadata (setup.py) ... done
Collecting croniter>=0.3.17
  Downloading croniter-2.0.1-py2.py3-none-any.whl (19 kB)
```

Esto instalará la última versión de Apache Airflow en tu entorno virtual.

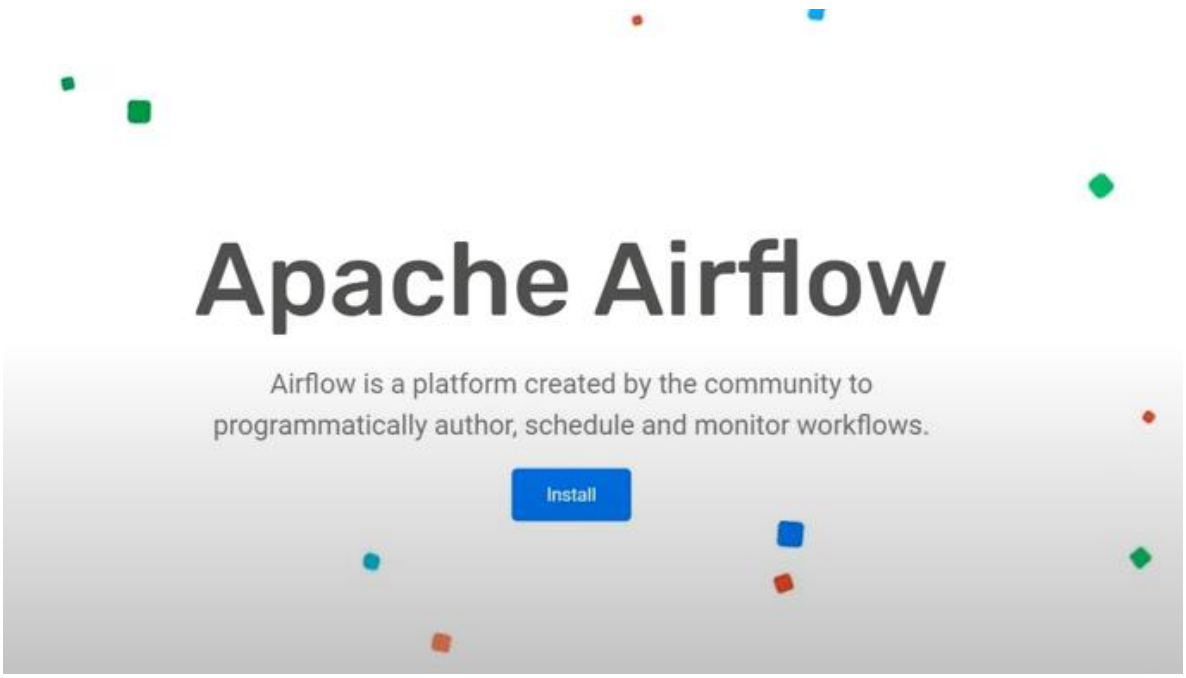
Luego, puedes inicializar la base de datos que Airflow utiliza para almacenar metadatos. Ejecuta el siguiente comando:

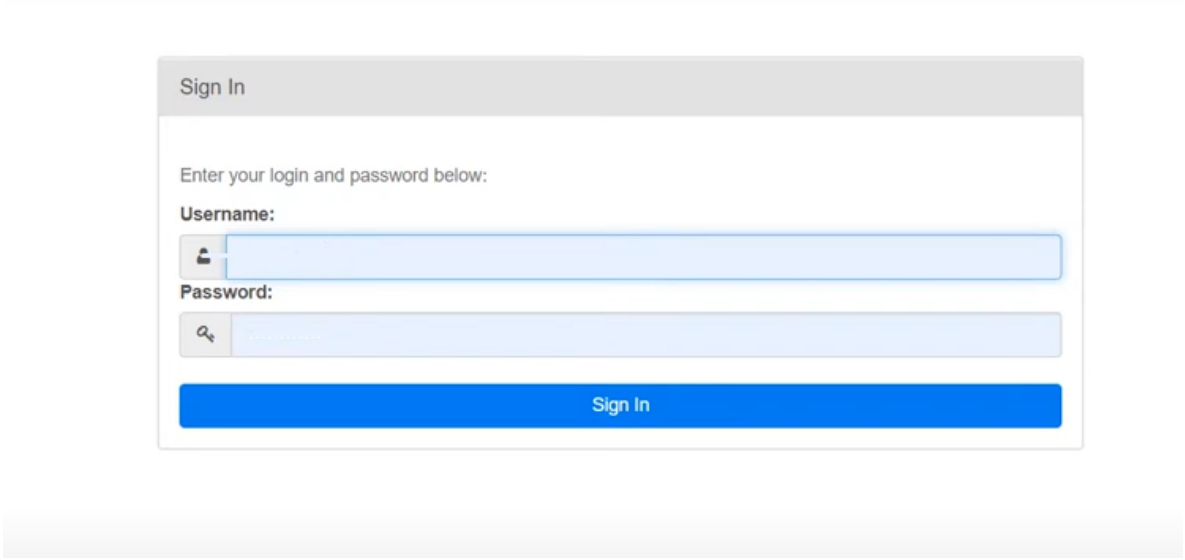
airflow db init

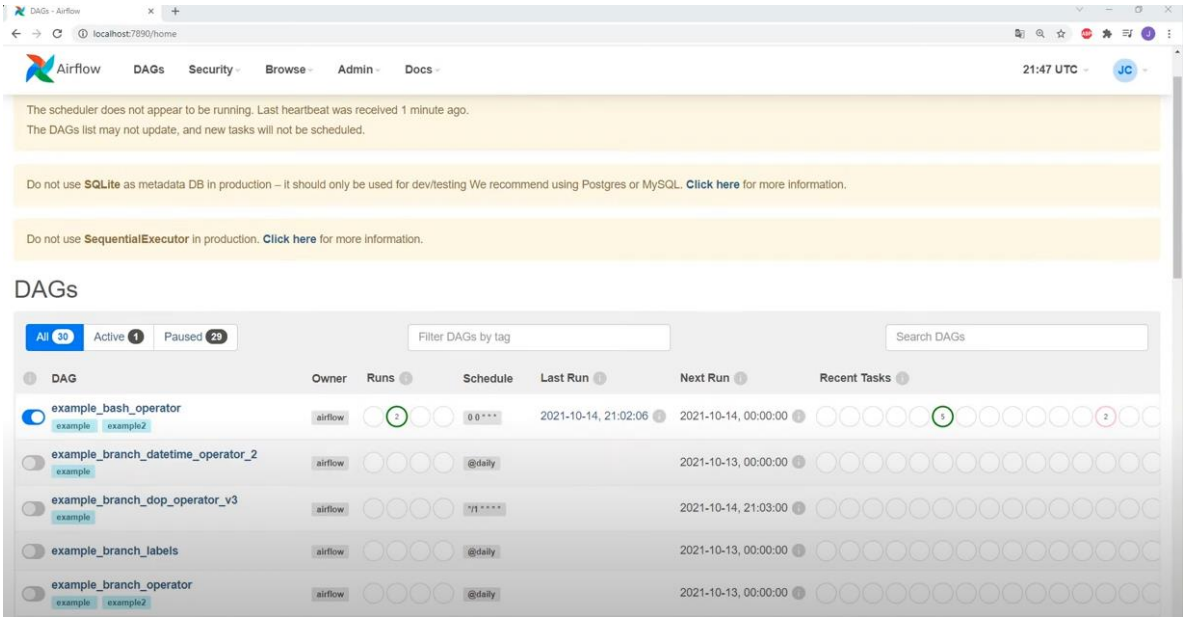
```
(documentos) C:\Windows\System32>airflow db init
C:\Windows\System32\documentos\Lib\site-packages\airflow\cli\commands\db_command.py:43 DeprecationWarning:
te the default connections
DB: sqlite:///C:\Users\Diego/airflow/airflow.db
[2023-10-16T23:58:16.649-0500] {migration.py:213} INFO - Context impl SQLiteImpl.
[2023-10-16T23:58:16.654-0500] {migration.py:216} INFO - Will assume non-transactional DDL.
INFO  [alembic.runtime.migration] Context impl SQLiteImpl.
INFO  [alembic.runtime.migration] Will assume non-transactional DDL.
INFO  [alembic.runtime.migration] Running stamp_revision  -> 405de8318b3a
WARNI [airflow.models.crypto] empty cryptography key - values will not be stored encrypted.
Initialization done
```

Finalmente, puedes iniciar el servidor web de Airflow y el Finalmente, puedes iniciar el servidor web de Airflow y el planificador con los siguientes comandos:

airflow webserver
airflow scheduler


Apache Airflow
Airflow is a platform created by the community to
programmatically author, schedule and monitor workflows.
Install


Sign In
Enter your login and password below:
Username:
Password:
Sign In



Conclusión

En esta actividad que se utilizo Apache Airflow se pudo ver diferentes herramientas de flujos de trabajo para varias aplicaciones como lo es la automatización de procesos, definición de dependencias, monitoreo y registro.

Ya que vimos que es altamente flexible y escalable, lo que lo hace adecuado para una amplia gama de casos de uso como lo son aplicaciones empresariales y de datos para tener un mejor control y automatización de la programación.

Bibliografía

- Biblioteca Prefect: https://docs.prefect.io/
- Documentación de Python: https://docs.python.org/3/
- Documentación de SQLite: https://sqlite.org/docs.html