



Centro Universitario de Ciencias
Exactas e Ingenierías.



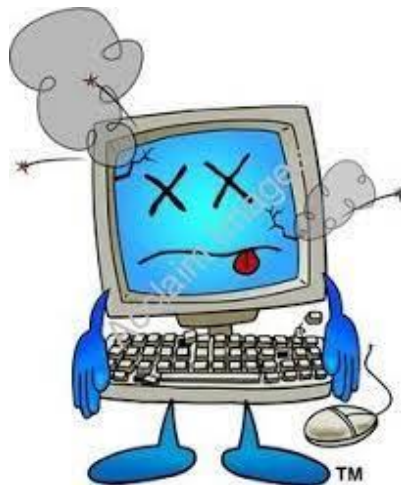
Ingeniería en computación.

Alumno: Vargas López David Guadalupe.

Computación tolerante a fallos.

Profesor: López Franco Michel Emanuel.

Sección: D06.



Guadalajara Jal. marzo del 2022.

AIRFLOW:

Instalación de Airflow

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart

Deployment Image Servicing and Management tool
Version: 10.0.18362.900

Error: 87

The online option is unknown.
For more information, refer to the help by running DISM.exe /?.

The DISM log file can be found at C:\Windows\Logs\DISM\dism.log
PS C:\Windows\system32>
```

En la imagen anterior se instala la instancia para que se pueda ejecutar Linux (Ubuntu) en Windows, mediante PowerShell en forma de administrador.

.matplotlib	08/02/2022 03:18 p. m.	Carpeta de archivos
.omnisharp	15/03/2021 07:30 p. m.	Carpeta de archivos
.templateengine	15/03/2021 07:38 p. m.	Carpeta de archivos
.tex	06/11/2020 01:30 p. m.	Carpeta de archivos
.VirtualBox	12/11/2021 04:46 p. m.	Carpeta de archivos
.virtualenvs	12/11/2021 08:20 a. m.	Carpeta de archivos
.vscode	07/09/2020 10:45 a. m.	Carpeta de archivos
actividad-11-David-1212	22/10/2020 07:05 p. m.	Carpeta de archivos
airflow	14/03/2022 09:18 a. m.	Carpeta de archivos
Búsquedas	23/03/2021 10:55 p. m.	Carpeta de archivos
Cisco Packet Tracer 7.3.1	21/11/2020 07:30 p. m.	Carpeta de archivos
Contactos	23/03/2021 10:55 p. m.	Carpeta de archivos
Descargas	14/03/2022 04:12 p. m.	Carpeta de archivos
Documents	12/05/2021 08:28 p. m.	Carpeta de archivos

Aquí se crea la carpeta llamada airflow, en donde se guardarán todos los dags que se crean para el funcionamiento del código de Python.




En la imagen anterior se puede observar como la forma de descargar Ubuntu en la Microsoft store para posteriormente proceder a su instalación.

```
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: airflow
New password:
Retype new password:
passwd: password updated successfully
Installation successful
```

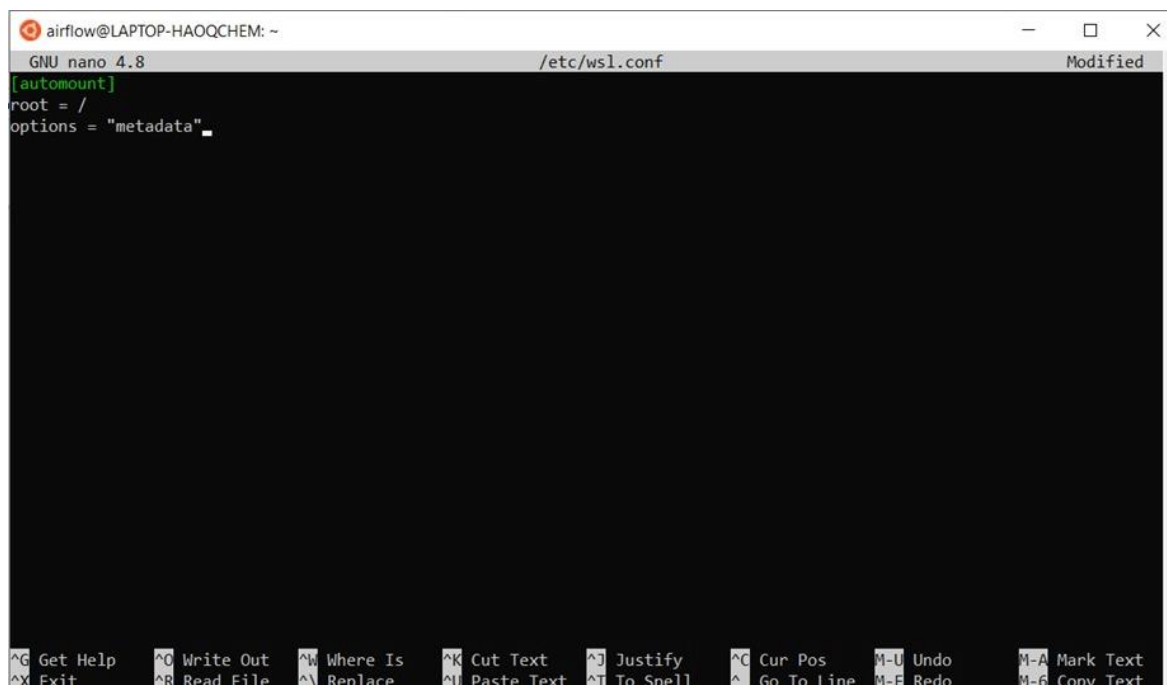
En esta parte se crea un usuario y una contraseña para poder utilizar Ubuntu y sus funciones en la que esta contraseña es para control interna del sistema.

```
airflow@LAPTOP-OMJ7SMP1:~$ sudo apt update && sudo apt upgrade
[sudo] password for airflow:
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:2 http://archive.ubuntu.com/ubuntu focal InRelease
Get:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [1317 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal/universe amd64 Packages [8628 kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [231 kB]
Get:8 http://security.ubuntu.com/ubuntu focal-security/main amd64 c-n-f Metadata [9808 B]
Get:9 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [799 kB]
Get:10 http://security.ubuntu.com/ubuntu focal-security/restricted Translation-en [114 kB]
Get:11 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [692 kB]
Get:12 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [121 kB]
Get:13 http://security.ubuntu.com/ubuntu focal-security/universe amd64 c-n-f Metadata [14.0 kB]
Get:14 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [20.7 kB]
Get:15 http://security.ubuntu.com/ubuntu focal-security/multiverse Translation-en [5196 B]
Get:16 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 c-n-f Metadata [500 B]
Get:17 http://archive.ubuntu.com/ubuntu focal/universe Translation-en [5124 kB]
Get:18 http://archive.ubuntu.com/ubuntu focal/universe amd64 c-n-f Metadata [265 kB]
Get:19 http://archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [144 kB]
Get:20 http://archive.ubuntu.com/ubuntu focal/multiverse Translation-en [104 kB]
Get:21 http://archive.ubuntu.com/ubuntu focal/multiverse amd64 c-n-f Metadata [9136 B]
Get:22 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1643 kB]
Get:23 http://archive.ubuntu.com/ubuntu focal-updates/main Translation-en [312 kB]
Get:24 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [14.8 kB]
Get:25 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [853 kB]
Get:26 http://archive.ubuntu.com/ubuntu focal-updates/restricted Translation-en [122 kB]
Get:27 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [910 kB]
```

Al poner el comando “sudo apt update && sudo apt upgrade” para asegurarnos que todo está actualizado.



Después de poner el comando “sudo nano /etc/wsl.conf” aparece esta pantalla para agregar 3 líneas para realizar algunos ajustes si se requieren.



Se ponen las tres líneas de comando y se guardan los cambios con ctrl+S para guardar datos y ctrl+x para cerrar.

```

airflow@LAPTOP-OMJ7SMPI:~$ cd ..
airflow@LAPTOP-OMJ7SMPI:/home$ ls
airflow
airflow@LAPTOP-OMJ7SMPI:/home$ dc ..
ls
Command 'dc' not found, but can be installed with:

sudo apt install dc

airflow@LAPTOP-OMJ7SMPI:/home$ cd ..
airflow@LAPTOP-OMJ7SMPI:/$ ls
bin    dev    home  lib    lib64  media  opt    root  sbin  srv  tmp  var
boot  etc    init  lib32  libx32 mnt    proc   run   snap  sys  usr
airflow@LAPTOP-OMJ7SMPI:/$

```

Aquí se muestran mediante el comando ls que existe dentro de cada unidad de las carpetas, así como también para ingresar a las mismas carpetas se utiliza el comando cd ...

```

airflow@LAPTOP-OMJ7SMPI:/$ cd home/airflow/
airflow@LAPTOP-OMJ7SMPI:~$ python3 --version
Python 3.8.10
airflow@LAPTOP-OMJ7SMPI:~$ sudo apt update
[sudo] password for airflow:
Hit:1 http://archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://archive.ubuntu.com/ubuntu focal-backports InRelease
Fetched 114 kB in 5s (24.1 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
airflow@LAPTOP-OMJ7SMPI:~$ _

```

Regresamos a la carpeta de home dentro de airflow , además que posteriormente se observa la versión de Python que se está ejecutando.

```

airflow@LAPTOP-OMJ7SMPI:~$ sudo apt install python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfwupdplugin1
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential cpp cpp-9 dpkg-dev fakeroot g++ g++-9 gcc gcc-9
  gcc-9-base libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5 libatomic1 libbinutils
  libc-dev-bin libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libdpkg-perl libexpat1-dev libfakeroot
  libfile-fcntllock-perl libgcc-9-dev libgomp1 libisl22 libitm1 liblsan0 libmpc3 libpython3-dev libpython3.8-dev
  libquadmath0 libstdc++-9-dev libtsan0 libubsan1 linux-libc-dev make manpages-dev python-pip-whl python3-dev
  python3-wheel python3.8-dev zlib1g-dev
Suggested packages:
  binutils-doc cpp-doc gcc-9-locales debian-keyring g++-multilib g++-9-multilib gcc-9-doc gcc-multilib autoconf
  automake libtool flex bison gdb gcc-doc gcc-9-multilib glibc-doc bzip libstdc++-9-doc make-doc
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential cpp cpp-9 dpkg-dev fakeroot g++ g++-9 gcc gcc-9
  gcc-9-base libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5 libatomic1 libbinutils

```

Se instala pip dentro de Python para poder utilizar estos paquetes dentro del airflow.

```

airflow@LAPTOP-OMJ7SMPI:~$ pip3 --version
pip 20.0.2 from /usr/lib/python3/dist-packages/pip (python 3.8)
airflow@LAPTOP-OMJ7SMPI:~$

```

Para observar la versión de pip que se instaló.

```

pip 20.0.2 from /usr/lib/python3/dist-packages/pip (python 3.8)
airflow@LAPTOP-OMJ7SMPI:~$ pip install apache-airflow
Collecting apache-airflow
  Downloading apache-airflow-2.2.4-py3-none-any.whl (5.3 MB)
    | 5.3 MB 1.8 MB/s
Collecting deprecated>=1.2.13
  Downloading Deprecated-1.2.13-py2.py3-none-any.whl (9.6 kB)
Collecting packaging>=14.0
  Downloading packaging-21.3-py3-none-any.whl (40 kB)
    | 40 kB 2.8 MB/s
Collecting apache-airflow-providers-sqlite
  Downloading apache_airflow_providers_sqlite-2.1.1-py3-none-any.whl (15 kB)
Collecting flask-wtf<0.15,>=0.14.3
  Downloading Flask_WTF-0.14.3-py2.py3-none-any.whl (13 kB)

```

Se instala apache airflow a través de pip install.

```

# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

export AIRFLOW_HOME=/c/Users/alexa/airflow_

# If not running interactively, don't do anything
case $- in
    *i*) ;;
    *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000

# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
shopt -s checkwinsize

# If set, the pattern "*" used in a pathname expansion context will
# match all files and zero or more directories and subdirectories.
#shopt -s globstar

```

Exportamos AIRFLOW_HOME para agregar las variables de entorno , posteriormente se cierra el Ubuntu para refrescar y que se realicen los cambios.

```

airflow@LAPTOP-OMJ7SMPI:~$ airflow version
2.2.4
airflow@LAPTOP-OMJ7SMPI:~$

```

Se observa la versión de airflow que se tiene para posteriormente utilizarlo.

```

airflow@LAPTOP-OMJ7SMPI:~$ airflow version
2.2.4
airflow@LAPTOP-OMJ7SMPI:~$ echo $AIRFLOW_HOME
/c/Usuarios/david/airflow
airflow@LAPTOP-OMJ7SMPI:~$

```

Imprime la ruta de conexión en la que nos encontramos con el comando, echo \$AIRFLOW_HOME.

Inicializamos airflow con la base de datos y esta se cree.

Se abre el web server para posteriormente observar su funcionamiento en el localhost.

Se crea un usuario y contraseña para poder ingresar a la pagina web.

Se declara usuario y contraseña.

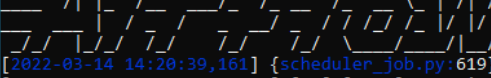
Usuario: david 12.

Contraseña: david.

Do not use **SequentialExecutor** in production. [Click here](#) for more information.

[illegible]

```
airflow@LAPTOP-OMJ7SMPI:~$ airflow scheduler
```



```
[2022-03-14 14:20:39,161] {scheduler_job.py:619} INFO - Starting the scheduler
[2022-03-14 14:20:39 -0600] [79] [INFO] Starting gunicorn 20.1.0
[2022-03-14 14:20:39,163] {scheduler_job.py:624} INFO - Processing each file at most -1 times
[2022-03-14 14:20:39 -0600] [79] [INFO] Listening at: http://0.0.0.0:8793 (79)
[2022-03-14 14:20:39 -0600] [79] [INFO] Using worker: sync
[2022-03-14 14:20:39,190] {manager.py:163} INFO - Launched DagFileProcessorManager with pid: 81
[2022-03-14 14:20:39,192] {scheduler_job.py:1137} INFO - Resetting orphaned tasks for active dag runs
[2022-03-14 14:20:39,203] {settings.py:55} INFO - Configured default timezone Timezone('UTC')
[2022-03-14 14:20:39,235] {manager.py:441} WARNING - Because we cannot use more than 1 thread (parsing_processes = 2) wh
en using sqlite. So we set parallelism to 1.
[2022-03-14 14:20:39 -0600] [80] [INFO] Booting worker with pid: 80
[2022-03-14 14:20:39 -0600] [82] [INFO] Booting worker with pid: 82
```

Nombre	Fecha de modificación	Tipo	Tamaño
logs	14/03/2022 02:20 p. m.	Carpeta de archivos	
airflow.cfg	14/03/2022 01:41 p. m.	Archivo CFG	44 KB
airflow	14/03/2022 02:24 p. m.	Data Base File	616 KB
airflow-webserver.pid	14/03/2022 02:10 p. m.	Archivo PID	1 KB
webserver_config	14/03/2022 01:41 p. m.	JetBrains PyCharm Com...	5 KB

Nos vamos a la carpeta de airflow, a el archivo seleccionado para modificarlo.

```
# Whether to load the DAG examples that ship with Airflow. It's good to
# get started, but you probably want to set this to ``False`` in a production
# environment
load_examples = False
```

Se modifica por false.

```
airflow@LAPTOP-OM75MP1:~$ airflow scheduler
[2022-03-14 14:31:50.000] {scheduler_job.py:619} INFO - Starting the scheduler
[2022-03-14 14:31:50.001] {scheduler_job.py:624} INFO - Processing each file at most -1 times
[2022-03-14 14:31:50.000] [514] [INFO] Starting gunicorn 20.1.0
[2022-03-14 14:31:50.000] [514] [INFO] Listening at: http://0.0.0.0:8793 (514)
[2022-03-14 14:31:50.000] [514] [INFO] Using worker: sync
[2022-03-14 14:31:50.072] {manager.py:163} INFO - Launched DagFileProcessorManager with pid: 515
[2022-03-14 14:31:50.074] {scheduler_job.py:1137} INFO - Resetting orphaned tasks for active dag runs
[2022-03-14 14:31:50.082] {settings.py:55} INFO - Configured default timezone Timezone('UTC')
[2022-03-14 14:31:50.092] {scheduler_job.py:1169} INFO - Marked 1 SchedulerJob instances as failed
[2022-03-14 14:31:50.095] {manager.py:441} WARNING - Because we cannot use more than 1 thread (parsing processes - 2) when using sqlite. So we set parallelism to 1.
[2022-03-14 14:31:51.000] [516] [INFO] Booting worker with pid: 516
[2022-03-14 14:31:51.000] [517] [INFO] Booting worker with pid: 517
[2022-03-14 14:32:39.000] [514] [INFO] Handling signal: winch
[2022-03-14 14:32:43.000] [514] [INFO] Handling signal: winch

airflow@LAPTOP-OM75MP1:~$ airflow webserver
, see help above.
airflow@LAPTOP-OM75MP1:~$ airflow webserver
[2022-03-14 14:31:34.344] {dagbag.py:500} INFO - Filling up the DagBag from /dev/null
[2022-03-14 14:31:34.711] {manager.py:512} WARNING - Refused to delete permission view, assoc with role exists DAG Runs
can create Admin
Traceback (most recent call last):
  File "/home/airflow/.local/bin/airflow", line 8, in <module>
    sys.exit(main())
  File "/home/airflow/.local/lib/python3.8/site-packages/airflow/_main_.py", line 48, in main
    args.func(args)
  File "/home/airflow/.local/lib/python3.8/site-packages/airflow/cli/cli_parser.py", line 48, in command
    return func(*args, **kwargs)
  File "/home/airflow/.local/lib/python3.8/site-packages/airflow/utils/cli.py", line 92, in wrapper
    return f(*args, **kwargs)
  File "/home/airflow/.local/lib/python3.8/site-packages/airflow/cli/commands/webserver_command.py", line 368, in webserver
    check_if_pidfile_process_is_running(pid_file=pid_file, process_name="webserver")
  File "/home/airflow/.local/lib/python3.8/site-packages/airflow/utils/process_utils.py", line 297, in check_if_pidfile_process_is_running
    raise AirflowException(f"The {process_name} is already running under PID {pid}.")
airflow.exceptions.AirflowException: The webserver is already running under PID 55.
airflow@LAPTOP-OM75MP1:~$
```

Y ahora se corre el Schedule para que deje de mandar el error a el webserver, ubuntu con el webserver para que se observe el funcionamiento de ambos.

Código de hola mundo:

Primero para realizar un código en Python:

```
from airflow import DAG
from airflow.operators.python import PythonOperator
from datetime import datetime

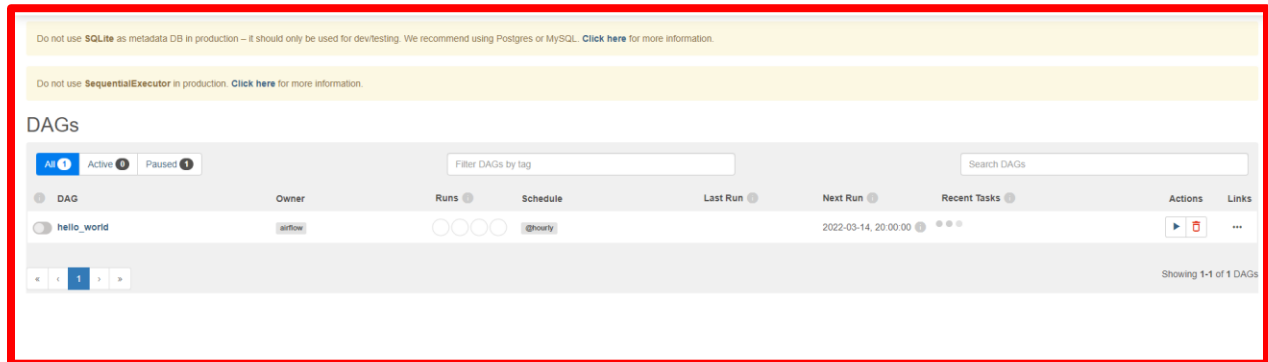
def hello_world():
    print("Hello world")

with DAG(dag_id="hello_world",
        start_date=datetime(2022,3,14),
        schedule_interval="@hourly",
        catchup=False) as dag:

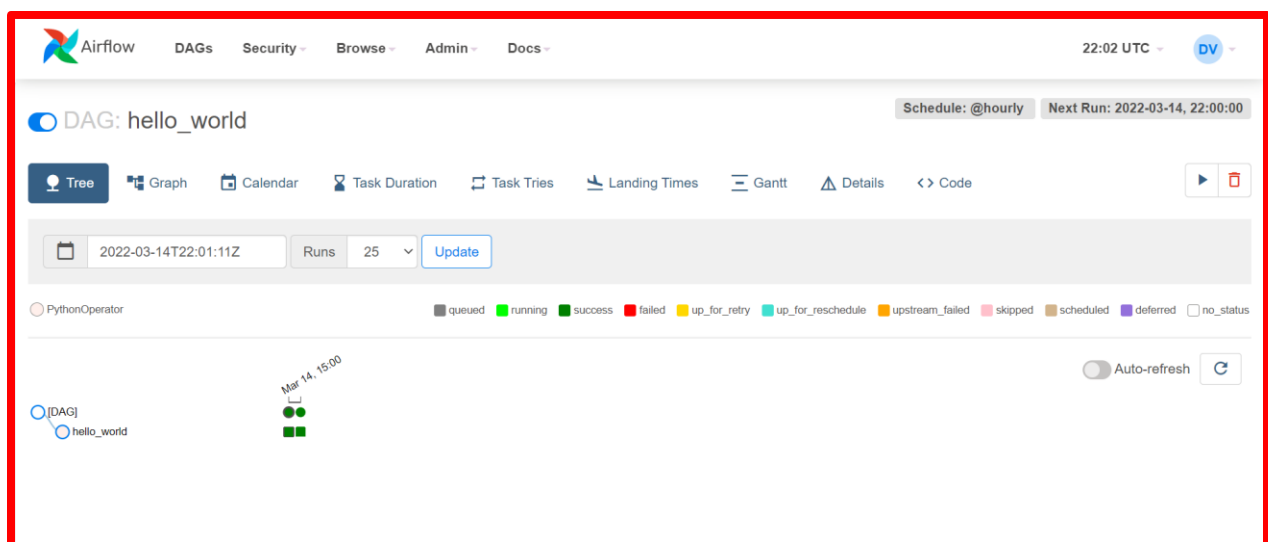
    task1 = PythonOperator(
        task_id="hello_world",
        python_callable=hello_world
```

```
)  
  
task1
```

Este código es de un simple hola mundo, solo para observar el funcionamiento.



Y se observa en el dag como se agregó de acuerdo con el nombre del task.



Aquí se muestra la evolución de la tarea, así como también el flujo de estas, así como también se muestra el árbol del algoritmo.

Triggered hello_world, it should start any moment now.

Do not use **SQLite** as metadata DB in production - it should only be used for dev/testing. We recommend using Postgres or MySQL. [Click here](#) for more information.

Do not use **SequentialExecutor** in production. [Click here](#) for more information.

DAGs

All 1 Active 1 Paused 0

Filter DAGs by tag

Search DAGs

DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks	Actions	Links
hello_world	airflow	<div><div></div><div></div><div></div></div>	@hourly	2022-03-14, 22:03:29	2022-03-14, 22:00:00		<div><div></div><div></div><div></div></div>	

Showing 1-1 of 1 DAGs

Aquí se muestra como ya está en ejecución y esta en la segunda etapa de la misma.

Airflow DAGs Security Browse Admin Docs

22:06 UTC DV

DAG: hello_world

Schedule: @hourly Next Run: 2022-03-14, 22:00:00

Tree Graph Calendar Task Duration Task Tries Landing Times Gantt Details <> Code

```
1 from airflow import DAG
2 from airflow.operators.python import PythonOperator
3 from datetime import datetime
4
5 def hello_world():
6     print("Hello world")
7
8 with DAG(dag_id="hello_world",
9         start_date=datetime(2022,3,14),
10        schedule_interval="@hourly",
11        catchup=False) as dag:
12
13    task1 = PythonOperator(
14        task_id="hello_world",
15        python_callable=hello_world
16    )
17
18 task1
```

Toggle Wrap

Aquí se muestra el código que se esta ejecutando, dentro de el mismo airflow.

DAG: hello_world

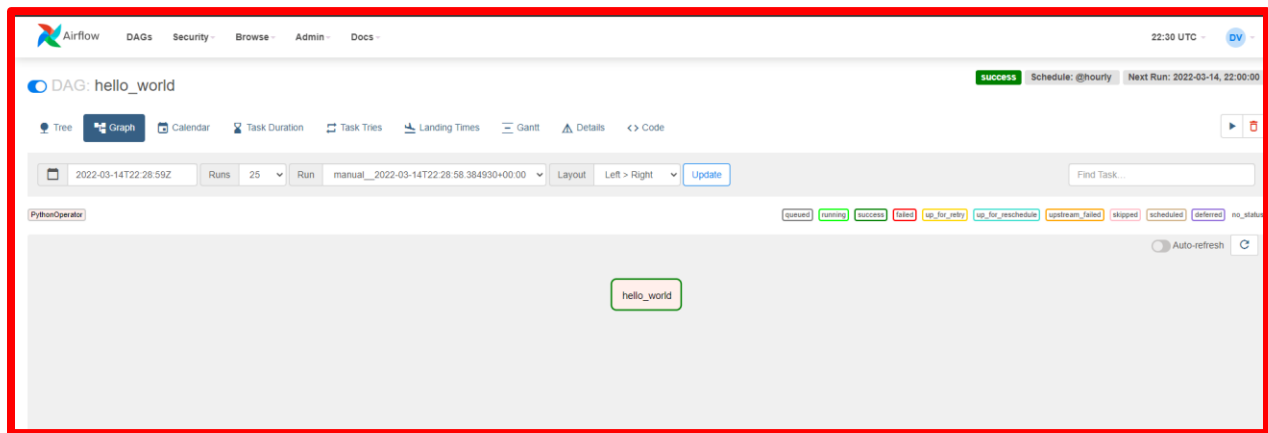
Tree Graph Calendar Task Duration Task Tries Landing Times Gantt Details <> Code

DAG Details

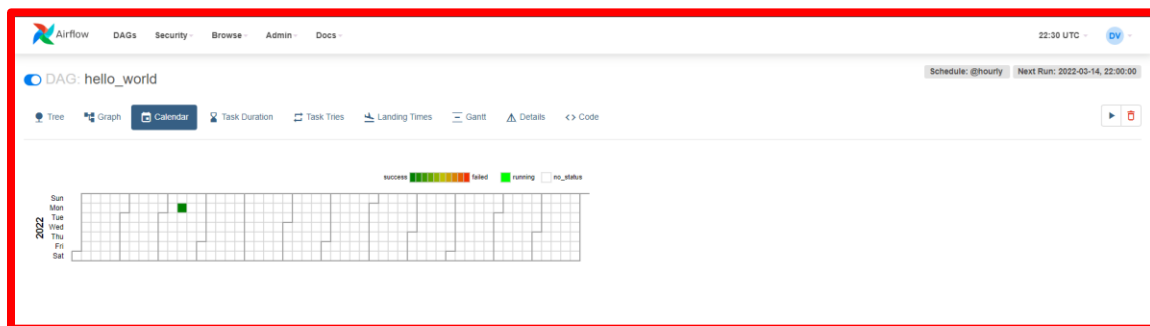
SUCCESS ?

Schedule Interval	@hourly
Catchup	False
Started	True
End Date	None
Max Active Runs	0 / 16
Concurrency	16
Default Args	{}
Tasks Count	1
Task IDs	['hello_world']
Relative file location	main.py
Owner	airflow
DAG Run Timeout	None
Tags	None

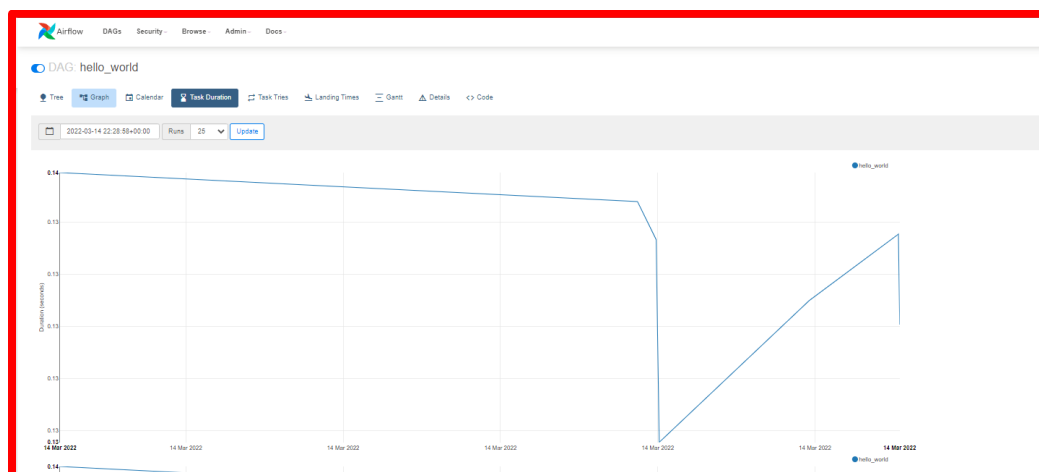
Algunos de los detalles del dag creado anteriormente.



Aquí se muestra el árbol, pero debido a que solo es un hola mundo solo tiene una rama.



El calendario de las actividades a realizar.



Se muestra el diagrama de flujo del código que se está ejecutando en Python.

Código de Python en GitHub:

<https://github.com/David-1212/airflow.py>

Conclusión:

En esta practica me quedo bastante claro el como se utiliza el airflow, así como también use por primera vez La distribución de Ubuntu, en línea de comandos, así como también entendí la estructura que debía tener para que se agregara el dag a airflow.

Al principio tuve bastantes problemas, debido a que no me reconocía la versión de el airflow, pero con el paso del tiempo me di cuenta de un pequeño error al instalar una librería de pip, por lo que no se realizaban correctamente las instalaciones y por ende hacia cosas inesperadas.

Para este ejemplo realice un simple hola mundo como antes se menciona, pero funciona correctamente y realiza el testeo cada hora.

Bibliografía:

- *#SGVirtual 20.12 - Automatizando ideas con Apache Airflow.* (2021, February 4).

YouTube. Retrieved March 14, 2022, from

<https://www.youtube.com/watch?v=ewK4KszmeTI>