Data Scheduling con Airflow

Luis Morales Meza
Data Engineer Senior

datapath

APACHE AIRFLOW

Apache Airflow

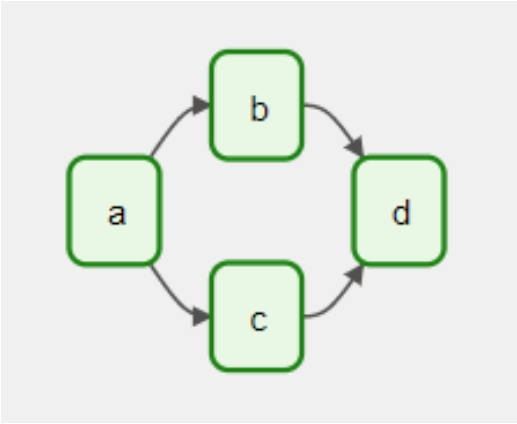
Es una herramienta de tipo workflow manager : Gestionar , monitorizar y planificar flujos de trabajos, usada como orquestador de servicios

Airflow se usa para automatizar trabajos dividiendolos en subtareas . Permite su planificación y monitoización desde una herramienta centralizada . Los casos más comunes son la automatización de Ingesta de datos , acciones de mantenimiento periódica y tareas de administración.

¿QUE ES AIRFLOW?

DAGs

Son colecciones de tareas o trabajos a ejecutar conectados mediante relaciones y dependencias . Es decir como deben ejecutarse



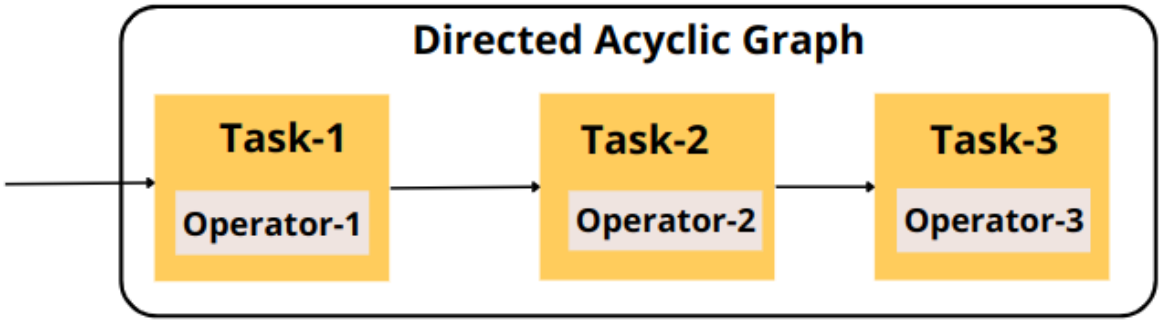
DAG = gráfico acíclico dirigido

Aciclico : No puededen formar ciclos , la ejecución no puede volver a un nodo ejecutado

Dírigido
: Las relaciones entre los nodos tienen un solo sentido

OPERADORES

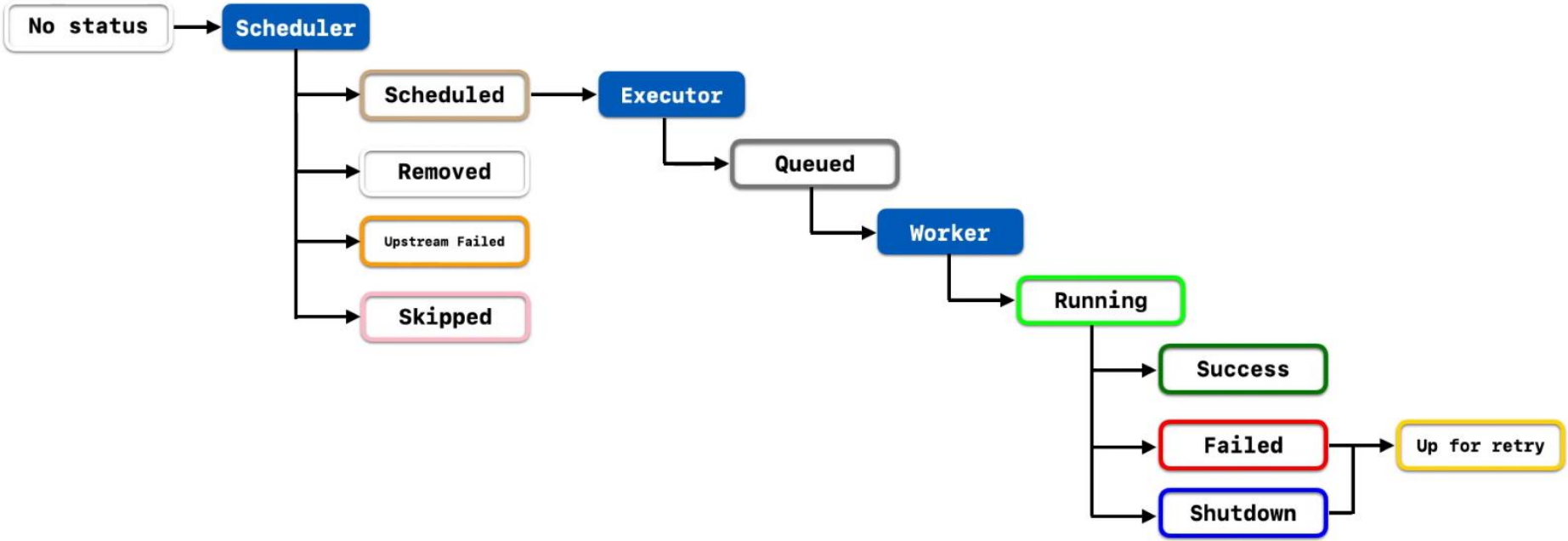
Son componentes centrales de cualquier flujo de trabajo definido en AirFlow. El operador representa una única tarea que se ejecuta de forma independiente sin compartir ninguna información. Los operadores pueden ejecutar varias acciones, como una función Python



TIPOS DE OPERADORES

- **BashOperator:** Se utiliza para ejecutar un comando bash.
- **PythonOperator:** se utiliza para ejecutar la función invocable o de Python.
- **EmailOperator:** Se utiliza para enviar correos electrónicos al receptor.
- **MySqlOperator:** Se utiliza para ejecutar la consulta SQL para la base de datos MySql.
- **S3ToHiveOperator:** Transfiere datos de Amazon S3 a Hive.
- **HttpOperator:** se utiliza para activar un punto final HTTP.
- **BranchingOperator:** Al igual que PythonOperator, excepto que espera un python_callable que devuelve un task_id.

CICLO DE VIDA DE UNA TAREA



Trigger Rules

Una regla de activación define por qué se ejecuta una tarea y en función de qué condiciones. De forma predeterminada, todas las tareas tienen la misma regla de activación all_success

- `all_success` : (predeterminado) todos los padres lo han logrado
- `all_failed` : todos los padres están en un estado `failed` o `upstream_failed`
- `all_done` : todos los padres han terminado con su ejecución
- `one_failed` : se activa tan pronto como al menos uno de los padres falla, no espera a que todos los padres hayan terminado
- `one_success` : se activa tan pronto como al menos uno de los padres tiene éxito, no espera a que todos los padres terminen
- `none_failed` : todos los padres no han fallado (`failed` o `upstream_failed`), es decir, todos los padres han tenido éxito o se han omitido
- `none_skipped` : ningún padre está en un `skipped` estado, es decir, todos los padres están en un estado `success` , `failed` o `upstream_failed`
- `dummy` : las dependencias son sólo para mostrar, se activan a voluntad

AIRFLOW SCHEDULER

Preestablecido	significado
`None`	No programe, utilícelo exclusivamente para DAG "activados externamente"
`@once`	Agendar una vez y solo una vez
`@hourly`	Corre una vez por hora al comienzo de la hora.
`@daily`	Corre una vez al día a medianoche.
`@weekly`	Corre una vez a la semana a la medianoche del domingo por la mañana.
`@monthly`	Ejecutarse una vez al mes a la medianoche del primer día del mes.
`@quarterly`	Ejecute una vez cada trimestre a la medianoche del primer día.
`@yearly`	Ejecutarse una vez al año a la medianoche del 1 de enero.

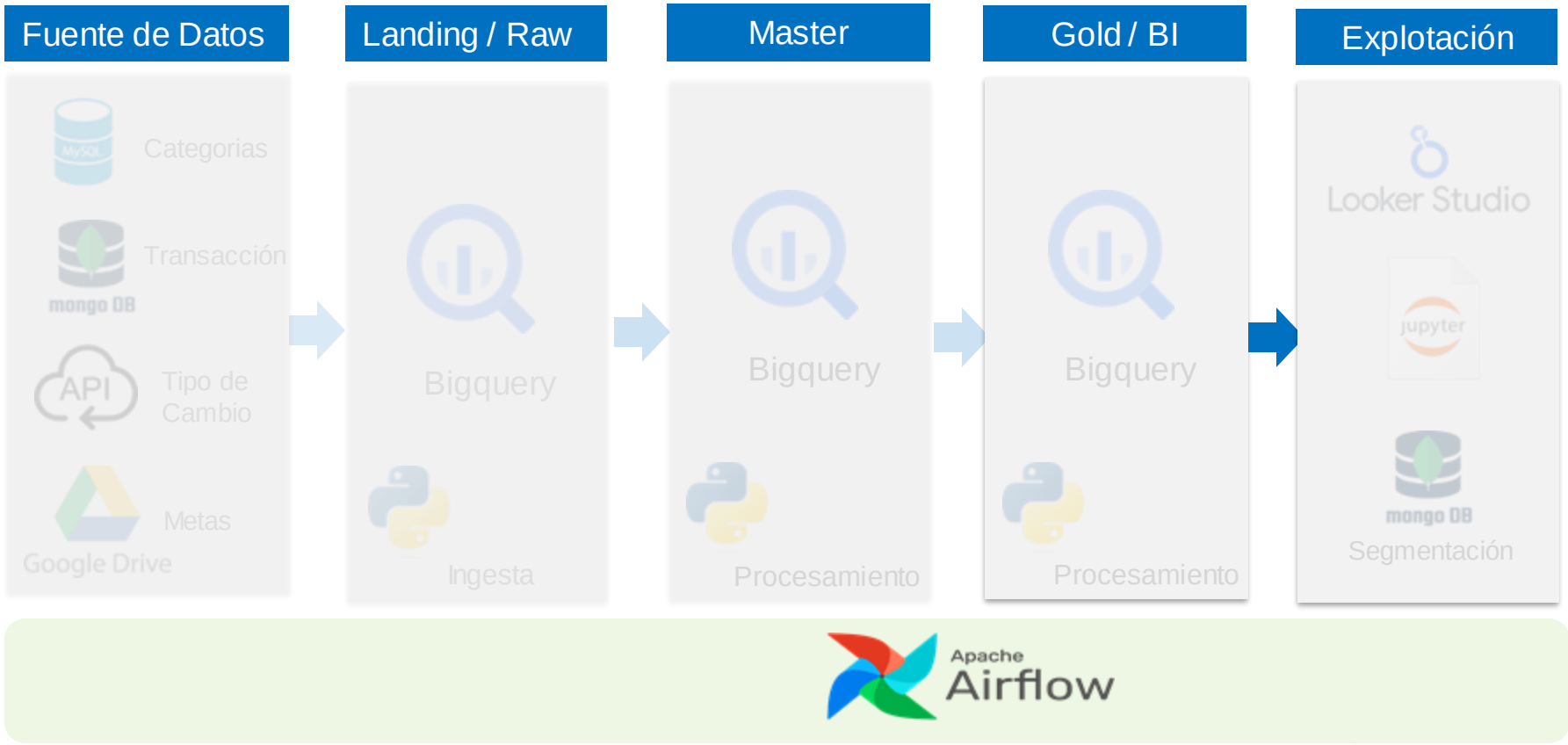
AIRFLOW SCHEDULER

```
# * * * * *  command to execute
# ┬ ┬ ┬ ┬ ┬
# │ │ │ │ │
# │ │ │ │ └──── day of week (0 - 7)
# │ │ │ └─────── month (1 - 12)
# │ │ └──────── day of month (1 - 31)
# │ └───────── hour (0 - 23)
# └──────── min (0 - 59)
```

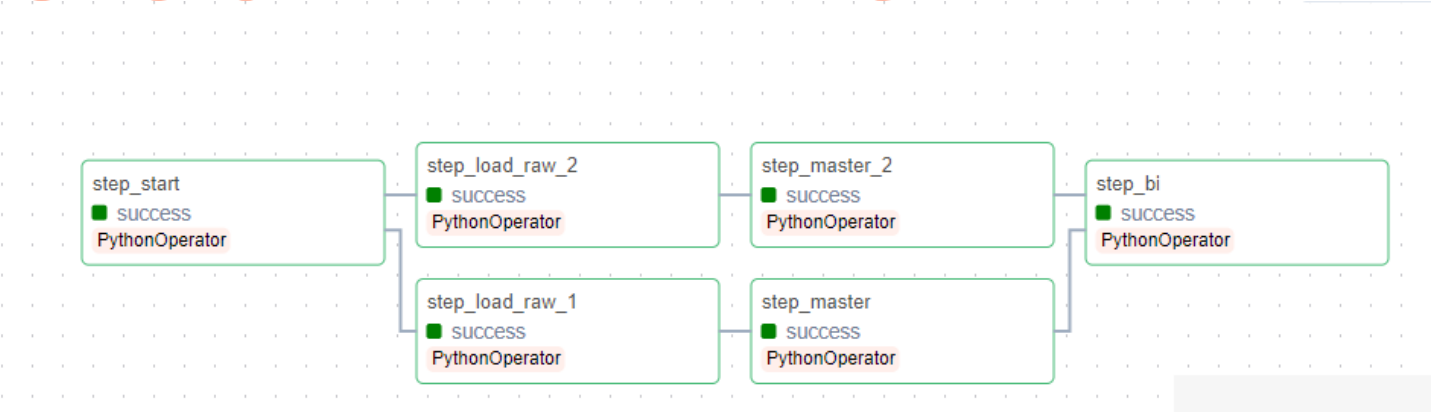
https://crontab.guru/

¿ PREGUNTAS ?

Arquitectura de datos del proyecto



TAREA : CREAR EL SIGUIENTE DAG



VAMOS A CREAR DAGs

Crear un ambiente de prueba en https://www.astronomer.io/

Gracias