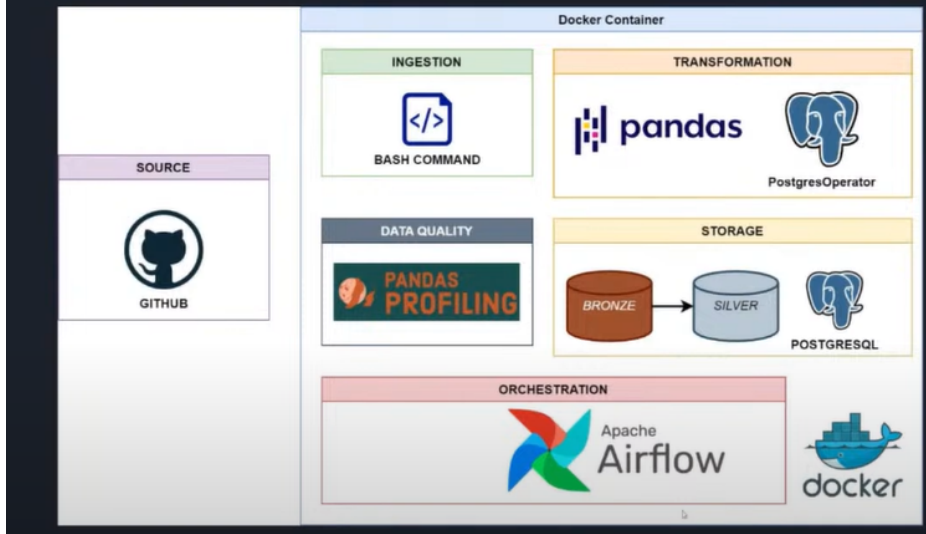


## Laboratorio:



A continuación se puede ver el archivo `docker-compose.yaml`, en el cual establecemos la configuración de Postgres.

The screenshot shows a code editor with the `docker-compose.yaml` file open. The file defines the services for the data pipeline, including the Airflow scheduler and workers, and the PostgreSQL database. The `environment` section is configured with various Airflow settings.

Iniciamos con `docker-compose`, apoyándonos en el comando **up airflow-init**.


```
jhon.edwar.b.salazar@C11-SFI4XGGFG32 MINGW64 ~/Documents/data-pipeline
$ docker-compose up airflow-init
time="2023-09-14T10:46:23-05:00" level=warning msg="The \"AIRFLOW_UID\" variable is not set. Defaulting to a blank string."
time="2023-09-14T10:46:23-05:00" level=warning msg="The \"AIRFLOW_UID\" variable is not set. Defaulting to a blank string."
[+] Running 20/15
- postgres 13 layers [#####] 54.64MB/102MB Pulling
- airflow-init 20 layers [#####] 24.54MB/103.3MB Pulling
```

Luego iniciamos los containers.

```
jhon.edwar.b.salazar@C11-SFI4XGGFG32 MINGW64 ~/Documents/data-pipeline
$ docker-compose up -d
time="2023-09-14T10:52:07-05:00" level=warning msg="The \"AIRFLOW_UID\" variable is not set. Defaulting to a blank string."
time="2023-09-14T10:52:07-05:00" level=warning msg="The \"AIRFLOW_UID\" variable is not set. Defaulting to a blank string."
[+] Running 5/5
✓ Container data-pipeline-postgres-1 Healthy 0.0s
✓ Container data-pipeline-airflow-init-1 Exited 0.0s
✓ Container data-pipeline-airflow-scheduler-1 Started 0.2s
✓ Container data-pipeline-airflow-triggerer-1 Started 0.2s
✓ Container data-pipeline-airflow-webserver-1 Started 0.2s

jhon.edwar.b.salazar@C11-SFI4XGGFG32 MINGW64 ~/Documents/data-pipeline
$ docker ps
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS                   PORTS                               NAMES
c319e76f05a7   apache/airflow:2.7.1 "/usr/bin/dumb-init ..." 54 seconds ago Up 21 seconds (health: starting) 0.0.0.0:8080->8080/tcp data-pipeline-airflow-webserver-1
46696f4e7f38   apache/airflow:2.7.1 "/usr/bin/dumb-init ..." 54 seconds ago Up 21 seconds (health: starting) 8080/tcp data-pipeline-airflow-scheduler-1
3a16f06b0669   apache/airflow:2.7.1 "/usr/bin/dumb-init ..." 54 seconds ago Up 21 seconds (health: starting) 8080/tcp data-pipeline-airflow-triggerer-1
10bfc896cd1f   postgres:13         "docker-entrypoint.s..." 4 minutes ago Up 4 minutes (healthy) 0.0.0.0:5432->5432/tcp data-pipeline-postgres-1
```

Airflow permite realizar conexiones a diferentes servicios de la nube, para los distintos proveedores. En este caso debemos establecer la conexión con Postgres.

 Airflow

DAGs

Security

Browse

Admin

Docs

Edit Connection

Conn Id \*

postgres\_docker

Conn Type \*

Postgres

Conn Type missing? Make sure you've installed the corresponding Airflow Provider Package.

Description

conexion a docker

Host

host.docker.internal

Schema

postgres

Login

airflow

Password

Port

5432

Utilizamos docker **build** para construir la imagen.



























```
jhon.edwar.b.salazar@C11-SFI4XGGFG32 MINGW64 ~/Documents/data-pipeline
$ docker build . --tag extending_airflow:latest202309
[+] Building 96.5s (9/9) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile              0.0s
=> => transferring dockerfile: 203B                               0.0s
=> [internal] load .dockerignore                                  0.0s
=> => transferring context: 2B                                       0.0s
=> [internal] load metadata for docker.io/apache/airflow:2.7.1  0.0s
=> [internal] load build context                                  0.0s
=> => transferring context: 76B                                       0.0s
=> CACHED [1/4] FROM docker.io/apache/airflow:2.7.1            0.0s
=> [2/4] COPY requirements.txt /requirements.txt                 0.0s
=> [3/4] RUN pip install --user --upgrade pip                   17.4s
=> [4/4] RUN pip install --no-cache-dir --user -r /requirements.txt 74.2s
=> exporting to image                                             4.6s
=> => exporting layers                                              4.6s
=> => writing image sha256:bf31976761eded52293e9153ae44fa479a6d68b82dfe10dd023bda436fcaa 0.0s
=> => naming to docker.io/library/extending_airflow:latest202309 0.0s
```

Se corre nuevamente el *webserver* y *scheduler* con la nueva versión `extending_airflow:latest202309`

```
$ docker-compose up -d --no-deps --build airflow-webserver airflow-scheduler
time="2023-09-14T11:27:49-05:00" level=warning msg="The \"AIRFLOW_UID\" variable is not set. Defaulting to a blank string."
time="2023-09-14T11:27:49-05:00" level=warning msg="The \"AIRFLOW_UID\" variable is not set. Defaulting to a blank string."
[+] Running 2/2
 ✓ Container data-pipeline-airflow-scheduler-1 Started                    5.8s
 ✓ Container data-pipeline-airflow-webserver-1 Started                   12.0s

jhon.edwar.b.salazar@C11-SFI4XGGFG32 MINGW64 ~/Documents/data-pipeline
$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS                PORTS                               NAMES
868731023c14   extending_airflow:latest202309      "/usr/bin/dumb-init ..." 42 seconds ago Up 29 seconds (health: starting) 8080/tcp      data-pipeline-airflow-scheduler-1
39fe635e605d   extending_airflow:latest202309      "/usr/bin/dumb-init ..." 42 seconds ago Up 29 seconds (health: starting) 0.0.0.0:8080->8080/tcp data-pipeline-airflow-webserver-1
3a16f06b0669   apache/airflow:2.7.1                "/usr/bin/dumb-init ..." 36 minutes ago Up 35 minutes (healthy) 8080/tcp      data-pipeline-airflow-triggerer-1
10bfc896cd1f   postgres:13                          "docker-entrypoint.s..." 39 minutes ago Up 39 minutes (healthy) 0.0.0.0:5432->5432/tcp data-pipeline-postgres-1
```

Ya se pueden ver las imágenes actualizadas en nuestro **Docker Desktop**.

	Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
	data-pipeline		Running (4/5)	57.19%		5 minutes ago	  
	postgres-1	postgres:13	Running	2.16%	5432:5432 	44 minutes ago	  
	airflow-init-1	apache/airflow:2.7.1	Exited	0%		41 minutes ago	  
	airflow-triggerer-1	apache/airflow:2.7.1	Running	51.35%		41 minutes ago	  
	airflow-webserver-1	extending_airflow:latest202309	Running	0.24%	8080:8080 	5 minutes ago	  
	airflow-scheduler-1	extending_airflow:latest202309	Running	3.44%		5 minutes ago	  

Ingresar a la terminal del scheduler, con la ayuda del **Container ID**.

```
jhon.edwar.b.salazar@C11-SFI4XGGFG32 MINGW64 ~/Documents/data-pipeline
$ docker exec -it 868731023c14 bash
airflow@868731023c14:/opt/airflow$
```

Realizamos el test a la tarea relacionada con descargar el archivo csv. El mensaje “Marking task as SUCCESS” nos deja saber que se ejecutó correctamente.

```
airflow@868731023c14:/opt/airflow/dags$ airflow tasks test dag_webinar descargar_csv 2023-01-01
[2023-09-14T17:08:21.578+0000] {dagbag.py:530} INFO - Filling up the DagBag from /opt/***/dags
nstance: dag_webinar.descargar_csv ***temporary_run 2023-09-14T17:02:32.938803+00:00__ [None]>
[2023-09-14T17:08:26.295+0000] {taskinstance.py:1157} INFO - Dependencies all met for dep_context=queueable deps ti=<TaskInstance: dag_webinar.descargar_csv ***tempor
ary_run 2023-09-14T17:02:32.938803+00:00__ [None]>
[2023-09-14T17:08:26.296+0000] {taskinstance.py:1359} INFO - Starting attempt 1 of 3
[2023-09-14T17:08:26.296+0000] {taskinstance.py:1428} WARNING - cannot record queued_duration for task descargar_csv because previous state change time has not been saved
[2023-09-14T17:08:26.299+0000] {taskinstance.py:1380} INFO - Executing <Task(BashOperator): descargar_csv> on 2023-01-01 00:00:00+00:00
[2023-09-14T17:08:26.761+0000] {taskinstance.py:1660} INFO - Exporting env vars: AIRFLOW_CTX_DAG_OWNER='jhon' AIRFLOW_CTX_DAG_ID='dag_webinar' AIRFLOW_CTX_TASK_ID='descarg
ar_csv' AIRFLOW_CTX_EXECUTION_DATE='2023-01-01T00:00:00+00:00' AIRFLOW_CTX_TRY_NUMBER='1' AIRFLOW_CTX_DAG_RUN_ID='***temporary_run 2023-09-14T17:02:32.938803+00:00__'
[2023-09-14T17:08:26.764+0000] {subprocess.py:63} INFO - Tmp dir root location: /tmp
[2023-09-14T17:08:26.765+0000] {subprocess.py:75} INFO - Running command: ['/bin/bash', '-c', 'curl -k -o /opt/***/dags/data/titanic.csv https://raw.githubusercontent.com/
ronnygang/webinar/main/titanic.csv']
[2023-09-14T17:08:26.779+0000] {subprocess.py:86} INFO - Output:
[2023-09-14T17:08:26.787+0000] {subprocess.py:93} INFO - % Total % Received % Xferd Average Speed Time Time Time Current
[2023-09-14T17:08:26.788+0000] {subprocess.py:93} INFO - Dload Upload Total Spent Left Speed
100 60301 0 60301 0 0 151k 0 --:--:-- --:--:-- --:--:-- 151k
[2023-09-14T17:08:27.184+0000] {subprocess.py:97} INFO - Command exited with return code 0
[2023-09-14T17:08:27.211+0000] {taskinstance.py:1398} INFO - Marking task as SUCCESS. dag_id=dag_webinar, task_id=descargar_csv, execution_date=20230101T000000, start_date
=, end_date=20230914T170827
```

Se genera el reporte de calidad de datos (profiling) al ejecutar la tarea correspondiente.

Data Quality Report

OverviewVariablesInteractionsCorrelationsMissing valuesSample

Overview

OverviewAlerts16Reproduction

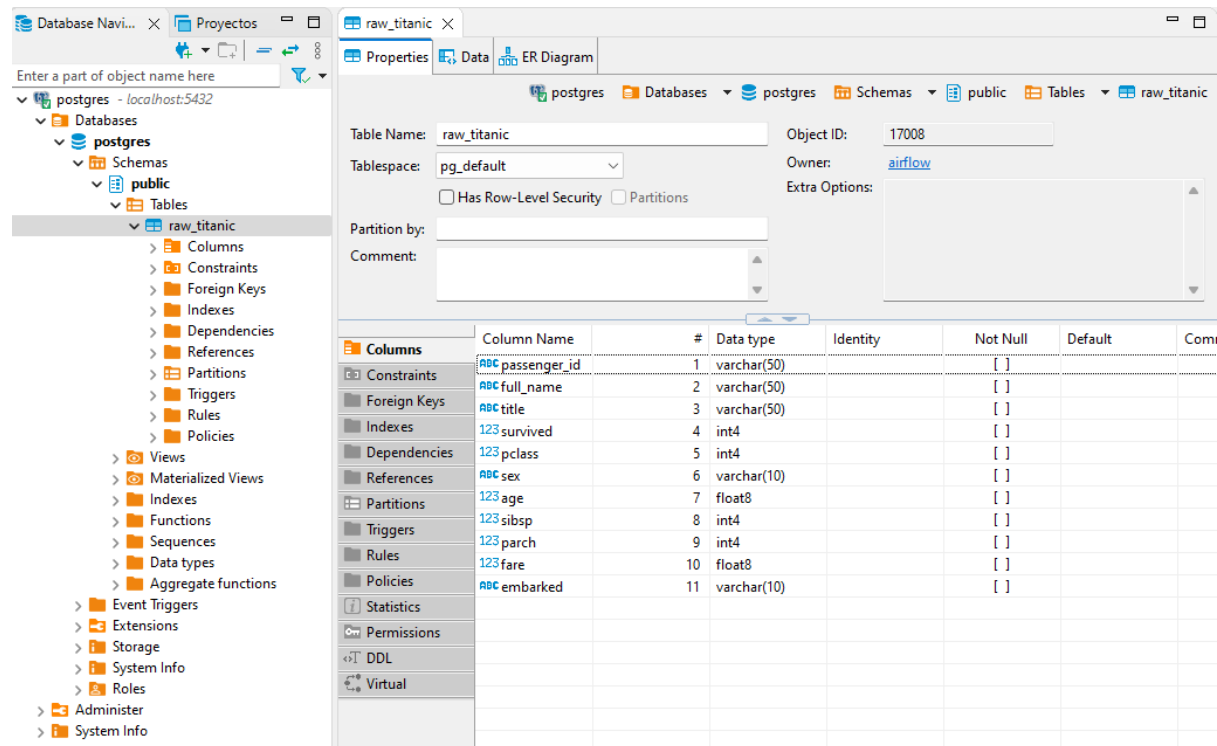
Dataset statistics

Number of variables	12
Number of observations	891
Missing cells	866
Missing cells (%)	8.1%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	83.7 KiB
Average record size in memory	96.1 B

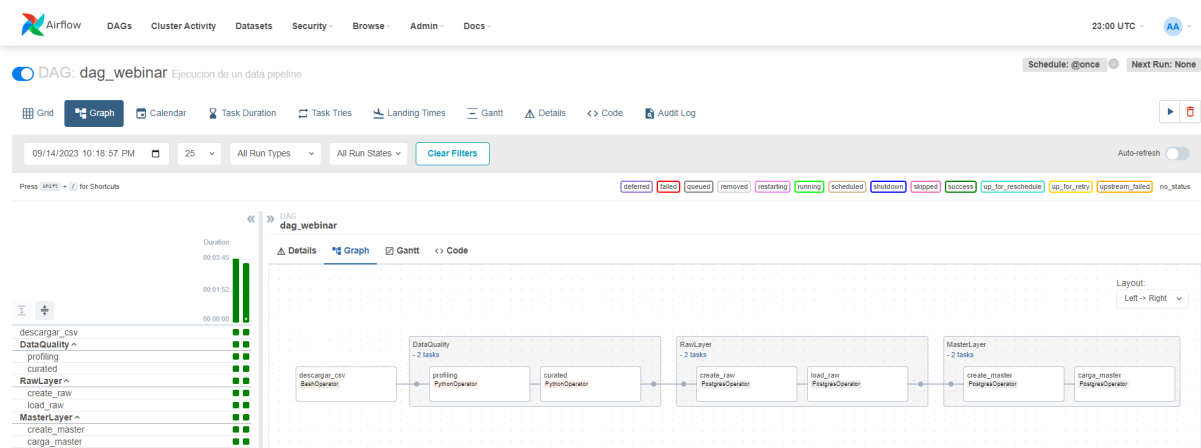
Variable types

Numeric	5
Categorical	7

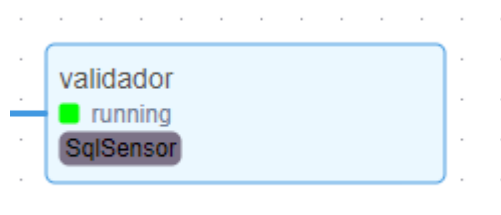
Luego de hacer una limpieza de datos a través de nuestra task *curated*, se realiza la conexión con Postgres.



Se crea una tabla de BD a través de una task del workflow y se cargan los datos con ayuda del archivo sql creado anteriormente. El workflow queda de la siguiente manera.



Luego de correr el workflow, seleccionamos la opción de *clear task* del validador que realiza un conteo de los registros del día.



GCP Composer - Revisar servicio ya que es económico en cuanto a oferta Cloud (aprox. \$2USD por día)