Ejercicio Clase 9

Consigna: Por cada ejercicio, escribir el código y agregar una captura de pantalla del resultado obtenido.

1. Crear una base de datos en Hive llamada northwind_analytics

2. Crear un script para importar un archivo .parquet de la base northwind que contenga la lista de clientes junto a la cantidad de productos vendidos ordenados de mayor a menor (campos customer_id, company_name, productos_vendidos). Luego ingestar el archivo a HDFS (carpeta /sqoop/ingest/clientes). Pasar la password en un archivo

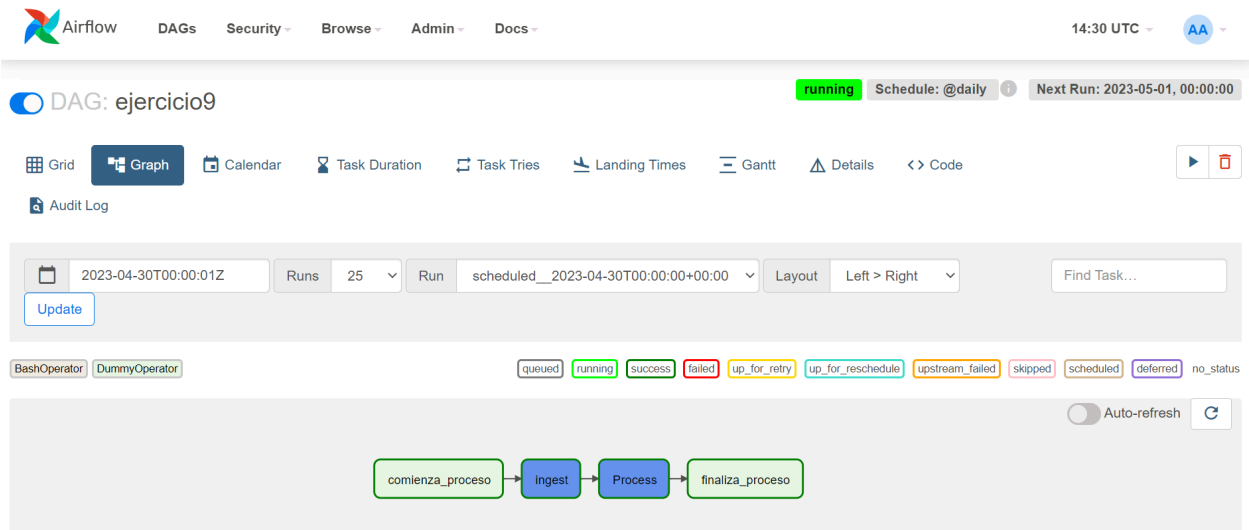
3. Crear un script para importar un archivo .parquet de la base northwind que contenga la lista de órdenes junto a qué empresa realizó cada pedido (campos order_id, shipped_date, company_name, phone). Luego ingestar el archivo a HDFS (carpeta /sqoop/ingest/envíos). Pasar la password en un archivo

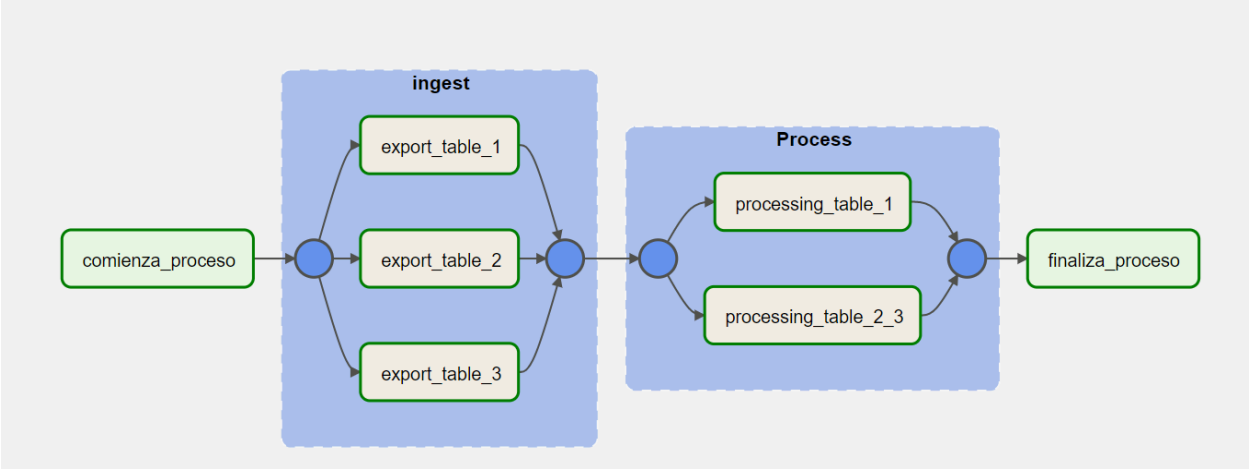
4. Crear un script para importar un archivo .parquet de la base northwind que contenga la lista de detalles de órdenes (campos order_id, unit_price, quantity, discount). Luego ingestar el archivo a HDFS (carpeta /sqoop/ingest/order_details). Pasar la password en un archivo

5. Generar un archivo .py que permita mediante Spark insertar en hive en la db northwind_analytics en la tabla products_sold, los datos del punto 2, pero solamente aquellas compañías en las que la cantidad de productos vendidos fue mayor al promedio.

6. Generar un archivo .py que permita mediante Spark insertar en hive en la tabla products_sent, los datos del punto 3 y 4, de manera tal que se vean las columnas order_id, shipped_date, company_name, phone, unit_price_discount (unit_price with discount), quantity, total_price (unit_price_discount * quantity). Solo de aquellos pedidos que hayan tenido descuento.

7. Realizar un proceso automático en Airflow que orqueste los pipelines creados en los puntos anteriores. Crear un grupo para la etapa de ingest y otro para la etapa de process. Correrlo y mostrar una captura de pantalla (del DAG y del resultado en la base de datos)


Airflow
DAGs
Security
Browse
Admin
Docs
14:30 UTC
AA
DAG: ejercicio9
running
Schedule: @daily
Next Run: 2023-05-01, 00:00:00
Grid
Graph
Calendar
Task Duration
Task Tries
Landing Times
Gantt
Details
Code
Audit Log
2023-04-30T00:00:01Z
Runs
25
Run
scheduled__2023-04-30T00:00:00+00:00
Layout
Left > Right
Find Task...
Update
BashOperator
DummyOperator
queued
running
success
failed
up_for_retry
up_for_reschedule
upstream_failed
skipped
scheduled
deferred
no_status
Auto-refresh
comienza_proceso
ingest
Process
finaliza_proceso


ingest
export_table_1
export_table_2
export_table_3
Process
processing_table_1
processing_table_2_3
comienza_proceso
finaliza_proceso