CSC12107 – HTTT PHỤC VỤ TRÍ TUỆ KINH DOANH

Apache Airflow

Seminar môn học 2024 – 2025

21HTTT2 – Trường ĐH Khoa học Tự nhiên - VNUHCM

GV hướng dẫn: ThS. Nguyễn Ngọc Minh Châu, ThS. Tiết Gia Hồng, ThS. Hồ Thị Hoàng Vy

Nhóm 11

✦

TP. Hồ Chí Minh, Tháng 11 năm 2024

Mục lục

Contents

Thông tin nhóm và phân công

- Demo: https://youtu.be/-q_QpyrBaQo

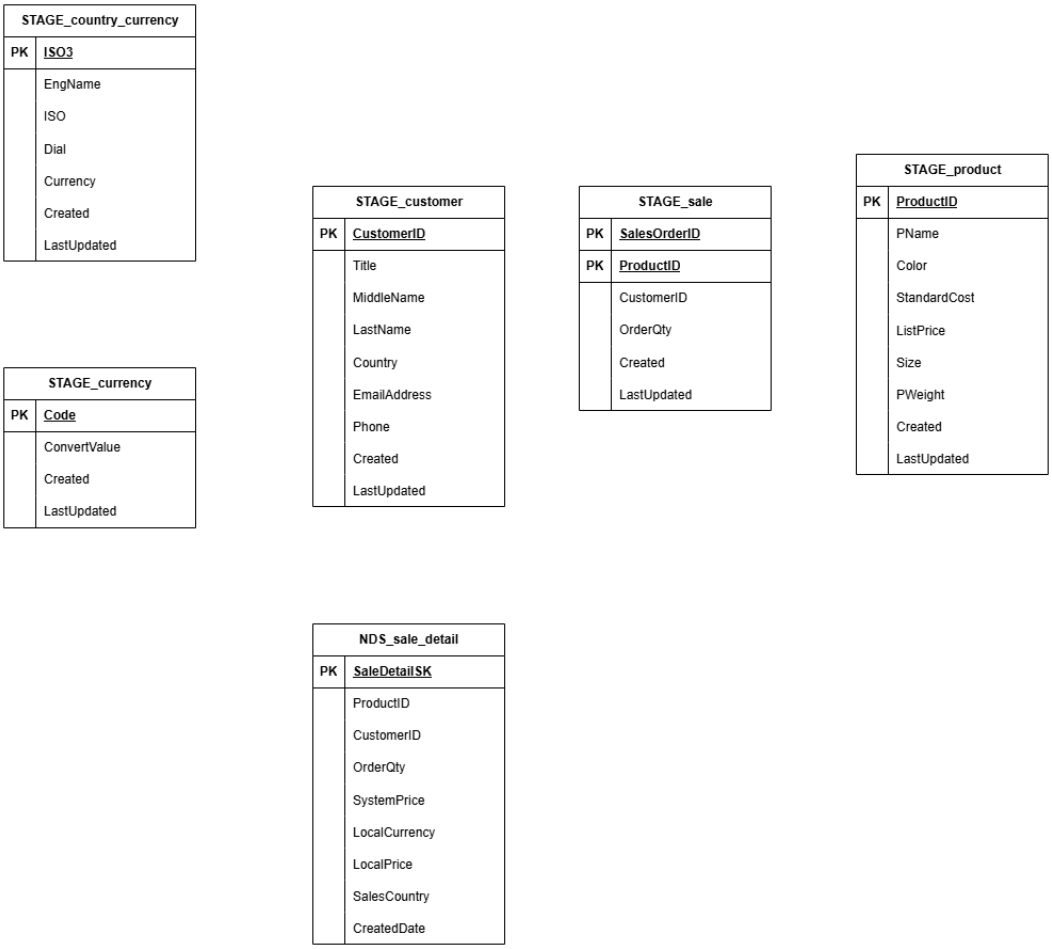
MSSV	Họ tên	Công việc	(%)
21127004	Trần Nguyễn An Phong	*Airflow và ETL:* - Phát sinh các data source dạng csv. - Transform và load các dữ liệu theo kịch bản. *Báo cáo và demo:* - Mục 3 – Hoạt động. - Quay video demo.	100
21127135	Diệp Hữu Phúc	*Airflow và ETL:* - Cài đặt môi trường airflow. - Trích xuất dữ liệu từ nguồn csv. *Báo cáo và demo:* - Mục 2 – Cài đặt.	100
21127296	Đặng Hà Huy	*Airflow và ETL:* - Khởi tạo các database và bảng cần thiết. - Lưu logging cho các dòng lỗi theo kịch bản. *Báo cáo và demo:* - Thiết kế slide powerpoint.	100
21127385	Phạm Uyển Nhi	*Airflow và ETL:* - Cài đặt giao thức trích xuất API ngoại tệ. - Trích xuất dữ liệu và transform từ nguồn API. *Báo cáo và demo:* - Mục 1 – Tổng quan.	100

1 Tổng quan

Apache Airflow (gọi tắt là Airflow) là một công cụ mã nguồn mở được sử dụng để **lập lịch**, **quản lý**, và **giám sát** các **quy trình xử lý dữ liệu**. Nó được sử dụng rộng rãi trong các hệ thống xử lý dữ liệu lớn để **tự động hóa** các quy trình xử lý dữ liệu phức tạp.

Với trọng tâm là **xây dựng** một **quy trình ETL với Airflow**, nhóm đã đề xuất một kịch bản đơn giản tuân theo cấu trúc ETL tự động, bao gồm ba khâu chính là **Extract & Transform**, **Load** và kèm theo là **Logging & DQ**.

Đồng thời, các **dữ liệu** phục vụ kịch bản này cũng được **chuẩn bị thủ công**, hoặc **tổng hợp từ nhiều nguồn**. Trong đó, dữ liệu của **ba khâu chính** sẽ được lưu trữ bằng **SQL Server** với tên database là **[ap_airflow]**. Còn dữ liệu của **Logging** sẽ được lưu trữ bằng **Postgres** với tên database là **logging** và tên schema là **public**.



Hình 1a. Lược đồ cơ sở dữ liệu của kịch bản.

1.1 Extract & Transform

Dữ liệu đầu vào gồm có **2 nguồn**,

- Từ **file csv** – Các **dữ liệu mô phỏng** được tạo thành các file với định dạng csv, dữ liệu gồm các loại như thông tin khách hàng, thông tin tiền tệ, thông tin sản phẩm, thông tin đơn hàng,...
- Từ **nguồn API trực tuyến** – Các dữ liệu về **tỉ giá quy đổi ngoại tệ** mới nhất được trích xuất từ API trực tuyến.

Các dữ liệu này sẽ được **biến đổi** và **đính thêm** một số **thông tin nhãn** trước khi được lưu vào cơ sở dữ liệu.

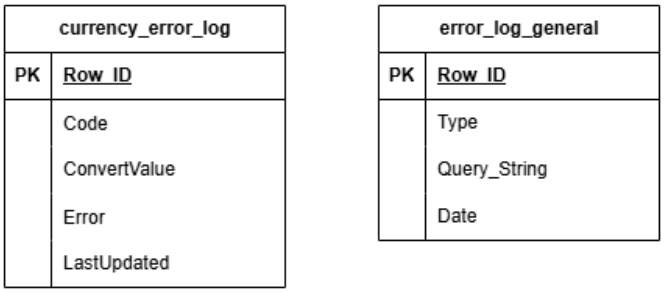
1.2 Load

Tiếp theo, dữ liệu được **kết hợp với nhau** để tạo thành một **bảng** chứa thông tin cụ thể hơn về **tổng giá tiền của một đơn bán** (sale) với **đơn vị tiền tệ hệ thống** (mặc định là “USD”).

Đồng thời, lưu trữ cả **quốc gia sinh sống của khách hàng** và **giá trị tiền tệ của đơn hàng** tại quốc gia đó.

1.3 Logging

Nếu xảy ra **vấn đề** khiến hệ thống **không thể ghi** được một dòng dữ liệu nào đó tại **giai đoạn trích xuất tỉ giá ngoại tệ**, hay tại giai đoạn từ **source vào stage** của csv thì thông tin của **dòng gặp lỗi** và **thông báo lỗi chi tiết** của hệ thống sẽ được **lưu trữ** lại để có thể kiểm tra sau này.



Hình 13a. Lược đồ cơ sở dữ liệu logging.

1.4 Data Quality Rule

Ngoài ra nhóm cũng áp dụng phương pháp **data quality metadata** để kiểm soát các references lỗi trong giai đoạn tạo **sale_detail.**

Để phục vụ cho kiểm thử, nhóm đã thêm vào kịch bản một **DQ rule** rằng **hệ thống hiện tại** chỉ đang **vận hành chính thức** ở ba quốc gia là "**USA**", "**CAN**" và "**AUS**". Vì vậy, những dòng dữ liệu ở giai đoạn Load **không thuộc** về ba quốc gia này sẽ được coi là **dòng gặp lỗi**.

2 Cài đặt

Project được tổ chức với các thành phần sau,

- **dags:** Folder chứa source code xây dựng các quy trình của Airflow.
  - **initilize_general.py** – Khởi tạo các database và bảng dữ liệu cho lần chạy đầu tiên.
  - **Initialize_metadata.py** – Khởi tạo các bảng metadata liệu cho lần chạy đầu tiên.
  - **csv_source.py** – Nạp dữ liệu từ nguồn là các file csv, được thiết lập để chạy mỗi 8h hàng ngày.
  - **api_source.py** – Nạp dữ liệu từ nguồn là API trực tuyến, được thiết lập để chạy mỗi 6h hàng ngày.
  - **creating_sale_detail.py** – Trích lọc và kết hợp dữ liệu đã có thành các thông tin tài chính, được thiết lập để chạy mỗi 12h hàng ngày.

- **source_data:** Folder chứa các file data là source dạng csv của mô hình.
  - **country_currency.csv** – Chứa thông tin về đơn vị tiền tệ của một quốc gia.
  - **customer.csv** – Chứa thông tin cá nhân của một khách hàng.
  - **product.csv** – Chứa thông tin về một sản phẩm.
  - **sale.csv** – Chứa thông tin về một đơn bán.
- Và còn lại là các file thư viện không liên quan trực tiếp tới nội dung của bài tập.

2.1 Cài đặt Docker và Astronomer

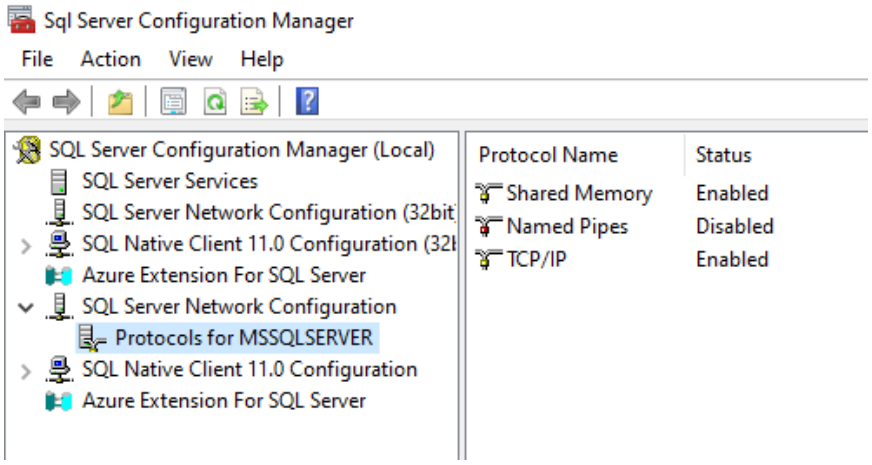
Apache Airflow và các thành phần phụ thuộc đều sẽ hoạt động trên nền tảng **Docker**, vì vậy Docker là một trong những thứ đầu tiền cần tải. Hãy đảm bảo Docker được cài đặt kèm theo tính năng **Hyper-V** (hoặc **Virtualization**) được bật.

Astronomer (được gọi tắt là **astro**) là một trong những **CLI** phổ biến nhất để làm việc với Airflow, astro giúp việc **khởi tạo**, **cài đặt** và **cập nhật** một dự án Airflow đơn giản nhất có thể. Hãy cài đặt astro theo hướng dẫn sau trước khi bắt đầu các phần tiếp theo,

- https://www.astronomer.io/docs/astro/cli/install-cli?tab=windows#install-the-astro-cli

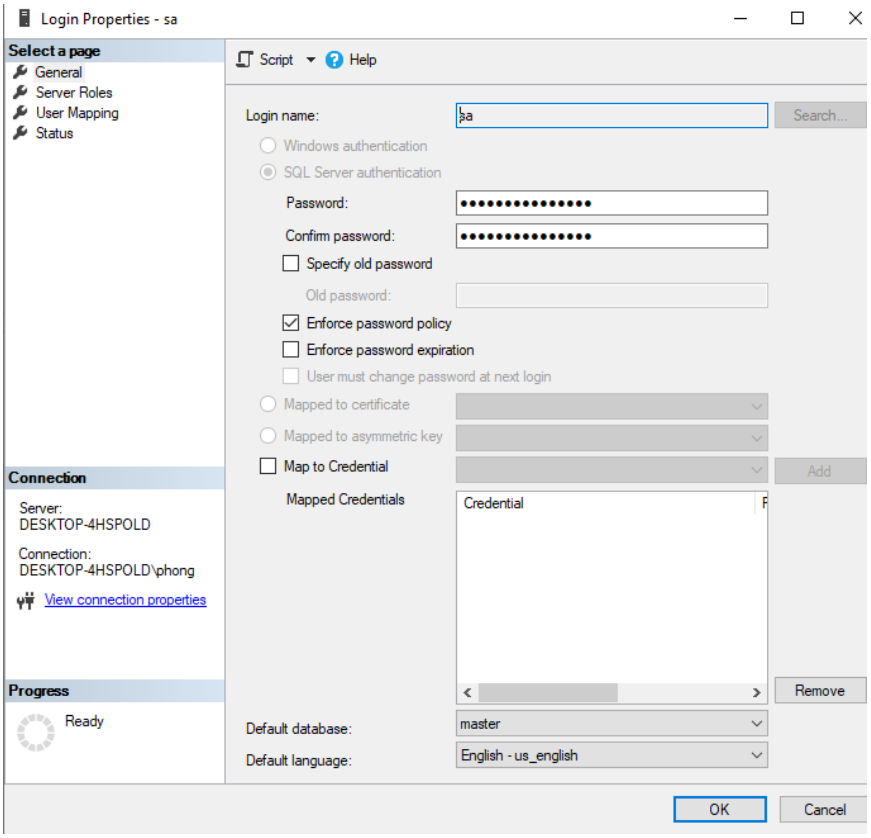
2.2 Kích hoạt giao thức TCP/IP cho SQL Server

Kết nối giữa Airflow và SQL Server có giao thức **TCP/IP**, vì vậy chúng ta cần cho phép thực hiện giao thức này thông qua **SQL Server Configuration Manager**.



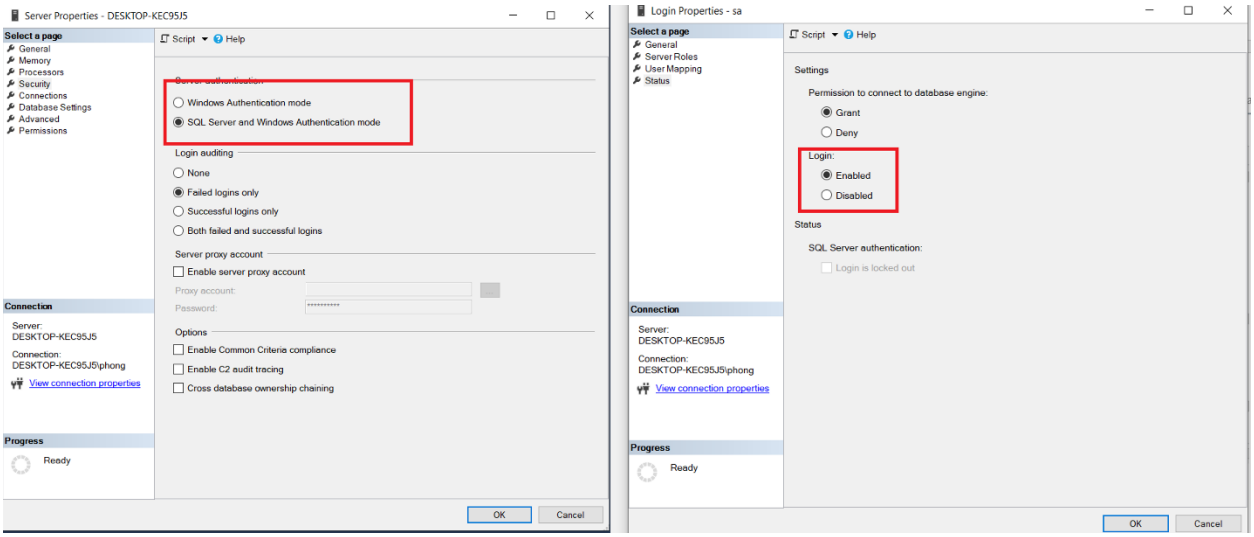
2.3 Đổi mật khẩu cho tài khoản "sa" của SQL Server

Phục vụ cho nhu cầu **kết nối**, **tạo bảng**, và **thêm dữ liệu** vào **SQL Server** thì Airflow cần kết nối bằng **tài khoản có một số quyền** nhất định. Do đó, nhóm chọn tài khoản "**sa**" có sẵn trong hệ thống, với mật khẩu được đổi thành "**12345**".



2.4 Cho phép giao thức kết nối bằng tài khoản SQL Server

Sử dụng **SSMS** để cấu hình cho phép **kết nối** bằng **tài khoản SQL Server** vào server tổng thể, và vào tài khoản "**sa**".



Hình 24a. (Trái) Lựa chọn dùng tài khoản SQL Server để kết nối vào server. (Phải) Kích hoạt Login vào "sa" cũng dùng SQL Server.

3 Hoạt động

Sau khi đã hoàn thành xong bước thiết lập, ta có thể kiểm tra hoạt động của hệ thống.

3.1 Khởi tạo một dự án Airflow

Việc khởi tạo một dự án airflow đã được tinh giản với astro, giờ đây chúng ta chỉ cần chạy một câu lệnh bằng **command prompt** tại **thư mục chứa nội dung dự án**.

```
astro dev init
```

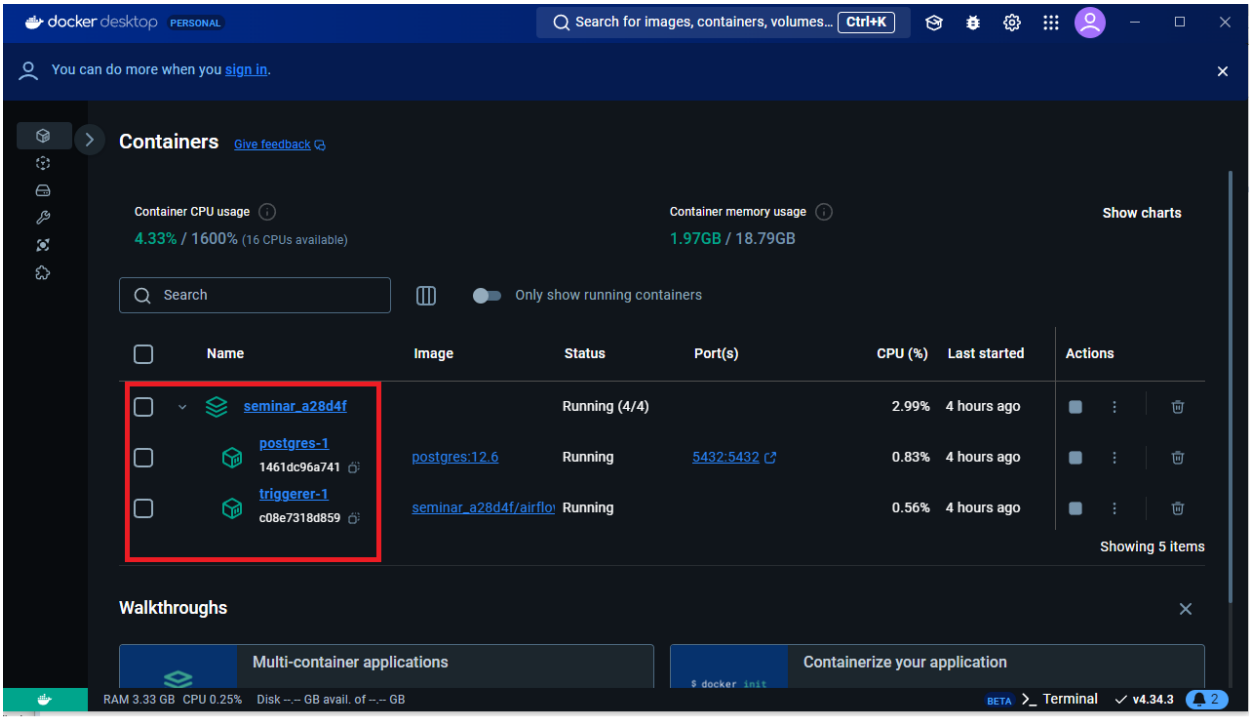
Astro sẽ thực hiện cài đặt Airflow và các container cần thiết ở Docker bao gồm,

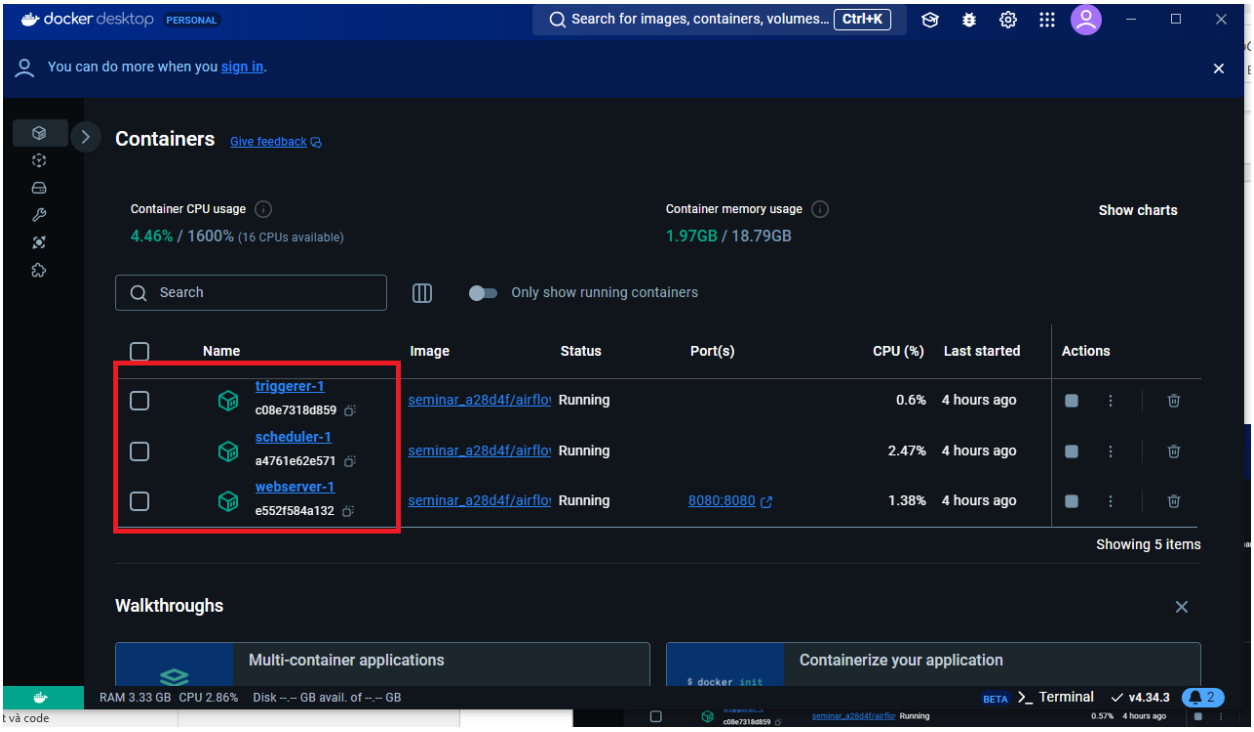
- **postgres** – Database lưu trữ các dữ liệu và hoạt động của Airflow.
- **triggerer** – Những trigger cho các dags của Airflow.
- **scheduler** – Bộ lập lịch cho hoạt động của Airflow.
- **webserver** – Server để tương tác và giao tiếp với Airflow thông qua nền web.

3.2 Khởi động Airflow

Sau quá trình khởi tạo dự án, tiếp theo ta khởi động **Airflow** và các **Docker container** liên quan qua câu lệnh,

```
astro dev start
```



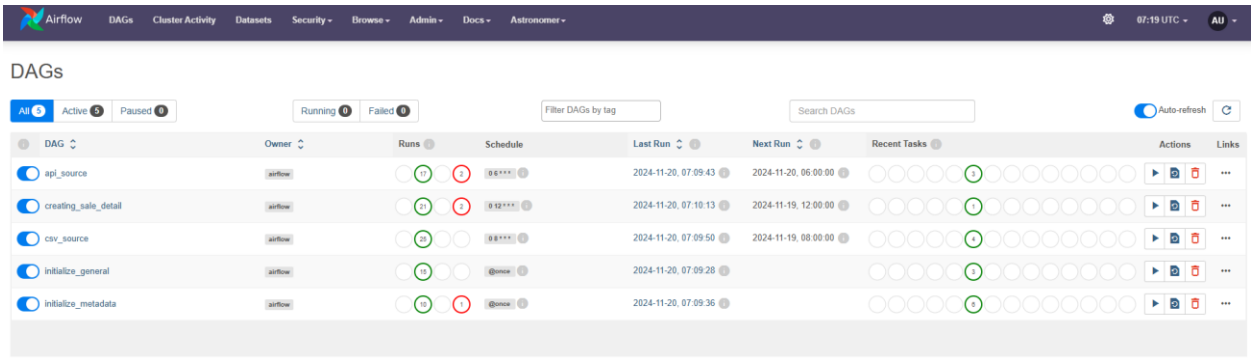


3.3 Truy cập UI của Airflow

Sau khi khởi động, ta có thể truy cập **UI** của Airflow thông qua **địa chỉ mặc định** là,

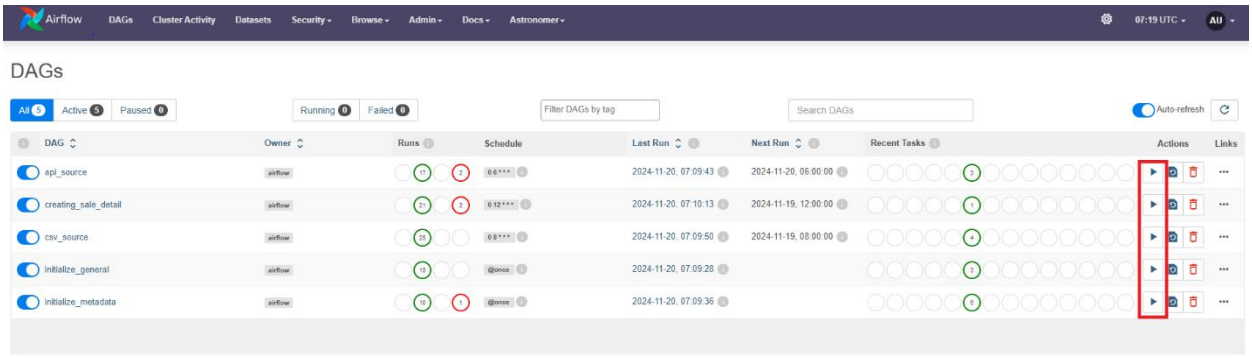
- http://localhost:8080/home

Trang chủ của Airflow chứa các thông tin về **dags** của dự án và **trạng thái hiện tại** của chúng, cũng như thông tin về **thời gian chạy** và **tình trạng các lần chạy**.



3.4 Chạy thử kịch bản đề ra

Để chạy thử kịch bản đề ra, ta có thể **thực thi** các **dag** thủ công bằng cách **trigger** từng dag tại mục **Actions.**



Theo như kịch bản, các dag sẽ được **trigger lần lượt** theo thứ tự sau,

1. initialize_general: Được thực thi duy nhất 1 lần dành cho việc khởi tạo DB.
2. initialize_metadata: Được thực thi duy nhất 1 lần dành cho việc khởi tạo metadata.
3. api_source.
4. csv_source.
5. creating_sale_detail.

Sau khi dags kết thúc thực thi, có thể trích xuất **thông tin** cụ thể về **các đơn hàng** được **nạp thành công** vào **SQL Server** như **Hình 34a**. Bên cạnh đó, các lỗi (được nêu trên kịch bản) nếu có xảy ra sẽ được lưu trữ lại ở Postgres, **Hình 34b**. Và các dòng dữ liệu vi phạm DQ rule cũng sẽ được bắt lại như hình **34c**.

SQLQuery6.sql - lo...EC95J5\phong (72)) SQLQuery5.sql - lo...EC95J5\phong (66)) SQLQuery1.sql - lo...EC95

```
SELECT TOP (1000) [SalesOrderID]
      ,[ProductID]
      ,[CustomerID]
      ,[OrderQty]
      ,[SystemPrice]
      ,[LocalCurrency]
      ,[LocalPrice]
      ,[SalesCountry]
      ,[LastUpdated]
  FROM [ap_airflow].[dbo].[sale_detail]
```

100 %

Results Messages

	SalesOrderID	ProductID	CustomerID	OrderQty	SystemPrice	LocalCurrency	LocalPrice	SalesCountry	LastUpdated
1	71774	822	30072	1	594.83	AUD	894.327029854817	AUS	2024-10-20
2	71774	836	30072	1	594.83	AUD	894.327029854817	AUS	2024-10-20
3	71780	743	29957	1	1349.6	AUD	2029.12388328104	AUS	2024-10-20
4	71780	748	29957	2	2729	AUD	4103.0520728171	AUS	2024-10-20
5	71780	780	29957	4	9279.96	AUD	13952.4218078636	AUS	2024-10-20
6	71780	782	29957	4	9179.96	AUD	13802.0717868736	AUS	2024-10-20
7	71780	783	29957	5	11474.95	AUD	17252.589733592	AUS	2024-10-20
8	71780	809	29957	3	185.76	AUD	279.290198991024	AUS	2024-10-20
9	71780	810	29957	1	120.27	AUD	180.825970244673	AUS	2024-10-20
10	71780	880	29957	1	54.99	AUD	82.677476542401	AUS	2024-10-20
11	71780	905	29957	4	1456.36	AUD	2189.63756568996	AUS	2024-10-20
12	71780	918	29957	2	528.1	AUD	793.99846084819	AUS	2024-10-20
13	71780	925	29957	1	249.79	AUD	375.559317430921	AUS	2024-10-20
14	71780	926	29957	1	249.79	AUD	375.559317430921	AUS	2024-10-20
15	71780	935	29957	2	80.98	AUD	121.753446997702	AUS	2024-10-20
16	71780	937	29957	1	80.99	AUD	121.768481999801	AUS	2024-10-20
17	71780	981	29957	2	1538.98	AUD	2313.8567530319	AUS	2024-10-20
18	71780	982	29957	3	2308.47	AUD	3470.78512954785	AUS	2024-10-20

Hình 34a. Kết quả query cho các đơn hàng được nạp thành công.

error_log_general | Enter a SQL expression to filter results (use Ctrl+Space)

Grid Text Record

	type	query_string	Date
1	country_currency	INSERT INTO S_country_currency ¶	2024-11-10
2	country_currency	INSERT INTO S_country_currency ¶	2024-11-10
3	country_currency	INSERT INTO S_country_currency ¶	2024-11-10
4	country_currency	INSERT INTO S_country_currency ¶	2024-11-10
5	customer	INSERT INTO S_customer ¶	2024-11-10
6	customer	INSERT INTO S_customer ¶	2024-11-10
7	customer	INSERT INTO S_customer ¶	2024-11-10
8	customer	INSERT INTO S_customer ¶	2024-11-10
9	customer	INSERT INTO S_customer ¶	2024-11-10
10	customer	INSERT INTO S_customer ¶	2024-11-10

Hình 34b. Các query lỗi trong giai đoạn source vào stage.

	ID	FLOW_ID	TAB	KEY	RULE	CREATED
1	1	6	STAGE_sale	71776	1	2024-11-20 00:00:00.000
2	2	6	STAGE_product	809	2	2024-11-20 00:00:00.000
3	3	6	STAGE_product	810	2	2024-11-20 00:00:00.000
4	4	6	STAGE_product	867	2	2024-11-20 00:00:00.000
5	5	6	STAGE_product	869	2	2024-11-20 00:00:00.000
6	6	6	STAGE_product	880	2	2024-11-20 00:00:00.000
7	7	6	STAGE_product	708	2	2024-11-20 00:00:00.000
8	8	6	STAGE_product	712	2	2024-11-20 00:00:00.000
9	9	6	STAGE_product	714	2	2024-11-20 00:00:00.000
10	10	6	STAGE_product	715	2	2024-11-20 00:00:00.000
11	11	6	STAGE_product	864	2	2024-11-20 00:00:00.000
12	12	6	STAGE_product	870	2	2024-11-20 00:00:00.000
13	13	6	STAGE_product	876	2	2024-11-20 00:00:00.000
14	14	6	STAGE_product	877	2	2024-11-20 00:00:00.000
15	15	6	STAGE_product	884	2	2024-11-20 00:00:00.000
16	16	6	STAGE_sale	71783	1	2024-11-20 00:00:00.000
17	17	6	STAGE_sale	71783	1	2024-11-20 00:00:00.000
18	18	6	STAGE_sale	71783	1	2024-11-20 00:00:00.000
19	19	6	STAGE_sale	71784	1	2024-11-20 00:00:00.000
20	20	6	STAGE_sale	71785	1	2024-11-20 00:00:00.000

Hình 34c. Các dòng dữ liệu vi phạm DQ rule.

Tham khảo

- **Các slide trong thư mục Seminar** cung cấp bởi ThS. Hồ Thị Hoàng Vy.
- https://www.astronomer.io/docs/astro/cli/get-started-cli
- https://airflow.apache.org/
- https://www.datacamp.com/tutorial/building-an-etl-pipeline-with-airflow
- https://currencyapi.com/