

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG  
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN  
THÔNG**



**BÁO CÁO CUỐI KỲ THỰC TẬP CƠ  
SỞ**

**XÂY DỰNG 1 HỆ THỐNG DATA PIPELINE  
ĐỂ PHÂN TÍCH DỮ LIỆU TỪ GLAMIRA**

**Giảng viên hướng dẫn : Kim Ngọc Bách**

**Họ và tên : Ngô Vũ Minh Quý**

**Mã sinh viên : B22DCVT427**

**Lớp : E22CQCN02-B**

## **Mục Lục**

**1. TỔNG QUAN ĐỀ TÀI**

**2. PHÂN TÍCH ĐỀ TÀI**

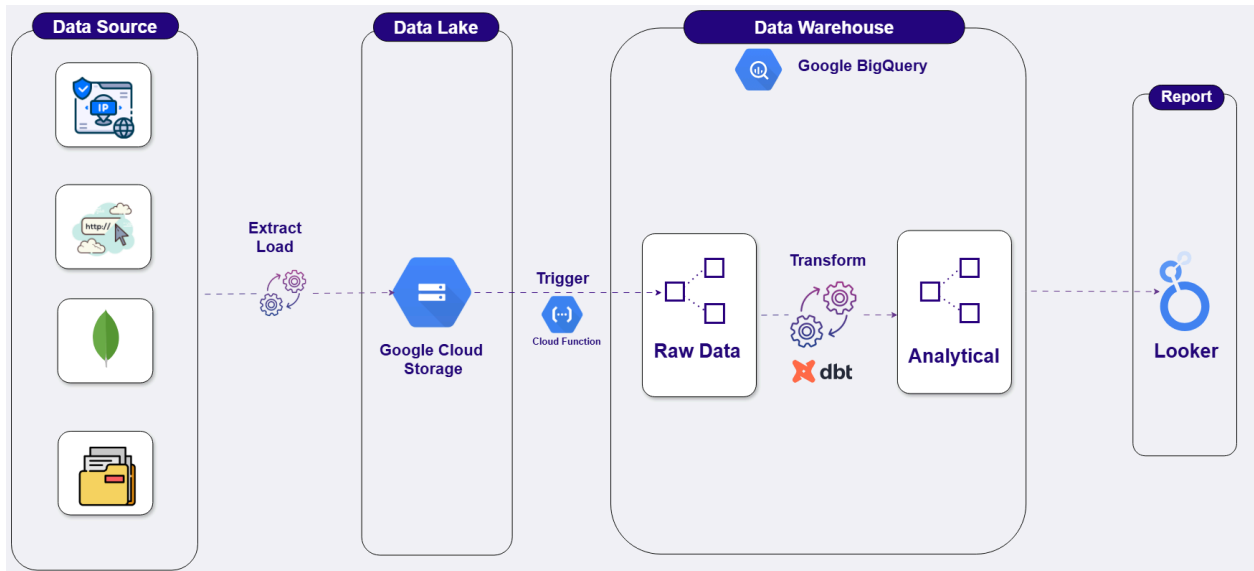
**3. CÔNG NGHỆ SỬ DỤNG**

**4. PHÂN TÍCH YÊU CẦU NGHIỆP VỤ CỦA DATA  
ENGINEER TRONG DỰ ÁN**

**5. THIẾT KẾ VÀ CÀI ĐẶT**

# 1. TỔNG QUAN VỀ ĐỀ TÀI

## 1.1. Giới thiệu chung



Trong thời đại dữ liệu số hiện nay, việc xây dựng một hệ thống thu thập, lưu trữ và xử lý dữ liệu là yếu tố then chốt để khai thác giá trị từ dữ liệu. Đề tài này hướng đến việc xây dựng một nền tảng hạ tầng xử lý dữ liệu hiện đại, tập trung vào khả năng tự động hóa quá trình thu thập và lưu trữ dữ liệu từ nhiều nguồn, sử dụng các công nghệ mã nguồn mở và triển khai trên nền tảng Google Cloud Platform.

Thay vì sử dụng dịch vụ lưu trữ đám mây thương mại như Google Cloud Storage, hệ thống sử dụng MinIO – một giải pháp lưu trữ đối tượng tương thích S3 có thể triển khai tại chỗ hoặc trong môi trường đám mây riêng. Dữ liệu được trích xuất từ MongoDB, xử lý bằng Python và được tải vào hệ quản trị cơ sở dữ liệu phân tích BigQuery để phục vụ mục tiêu phân tích về sau.

## 1.2. Mục tiêu đề tài

- Thiết lập hạ tầng xử lý dữ liệu trên MinIO với các thành phần chính: Virtual Machine, MinIO và Postgres.
- Trích xuất và chuyển đổi dữ liệu từ MongoDB, bao gồm xử lý địa chỉ IP và thu thập tên sản phẩm.

- Tự động hóa quá trình tải dữ liệu từ MinIO sang Postgres
- Kiểm tra chất lượng dữ liệu và lập tài liệu mô tả cấu trúc dữ liệu, phục vụ các bước phân tích tiếp theo.

### 1.3. Phạm vi đề tài

Đề tài được thực hiện trong vòng 5 tuần và bao gồm ba giai đoạn chính. Giai đoạn đầu tập trung vào việc thiết lập môi trường làm việc, bao gồm cấu hình GCP, triển khai MongoDB trên máy ảo (VM), thiết lập MinIO để lưu trữ dữ liệu, và tải dữ liệu thô. Nhóm cũng thực hiện xử lý dữ liệu cơ bản bằng Python như phân tích địa chỉ IP để xác định vị trí địa lý và thu thập tên sản phẩm từ các URL.

Giai đoạn hai là xây dựng pipeline ETL tự động: dữ liệu được trích xuất từ MongoDB, chuyển đổi định dạng phù hợp (CSV, JSONL...), tải lên MinIO và sau đó được Cloud Function kích hoạt để đẩy dữ liệu vào BigQuery. Giai đoạn cuối cùng bao gồm kiểm thử toàn bộ pipeline, giám sát quá trình xử lý, kiểm tra chất lượng dữ liệu và lập tài liệu chi tiết về quy trình và cấu trúc dữ liệu. Đề tài không bao gồm các bước phân tích chuyên sâu hay trực quan hóa dữ liệu bằng các công cụ như dbt hoặc Looker.

## 2. PHÂN TÍCH YÊU CẦU

### 2.1. Yêu cầu chức năng

#### -> Thu thập và lưu trữ dữ liệu

- Cho phép người dùng tải dữ liệu gốc (raw data) từ các nguồn như file CSV/JSON vào hệ thống.
- Hỗ trợ nhập dữ liệu vào MongoDB (trên máy ảo hoặc container).
- Tải dữ liệu từ MongoDB lên **MinIO** dưới định dạng phù hợp (CSV, JSON, hoặc Parquet).
- Duy trì cấu trúc và metadata cần thiết trong quá trình chuyển đổi dữ liệu.

#### -> Tự động hóa xử lý dữ liệu

- Tự động phát hiện file mới được upload lên MinIO.

Khi có file mới, kích hoạt tiến trình ETL để:

- Đọc dữ liệu từ MinIO.
- Làm sạch và chuyển đổi dữ liệu.
- Tải dữ liệu vào PostgreSQL (raw → staging → transformed).
- Ghi log quá trình xử lý và lỗi (nếu có).

-> **Phân tích & mô hình hóa dữ liệu**

- Chuẩn hóa tên sản phẩm dựa trên `product_id` và `current_url`.
- Xử lý thông tin định vị IP (IP to Location) và lưu trữ riêng.
- Sử dụng công cụ như `dash` để mô hình hóa dữ liệu trong PostgreSQL.

-> **Trực quan hóa và báo cáo**

- Xây dựng dashboard trên Looker Studio hoặc công cụ tương đương (ví dụ: Metabase, Superset).
- Dashboard bao gồm các chỉ số: doanh thu, phân bố địa lý, xu hướng thời gian, hiệu suất sản phẩm.
- Cho phép lọc dữ liệu theo thời gian, vị trí, danh mục sản phẩm, v.v.

## 2.2. Yêu cầu phi chức năng (Non-Functional Requirements)

-> **Hiệu năng**

- Hệ thống phải xử lý được hàng nghìn dòng dữ liệu trong mỗi lần ETL mà không gián đoạn.
- Phản hồi nhanh khi truy vấn dữ liệu phân tích.

-> **Tự động hóa & mở rộng**

- Toàn bộ pipeline phải hỗ trợ kích hoạt tự động khi có dữ liệu mới.

- Cấu trúc hệ thống cho phép mở rộng thêm nguồn dữ liệu và dashboard mới.

#### **-> Khả năng ghi nhận và giám sát**

- Ghi log đầy đủ quá trình ETL và các lỗi phát sinh.
- Cung cấp cơ chế giám sát tiến trình pipeline và thông báo khi lỗi.

#### **-> Bảo mật**

- Quản lý quyền truy cập vào MinIO, PostgreSQL, dashboard theo vai trò người dùng.
- Giữ dữ liệu nhạy cảm (ví dụ IP, thông tin khách hàng) ở dạng đã xử lý/an toàn.

#### **-> Tài liệu hóa**

- Mỗi thành phần cần có tài liệu hướng dẫn:
  - Cách thiết lập.
  - Cách chạy pipeline.
  - Ý nghĩa các bảng/mô hình trong PostgreSQL.
  - Hướng dẫn sử dụng dashboard.

## **3. CÔNG NGHỆ VÀ CÔNG CỤ SỬ DỤNG**

### **3.1. Ngôn ngữ lập trình**

- Python
  - Dùng cho xử lý dữ liệu, thao tác với MongoDB, MinIO, PostgreSQL, IP Geolocation, logging và tự động hóa pipeline (ETL).

- **SQL (PostgreSQL dialect)**  
→ Dùng để tạo schema, thao tác dữ liệu, viết truy vấn cho dashboard, và mô hình hóa dữ liệu với dash.

### 3.2. Hệ quản trị cơ sở dữ liệu

- **MongoDB**  
→ Lưu trữ dữ liệu thô ban đầu, xử lý IP, sản phẩm, hành vi người dùng từ log.
- **PostgreSQL (Thay BigQuery)**  
→ Là kho dữ liệu phân tích chính. Bao gồm các bảng raw, staging, dimension và fact table.

### 3.3 Lưu trữ đối tượng

- **MinIO (Thay GCS – Google Cloud Storage)**  
→ Hệ thống lưu trữ object self-hosted, dùng để lưu các file dữ liệu trung gian (CSV, JSON, Parquet).  
→ Hỗ trợ API S3 tương thích giúp dễ dàng tích hợp với Python hoặc Airflow.

### 3.4. Công cụ ETL và mô hình hóa dữ liệu

- **dbt (Data Build Tool)**  
→ Mô hình hóa dữ liệu trong PostgreSQL:
  - Tạo dimension & fact tables
  - Viết logic chuyển đổi bằng SQL
  - Tạo data test và documentation
- **ip2location-python**  
→ Dùng để tra cứu vị trí địa lý từ IP.

---

### 3.5. Trực quan hóa dữ liệu

- **Looker Studio** (hoặc thay thế như Metabase, Superset)  
→ Kết nối trực tiếp với PostgreSQL để trực quan hóa dữ liệu, tạo dashboard báo cáo:
  - Phân tích doanh thu
  - Hiệu suất sản phẩm
  - Phân bố theo vị trí địa lý và thời gian

### 3.6. Hạ tầng và môi trường triển khai

- **Virtual Machine (VM)**  
→ Chạy MongoDB, Python script, MinIO server, PostgreSQL nếu sử dụng cục bộ.

### 3.7. Thư viện & Framework hỗ trợ trong Python

- **pymongo** – Kết nối MongoDB
- **pandas** – Xử lý dữ liệu
- **minio** – Kết nối và thao tác với MinIO
- **psycopg2** hoặc **sqlalchemy** – Kết nối PostgreSQL
- **logging** – Ghi log ETL
- **schedule**, **watchdog**, **cron** – Tạo trigger ETL tự động
- **ip2location** – Xử lý IP định vị

## 4. PHÂN TÍCH YÊU CẦU NGHIỆP VỤ CỦA DATA ENGINEER TRONG DỰ ÁN



## 4.1. Mục tiêu nghiệp vụ chính

Data Engineer cần đảm bảo rằng dữ liệu thô từ các nguồn khác nhau được thu thập, xử lý, lưu trữ và cung cấp sẵn sàng cho phân tích, trực quan và ra quyết định. Trong phạm vi này, các yêu cầu nghiệp vụ chính gồm:

- Thiết kế kiến trúc hạ tầng dữ liệu phù hợp với mục tiêu phân tích
- Thu thập và tổ chức dữ liệu từ nhiều nguồn (web logs, MongoDB, file uploads...)
- Xây dựng pipeline dữ liệu tự động, hiệu quả và có thể mở rộng
- Đảm bảo chất lượng và tính sẵn sàng của dữ liệu (data quality, availability)
- Tối ưu hóa chi phí và hiệu năng lưu trữ, truy vấn

## 4.2. Yêu cầu cụ thể theo nghiệp vụ



### A. Thu thập và lưu trữ dữ liệu

- Kết nối và trích xuất dữ liệu từ các nguồn: MongoDB, files trong VM hoặc hệ thống log.
- Lưu trữ dữ liệu vào MinIO dưới định dạng phù hợp (CSV, JSONL, Parquet) để dễ dàng xử lý và nạp.
- Áp dụng các chiến lược chia batch, ghi log tiến trình thu thập, và xử lý lỗi.

### B. Thiết kế kho dữ liệu (Data Warehouse Layer)

- Xây dựng cấu trúc schema cho PostgreSQL: phân tầng raw → staging → modeling.
- Đảm bảo khả năng truy vấn hiệu quả bằng cách chuẩn hóa dữ liệu (dimension/fact).
- Thiết kế schema phù hợp với mục tiêu phân tích như: revenue analysis, time trends, product insights.

### **C. Xây dựng pipeline xử lý (ETL/ELT)**

- Tự động hóa quá trình tải dữ liệu từ MinIO vào PostgreSQL.
- Dùng Python để:
  - Theo dõi thay đổi file mới (trigger)
  - Làm sạch và chuyển đổi dữ liệu
  - Đưa dữ liệu vào bảng staging/raw
- Kiểm thử đầu cuối (E2E test), ghi log và gửi cảnh báo khi lỗi.

### **D. Chuẩn bị dữ liệu cho phân tích**

- Sử dụng dbt để:
  - Chuyển đổi dữ liệu từ raw thành bảng mô hình hóa
  - Xây dựng dimension và fact table
  - Kiểm tra chất lượng dữ liệu (data tests)
- Gắn metadata, mô tả các trường (documentation) để phục vụ phân tích viên.

### **E. Đảm bảo chất lượng và bảo trì dữ liệu**

- Triển khai profiling và kiểm tra chất lượng dữ liệu:
  - Kiểm tra null, duplicate, type mismatch
  - Theo dõi sự thay đổi schema nguồn
- Tối ưu pipeline và xử lý lỗi kịp thời.

### **F. Phối hợp với các bên liên quan**

- Làm việc với Data Analyst, Business Analyst để hiểu rõ yêu cầu phân

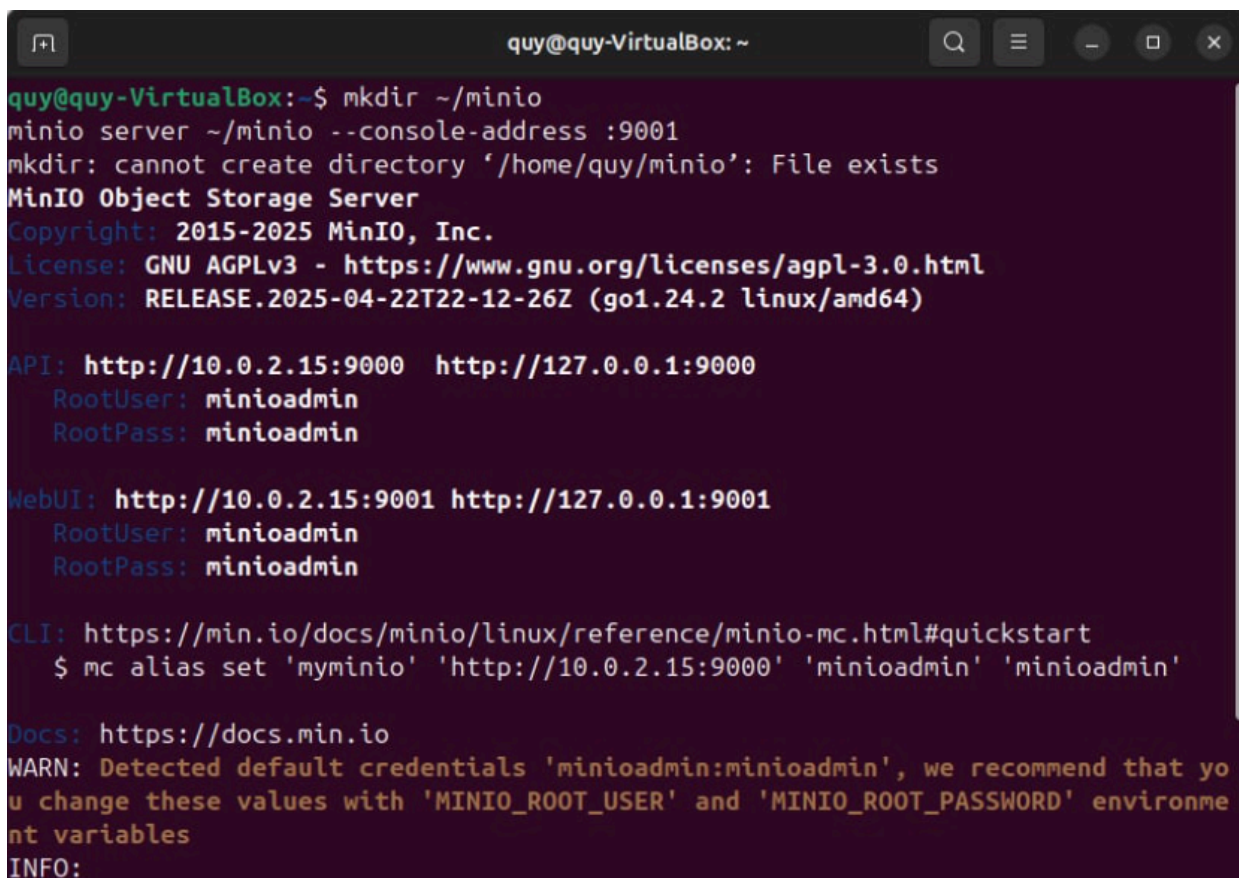
tích.

- Thiết kế pipeline và mô hình dữ liệu phù hợp mục tiêu kinh doanh (ví dụ: hành vi người dùng, phân tích sản phẩm).
- Hỗ trợ giải thích pipeline, data flow, định nghĩa dữ liệu.

## 5. THIẾT KẾ VÀ CÀI ĐẶT

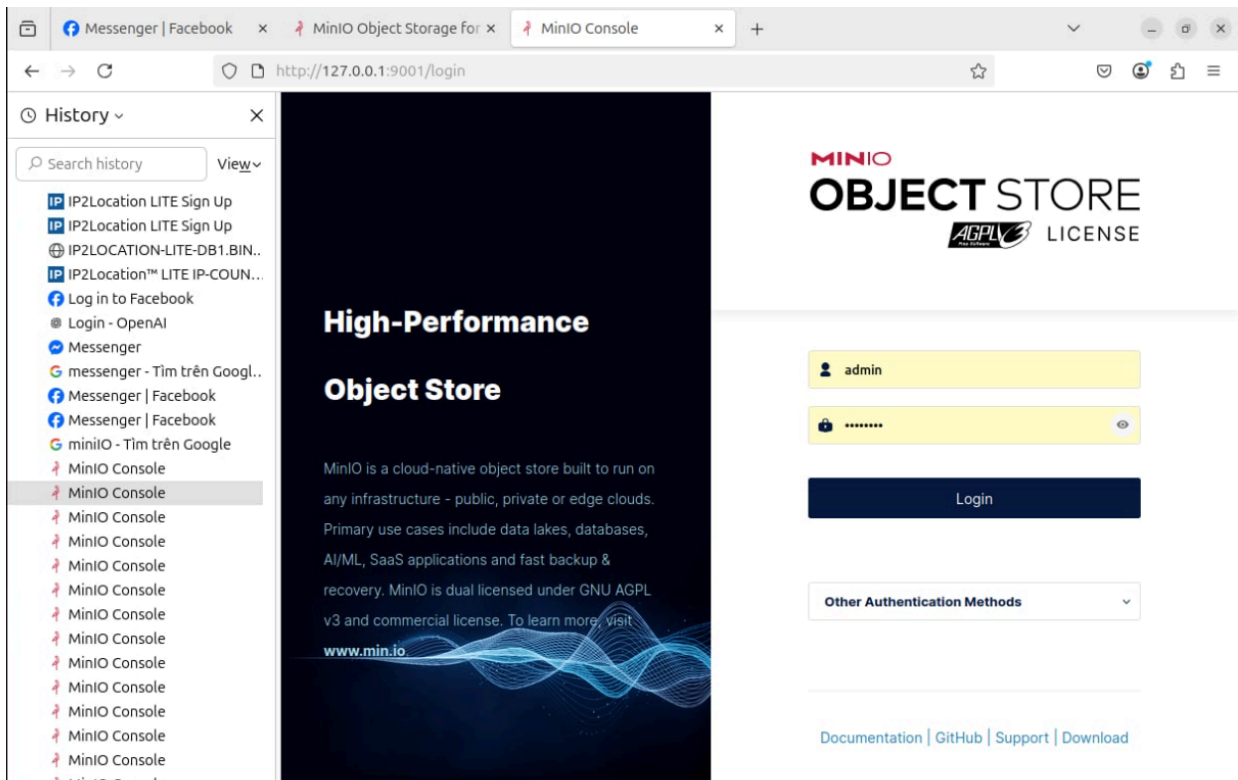
### 5.1. Thiết lập môi trường miniO

-> Khởi động kết nối giữa máy chủ ubuntu và miniO

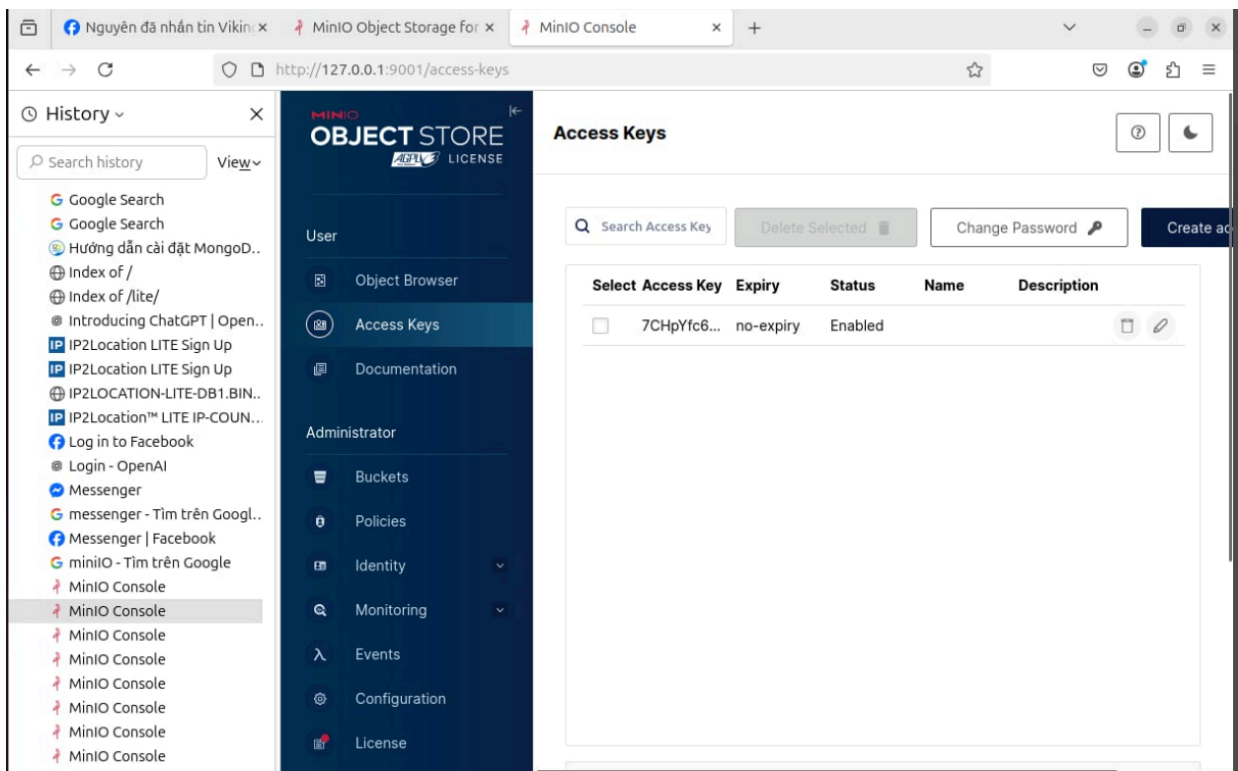


```
quy@quy-VirtualBox: ~  
quy@quy-VirtualBox:~$ mkdir ~/minio  
minio server ~/minio --console-address :9001  
mkdir: cannot create directory '/home/quy/minio': File exists  
MinIO Object Storage Server  
Copyright: 2015-2025 MinIO, Inc.  
License: GNU AGPLv3 - https://www.gnu.org/licenses/agpl-3.0.html  
Version: RELEASE.2025-04-22T22-12-26Z (go1.24.2 linux/amd64)  
  
API: http://10.0.2.15:9000 http://127.0.0.1:9000  
RootUser: minioadmin  
RootPass: minioadmin  
  
WebUI: http://10.0.2.15:9001 http://127.0.0.1:9001  
RootUser: minioadmin  
RootPass: minioadmin  
  
CLI: https://min.io/docs/minio/linux/reference/minio-mc.html#quickstart  
$ mc alias set 'myminio' 'http://10.0.2.15:9000' 'minioadmin' 'minioadmin'  
  
Docs: https://docs.min.io  
WARN: Detected default credentials 'minioadmin:minioadmin', we recommend that you change these values with 'MINIO_ROOT_USER' and 'MINIO_ROOT_PASSWORD' environment variables  
INFO:
```

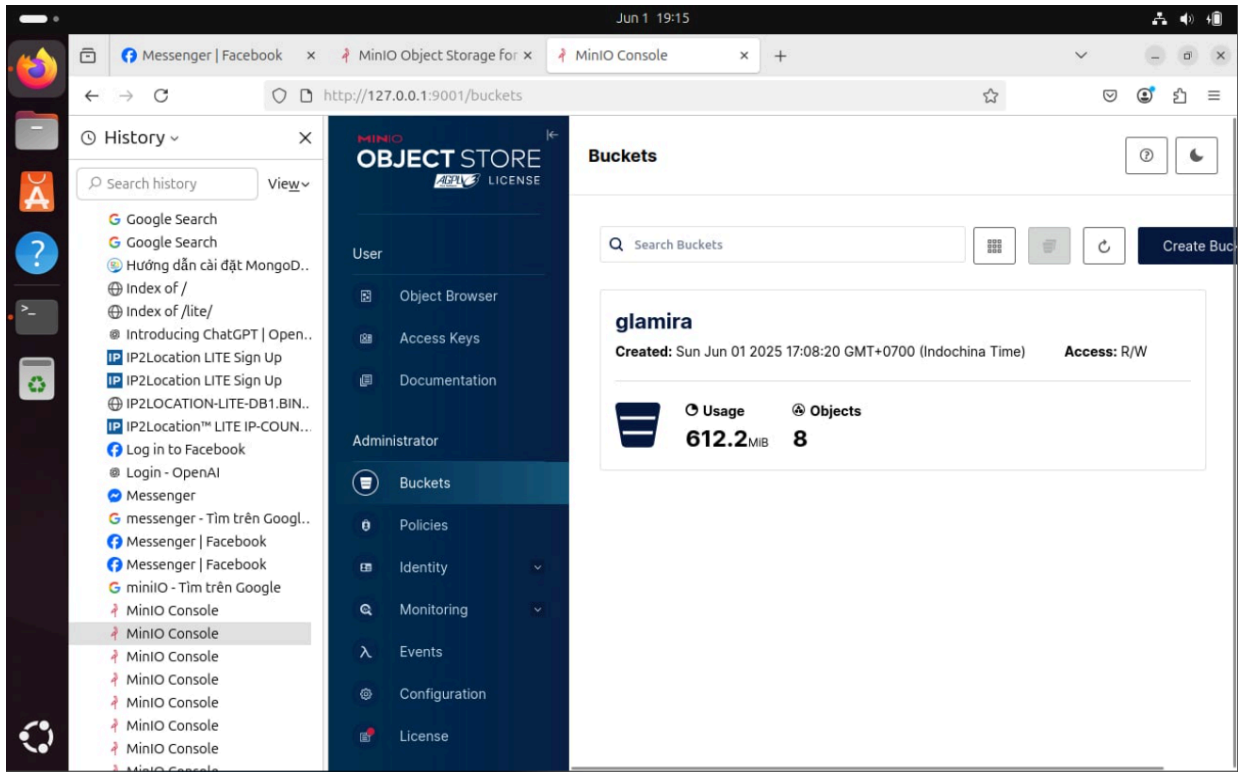
-> Kết nối thành công và đăng nhập tới tài khoản mật khẩu trên



-> Tiến hành authorization với accesskey và secretkey (Ở dưới đã tạo)



-> Tiến hành tạo 1 bucket mới để lưu trữ dữ liệu của dự án ( Ở dưới là bucket glamira đã test upload file lên và đã upload raw data lên)



## 5.2 Thiết lập mongodb trên máy ảo

-> Ta cần tải thư viện mongod ( Do đã cài nên chỉ để code bash )

### # Cài đặt GPG key

```
wget -qO - https://www.mongodb.org/static/pgp/server-6.0.asc | sudo  
apt-key add -
```

### # Thêm repo MongoDB

```
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu  
focal/mongodb-org/6.0 multiverse" | sudo tee  
/etc/apt/sources.list.d/mongodb-org-6.0.list
```

### # Cập nhật và cài đặt

```
sudo apt update  
sudo apt install -y mongodb-org
```

### # Khởi động MongoDB

```
sudo systemctl start mongod  
sudo systemctl enable mongod  
-> khởi động mongod
```



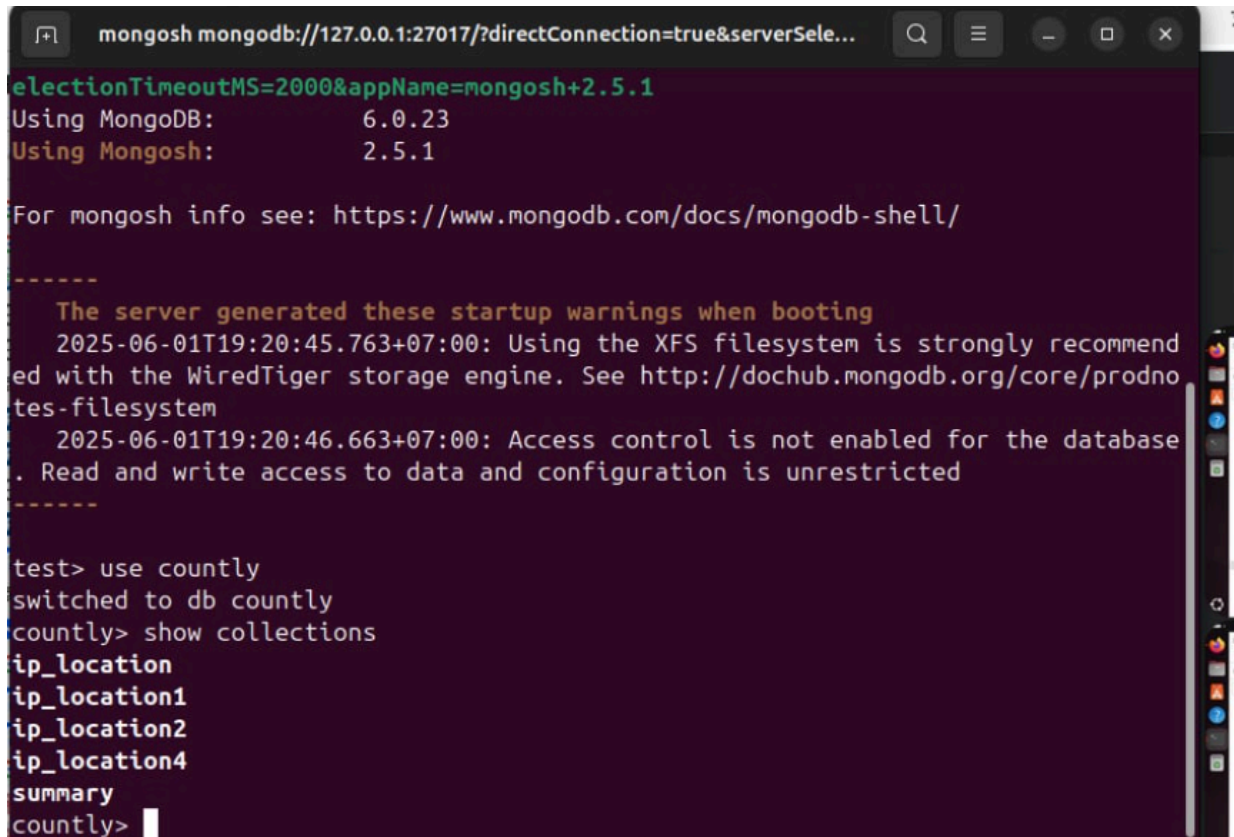
```
quy@quy-VirtualBox: ~  
quy@quy-VirtualBox:~$ source ven/bin/activate  
bash: ven/bin/activate: No such file or directory  
quy@quy-VirtualBox:~$ source venv/bin/activate  
(venv) quy@quy-VirtualBox:~$ pip install mongod  
ERROR: Could not find a version that satisfies the requirement mongod (from versions: none)  
ERROR: No matching distribution found for mongod  
(venv) quy@quy-VirtualBox:~$ pip install mongosh  
ERROR: Could not find a version that satisfies the requirement mongosh (from versions: none)  
ERROR: No matching distribution found for mongosh  
(venv) quy@quy-VirtualBox:~$ sudo systemctl start mongod  
[sudo] password for quy:  
(venv) quy@quy-VirtualBox:~$
```

-> Đăng nhập vào mongodb

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSele...  
ERROR: Could not find a version that satisfies the requirement mongosh (from versions: none)  
ERROR: No matching distribution found for mongosh  
(venv) quy@quy-VirtualBox:~$ sudo systemctl start mongod  
[sudo] password for quy:  
(venv) quy@quy-VirtualBox:~$ mongosh  
Current Mongosh Log ID: 683c45aaf33bbae59bc59f34  
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverS  
electionTimeoutMS=2000&appName=mongosh+2.5.1  
Using MongoDB:      6.0.23  
Using Mongosh:      2.5.1  
  
For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/  
  
-----  
The server generated these startup warnings when booting  
2025-06-01T19:20:45.763+07:00: Using the XFS filesystem is strongly recommend  
ed with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem  
2025-06-01T19:20:46.663+07:00: Access control is not enabled for the database  
. Read and write access to data and configuration is unrestricted  
-----  
test> 
```

-> Bên dưới là database và các collections chính của dự án

- +database chính là countly
- +file raw data là file summary
- +file chứa các ip độc nhất là ip\_location



```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSele...
electionTimeoutMS=2000&appName=mongosh+2.5.1
Using MongoDB: 6.0.23
Using Mongosh: 2.5.1

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

-----
  The server generated these startup warnings when booting
  2025-06-01T19:20:45.763+07:00: Using the XFS filesystem is strongly recommend
ed with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodno
tes-filesystem
  2025-06-01T19:20:46.663+07:00: Access control is not enabled for the database
. Read and write access to data and configuration is unrestricted
-----

test> use countly
switched to db countly
countly> show collections
ip_location
ip_location1
ip_location2
ip_location4
summary
countly>
```

## 6.Kết Luận

Qua quá trình thực hiện chuỗi ba dự án bao gồm: Thu thập và lưu trữ dữ liệu, Xây dựng pipeline tự động, và Chuyển đổi & trực quan hóa dữ liệu, nhóm đã có cơ hội tiếp cận và vận dụng toàn bộ chu trình xử lý dữ liệu thực tế, từ thu thập đến phân tích và trực quan hóa.

Trong giai đoạn đầu, nhóm triển khai cài đặt cơ sở hạ tầng với MongoDB trên máy ảo (VM), đồng thời thực hiện việc tải, khám phá và làm sạch dữ liệu thô từ các bộ dữ liệu liên quan đến hành vi người dùng và sản phẩm. Quá trình xử lý IP định vị, lọc dữ liệu cần thiết, và xây dựng dictionary là nền tảng quan trọng cho bước tiếp theo.

Sang giai đoạn hai, nhóm xây dựng pipeline tự động sử dụng MinIO để thay thế Google Cloud Storage. Các script Python được thiết kế nhằm trích xuất dữ liệu từ MongoDB, chuyển đổi sang định dạng phù hợp (CSV/JSON), và tải lên MinIO. Đồng thời, pipeline ETL được hoàn thiện

với khả năng kích hoạt tự động thông qua trình theo dõi sự kiện, tiếp tục nạp dữ liệu vào hệ quản trị cơ sở dữ liệu PostgreSQL theo cấu trúc tầng raw. Việc tích hợp MinIO và PostgreSQL thay thế GCP thể hiện tính linh hoạt và khả năng áp dụng công nghệ mã nguồn mở vào thực tế.

Ở giai đoạn cuối, nhóm tập trung xây dựng các mô hình dữ liệu dạng chiều (dimensional modeling) và thực hiện phân tích dữ liệu phục vụ cho mục tiêu kinh doanh. Dữ liệu được xử lý và chuyển đổi theo cấu trúc fact/dimension bằng Python. Sau đó, nhóm sử dụng các công cụ trực quan hóa (như Power BI, Metabase hoặc Superset thay cho Looker) để tạo ra các dashboard theo dõi doanh thu, hành vi người dùng, sản phẩm bán chạy và phân bố theo khu vực địa lý.

Thông qua quá trình này, nhóm không chỉ nâng cao kỹ năng kỹ thuật như lập trình Python, quản lý dữ liệu NoSQL/SQL, sử dụng MinIO và PostgreSQL, mà còn rèn luyện tư duy hệ thống và khả năng phân tích nghiệp vụ. Bên cạnh đó, kỹ năng phối hợp nhóm, quản lý thời gian và trình bày kết quả cũng được cải thiện đáng kể.

Tuy còn một số hạn chế về dữ liệu và thời gian thực hiện, nhóm tin rằng kết quả đạt được đã phản ánh rõ ràng kiến thức và kỹ năng cốt lõi của một Data Engineer trong việc xây dựng một hệ thống dữ liệu hiện đại, tự động, có khả năng mở rộng và ứng dụng thực tế cao.

## 7.TÀI LIỆU THAM KHẢO

- MongoDB Documentation – <https://www.mongodb.com/docs>
- MinIO Documentation – <https://min.io/docs/minio>
- PostgreSQL Official Site – <https://www.postgresql.org/docs>
- IP2Location Python Library – <https://github.com/ip2location/ip2location-python>
- Python Official Documentation – <https://docs.python.org>
- Pymongo – MongoDB Driver for Python – <https://pymongo.readthedocs.io>
- SQL Style Guide – <https://www.sqlstyle.guide>



- Superset – Modern Data Visualization Platform – <https://superset.apache.org>
- Metabase Documentation – <https://www.metabase.com/docs>
- Real-World Data Engineering Projects – DataTalksClub – <https://datatalks.club>
- Google Cloud (được dùng tham khảo giai đoạn đầu) – <https://cloud.google.com>
- Python Logging Documentation – <https://docs.python.org/3/library/logging.html>
- Open Source ETL Tools Comparison – Towards Data Science Blog
- Clean Code for Data – Best Practices in Data Pipelines – Data Engineering Weekly