BỘ THÔNG TIN VÀ TRUYỀN THÔNG

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

BÁO CÁO THỰC TẬP CƠ SỞ

Triển khai bước 1 của dự án

Giảng viên hướng dẫn : Kim Ngọc Bách

Họ và tên : Ngô Vũ Minh Quý

Mã sinh viên : B22DCVT427

Lớp : E22CQCN02-B

MỤC LỤC

I. Giới thiệu dự án

II. Hạ tầng và công nghệ sử dụng

III. Các bước triển khai chi tiết

IV. Kết quả đạt được

V. Đánh giá và kết luận

I. Giới thiệu dự án

Dự án “Data Collection & Storage Foundation” là một dự án thực hành trong chương trình đào tạo, tập trung vào việc xây dựng nền tảng lưu trữ và xử lý dữ liệu thô ban đầu, phục vụ cho các bước phân tích và trực quan hóa dữ liệu sau này. Dự án này cho phép sinh viên tiếp cận với các công nghệ hiện đại như **MinIO** (thay thế GCS), **MongoDB**, **Python**, và thư viện **IP2Location**, từ đó hiểu được luồng dữ liệu từ khâu thu thập đến xử lý lưu trữ.

II. Hạ tầng và công nghệ sử dụng

- **MinIO**: Hệ thống lưu trữ đối tượng tương thích S3, thay thế Google Cloud Storage, dùng để lưu trữ dữ liệu thô.
- **MongoDB**: Cơ sở dữ liệu NoSQL dạng document để lưu trữ và xử lý dữ liệu bán cấu trúc.
- **Python**: Ngôn ngữ xử lý dữ liệu, tích hợp với thư viện IP2Location.
- **Virtual Machine (VM)**: Máy chủ ảo để triển khai MongoDB và môi trường xử lý dữ liệu.
- **IP2Location**: Thư viện xác định vị trí địa lý từ địa chỉ IP.

III. Các bước triển khai chi tiết

1. Cài đặt môi trường (1 ngày)

- Tạo máy ảo trên nền tảng cloud hoặc cục bộ (VD: VirtualBox).
- Cài đặt Python và MongoDB trên VM.
- Tải về tập dữ liệu **Glamira dataset** để sử dụng trong dự án.

2. Thiết lập MinIO (1 ngày)

- Cài đặt và khởi chạy MinIO server:

```
from minio import Minio
from minio.error import S3Error

client = Minio(
    "localhost:9000",  # Hoặc dùng IP VM nếu remote
    access_key="minioadmin",
    secret_key="minioadmin",
    secure=False
)

bucket_name = "raw-data"
file_path = "glamira_dataset.csv"
object_name = "glamira_dataset.csv"

# Tạo bucket nếu chưa có
if not client.bucket_exists(bucket_name):
    client.make_bucket(bucket_name)

# Upload file
client.fput_object(bucket_name, object_name, file_path)
print(f"✅ Uploaded {file_path} to bucket {bucket_name}")
```

- Tạo bucket `raw_data/` để chứa tập tin dữ liệu ban đầu.

- Cấu hình xác thực qua Access Key và Secret Key.
- Upload file dữ liệu thô lên bucket.

3. Cài đặt MongoDB và kết nối (1 ngày)

- Cài đặt MongoDB trên máy chủ ảo:

```
from pymongo import MongoClient
import csv

client = MongoClient("mongodb://localhost:27017/")
db = client["glamira"]
collection = db["raw_events"]

with open("glamira_dataset.csv", "r", encoding="utf-8") as f:
    reader = csv.DictReader(f)
    batch = []
    for row in reader:
        batch.append(row)
        if len(batch) == 1000:
            collection.insert_many(batch)
            batch = []
    if batch:
        collection.insert_many(batch)

print("✅ Data inserted into MongoDB")
```

- Khởi tạo database và kiểm tra kết nối từ máy khách.
- Cấu hình remote access nếu cần.

4. Nạp dữ liệu ban đầu (2 ngày)

- Viết script Python để parse và insert dữ liệu vào MongoDB.
- Tạo các collection tương ứng trong MongoDB.
- Thực hiện một số truy vấn cơ bản để kiểm tra dữ liệu.
- Xây dựng **data dictionary** mô tả các trường dữ liệu và mối liên hệ.

5. Xử lý địa chỉ IP (3 ngày)

- Cài đặt thư viện `ip2location-python`.
- Viết script Python để:
    - Truy vấn danh sách địa chỉ IP duy nhất trong MongoDB.
    - Dùng IP2Location để xác định vị trí (quốc gia, thành phố, v.v.).
    - Lưu kết quả vào collection mới trong MongoDB.
- Xử lý lỗi và test thử trên dữ liệu mẫu.

python
Sao chépChỉnh sửa

```
from pymongo import MongoClient
import ip2location
```

```
def process_ip_locations():
    db = MongoClient().mydb
    ip_list = db.logs.distinct('ip')
    for ip in ip_list:
        result =
ip2location.IP2Location().get_all(ip)
        db.ip_location.insert_one({
            'ip': ip,
            'country': result.country_short,
            'city': result.city
        })
```

6. Thu thập tên sản phẩm (1 ngày)

- Lọc ra các bản ghi có `event_type` là:
    - `view_product_detail`
    - `select_product_option`
    - `select_product_option_quality`
- Trích xuất `product_id` và `current_url`.
- Crawling trang sản phẩm để lấy tên sản phẩm (chỉ lấy tên **duy nhất** cho mỗi `product_id`).
- Lưu dữ liệu kết quả vào file CSV phục vụ cho giai đoạn ETL.

7. Tài liệu hóa và kiểm thử (2 ngày)

- Ghi lại quy trình cài đặt và xử lý dữ liệu.
- Kiểm tra tính đầy đủ và hợp lệ của dữ liệu:
    - Thống kê số bản ghi null, dữ liệu bị thiếu.

    - Kiểm tra kiểu dữ liệu các trường.
    - Thống kê giá trị khác biệt.
- Tạo báo cáo profiling về dữ liệu.

IV. Kết quả đạt được

- Xây dựng thành công pipeline đơn giản thu thập – lưu trữ – xử lý dữ liệu ban đầu.
- Dữ liệu được tổ chức rõ ràng trên MongoDB, có bổ sung thông tin vị trí từ IP.
- Lấy thành công tên sản phẩm và lưu thành định dạng CSV phục vụ phân tích tiếp theo.
- Quản lý tập tin dữ liệu thô trên MinIO, cấu hình đơn giản, dễ triển khai nội bộ.

V. Đánh giá và kết luận

Dự án giúp em hiểu được quy trình thu thập và xử lý dữ liệu ban đầu, một khía cạnh rất quan trọng trong các hệ thống phân tích dữ liệu thực tế. Việc làm quen với MinIO giúp em hiểu được mô hình lưu trữ đối tượng tương tự GCS hoặc S3. Bên cạnh đó, việc triển khai MongoDB giúp hiểu rõ cấu trúc dữ liệu NoSQL, phù hợp với dữ liệu logs và bán cấu trúc hiện đại.

Khó khăn lớn nhất là việc crawling tên sản phẩm yêu cầu hiểu về HTML DOM và xử lý mạng, nhưng cũng từ đó em học được cách sử dụng thư viện `requests` và `BeautifulSoup` trong Python. Ngoài ra, việc profiling dữ liệu giúp em nhìn rõ hơn về chất lượng dữ liệu và sự cần thiết của giai đoạn tiền xử lý.