

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

— * —

Project 3

ĐỀ TÀI: Reddit Data Pipeline

GVHD: ThS. Nguyễn Duy Hiệp

Người thực hiện:

Bùi Huy Thái – 20204688

Hà Nội, tháng 01 năm 2024

MỤC LỤC

Chương 1: MỞ ĐẦU	3
1.1 Giới thiệu đề tài	3
1.2 Mục tiêu thực hiện	3
Chương 2: TỔNG QUAN & CÔNG NGHỆ	5
2.1 Tổng quan Project	5
2.2 : Các công nghệ sử dụng	5
2.2.1: Reddit API & PRAW	5
2.2.2: Terraform	6
2.2.3: Prefect	6
2.2.4: Google Cloud Platform	7
2.2.4: dbt (Data Build Tool)	7
2.2.5: Google Data Studio	8
Chương 3: Chi tiết cấu trúc Project	8
3.1 Thu thập dữ liệu	8
3.2 Cấu hình GCP với Terraform	9
3.3 Đưa dữ liệu lên Google Cloud Storage và chuyển qua Google BigQuery	11
3.4 Transform data với dbt	12
3.5 Tạo luồng và lập lịch với Prefect	14
3.6 Phân tích dữ liệu với Google Data Studio	16
Chương 4: Khó khăn và kết quả	18
4.1 Khó khăn	18
4.2 Kết quả	18
4.3 Phát triển thêm	18
Chương 5: Kết luận	19

Chương 1: MỞ ĐẦU

1.1 Giới thiệu đề tài

Trong thời đại số hóa ngày nay, dữ liệu ngày càng trở thành một tài nguyên vô cùng quý giá. Reddit, một trong những mạng xã hội lớn nhất và phổ biến nhất trên Internet, cung cấp một nguồn dữ liệu phong phú và đa dạng về các chủ đề, ý kiến và sự tương tác của người dùng. Việc thu thập và phân tích dữ liệu từ Reddit đòi hỏi một hệ thống Data Pipeline hiệu quả.

Việc xây dựng một Data Pipeline hiệu quả cho việc thu thập và phân tích dữ liệu từ Reddit mang lại nhiều lợi ích quan trọng. Điều này có thể hỗ trợ các nhà nghiên cứu, nhà phân tích dữ liệu và các doanh nghiệp trong việc hiểu rõ hơn về ý kiến, xu hướng và tương tác của người dùng trên Reddit. Dữ liệu từ Reddit cũng có thể được ứng dụng trong việc phân tích thị trường, dự đoán xu hướng và tạo ra các sản phẩm hoặc dịch vụ phù hợp với nhu cầu của người dùng.

1.2 Mục tiêu thực hiện

Project này sẽ tập trung vào xây dựng một Data Pipeline toàn diện để thu thập dữ liệu từ Reddit và tiến hành lưu trữ và phân tích. Phạm vi của nghiên cứu bao gồm việc thiết kế kiến trúc hệ thống, triển khai công cụ thu thập dữ liệu, xây dựng hệ thống lưu trữ dữ liệu, phát triển các kỹ thuật phân tích dữ liệu và tạo ra giao diện người dùng cho việc truy vấn và trực quan hóa dữ liệu thu thập được.

Phương pháp thực hiện dự kiến:

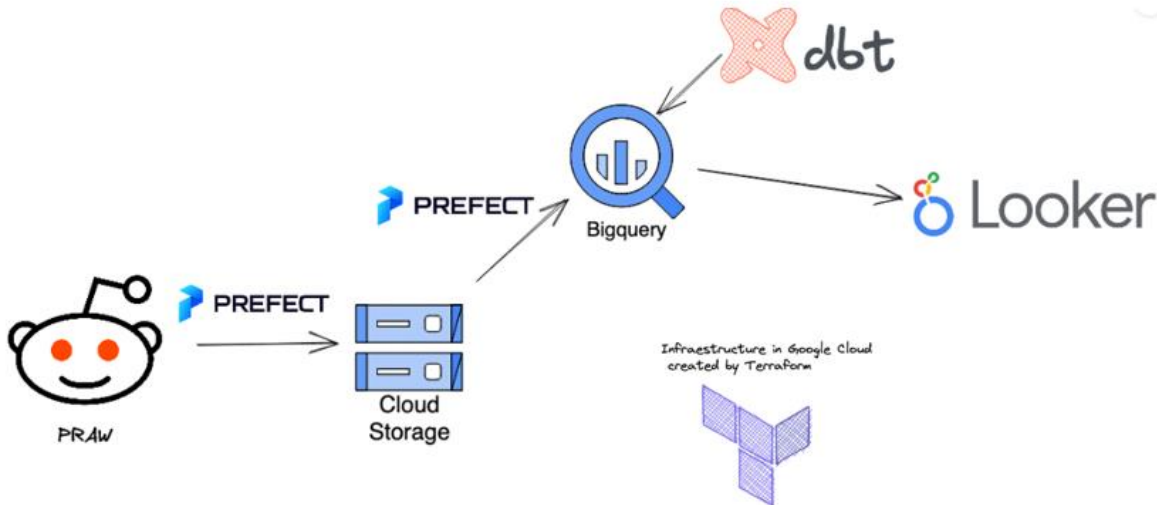
- Tìm hiểu về cấu trúc dữ liệu và API của Reddit để hiểu cách thu thập dữ liệu từ các subreddit và bài viết.
- Thiết kế và triển khai một Data Pipeline bao gồm các bước thu thập dữ liệu, lưu trữ và phân tích.
- Sử dụng các công nghệ và công cụ phù hợp để xây dựng hệ thống lưu trữ dữ liệu có khả năng mở rộng và đảm bảo tính bảo mật.
- Áp dụng các kỹ thuật phân tích dữ liệu như xử lý ngôn ngữ tự nhiên, phân loại và phân tích tương tác cộng đồng để trích xuất thông tin quan trọng từ dữ liệu Reddit.

Dự kiến kết quả:

Kết quả của nghiên cứu sẽ là một Data Pipeline hoàn chỉnh cho việc thu thập, lưu trữ và phân tích dữ liệu từ Reddit. Hệ thống sẽ có khả năng tự động thu thập dữ liệu mới từ Reddit và lưu trữ nó trong một hệ thống lưu trữ hiệu quả. Ngoài ra, dữ liệu thu thập được sẽ được phân tích để trích xuất thông tin quan trọng và hiển thị trực quan thông qua giao diện người dùng.

Chương 2: TỔNG QUAN & CÔNG NGHỆ

2.1 Tổng quan Project



Project của em có luồng cơ bản như sau:

- Dữ liệu được thu thập bằng Reddit API cùng với thư viện PRAW
- Project sử dụng Google Cloud:
 - o Sử dụng Terraform để xây dựng Infrastructure của Google Cloud
 - o Google Cloud VM để chạy code thu thập và xử lý logic data
 - o Data sau khi thu về được lưu lên Google Cloud Storage
 - o Dữ liệu được lấy từ GCS đẩy sang Google BigQuery
 - o Dữ liệu trong BigQuery được xử lý bằng dbt
- Luồng dữ liệu được quản lý và lập lịch bởi Prefect
- Data được phân tích bằng Google Data Studio

2.2: Các công nghệ sử dụng

2.2.1: Reddit API & PRAW

- Reddit API:

Reddit API là API được cung cấp bởi Reddit, nền tảng mạng xã hội lớn nhất và phổ biến nhất trên internet. Reddit API cho phép các nhà phát triển truy cập và tương tác với dữ liệu và

chức năng của Reddit thông qua các yêu cầu HTTP. API cung cấp khả năng truy cập vào các subreddit, bài viết, bình luận, thông tin người dùng và nhiều tính năng khác trên Reddit.

- PRAW (Python Reddit API Wrapper):

PRAW là một thư viện Python phổ biến và mạnh mẽ được sử dụng để tương tác với Reddit thông qua Reddit API. PRAW cho phép các nhà phát triển dễ dàng thực hiện các yêu cầu API, thu thập dữ liệu và tham gia vào các hoạt động trên Reddit. Với PRAW, việc lấy dữ liệu từ các subreddit, đăng bài, đăng bình luận và thực hiện các tác vụ khác trên Reddit trở nên đơn giản và tiện lợi.

- Tính năng của Reddit API và PRAW:

- Truy cập dữ liệu subreddit: Reddit API cho phép truy cập vào các thông tin về subreddit, bao gồm tên, mô tả, thành viên, ngày thành lập và nhiều thông tin khác. PRAW cung cấp các phương thức để truy cập và xử lý dễ dàng các thông tin này.
- Thu thập bài viết và bình luận: Reddit API cho phép truy cập các bài viết và bình luận trong một subreddit cụ thể. PRAW cung cấp các phương thức để lấy danh sách các bài viết, thông tin chi tiết về từng bài viết và các bình luận liên quan.

2.2.2: Terraform

Terraform là một công cụ mã hóa cấu hình (Infrastructure as Code) phổ biến và mở rộng, được phát triển bởi HashiCorp. Nó cho phép bạn xây dựng, quản lý và tự động hóa hạ tầng phần mềm (infrastructure) của bạn trên đám mây và môi trường máy chủ. Terraform sử dụng ngôn ngữ khai báo đơn giản và đồng nhất, cho phép bạn xác định cấu trúc hạ tầng của mình dưới dạng mã và quản lý tài nguyên như máy chủ, mạng, cơ sở dữ liệu và nhiều hơn nữa. Terraform giúp đơn giản hóa việc triển khai và duy trì hạ tầng của bạn, đồng thời đảm bảo tính nhất quán và tái sử dụng mã nguồn trong quá trình phát triển ứng dụng và cơ sở hạ tầng.

2.2.3: Prefect

Prefect là một công cụ mã hóa quy trình dữ liệu (data orchestration) mã nguồn mở và mạnh mẽ. Nó được thiết kế để giúp tổ chức và quản lý các quy trình dữ liệu phức tạp, từ việc gửi, xử lý, và lưu trữ dữ liệu đến việc lập lịch thực hiện các tác vụ. Prefect cung cấp một mô hình lập trình đơn giản và trực quan, cho phép người dùng xác định và điều chỉnh các quy trình dữ liệu của họ dưới dạng mã nguồn.

Các tính năng chính của Prefect bao gồm:

- Mô hình dữ liệu:

Prefect cho phép xây dựng các quy trình dữ liệu bằng Python, với khả năng sử dụng các thư viện và công cụ phổ biến trong cộng đồng Python.

- Quản lý động và linh hoạt:
Prefect giúp quản lý các quy trình dữ liệu phức tạp thông qua việc xác định các luồng công việc, điều kiện và phụ thuộc dữ liệu. Bạn có thể điều chỉnh và cập nhật quy trình dữ liệu một cách linh hoạt mà không cần thay đổi mã nguồn.
- Tích hợp công cụ và hệ thống khác:
Prefect tích hợp một cách tốt với các công cụ và hệ thống khác như Kubernetes, Docker, AWS, GCP và nhiều hơn nữa. Điều này cho phép bạn triển khai và chạy các quy trình dữ liệu trên nhiều môi trường và nền tảng khác nhau.
- Giám sát và xử lý lỗi:
Prefect cung cấp khả năng giám sát và theo dõi quy trình dữ liệu ngay từ khi chúng được triển khai, ghi lại, thông báo, và xử lý lỗi mạnh mẽ để giúp bạn phát hiện và khắc phục sự cố trong quy trình dữ liệu của mình.

2.2.4: Google Cloud Platform

- Google Cloud Platform (GCP) là một nền tảng điện toán đám mây công cộng được cung cấp bởi Google. Nó cung cấp một loạt các dịch vụ đám mây để lưu trữ, quản lý và xử lý dữ liệu, triển khai ứng dụng và xây dựng hạ tầng.
- Google Cloud Storage là một dịch vụ lưu trữ đám mây của GCP, cung cấp khả năng lưu trữ và truy xuất dữ liệu từ bất kỳ đâu trên Internet. Nó cho phép bạn lưu trữ các đối tượng không giới hạn và có khả năng mở rộng, đồng thời cung cấp tính năng bảo mật và độ tin cậy cao.
- Google BigQuery là một dịch vụ phân tích dữ liệu và truy vấn dữ liệu trên đám mây của GCP. Nó cung cấp một kho dữ liệu phân tán và khả năng truy vấn nhanh chóng cho phép bạn thực hiện phân tích dữ liệu lớn với hiệu suất cao. BigQuery hỗ trợ các công cụ phân tích dữ liệu phổ biến như SQL và cung cấp tính năng mở rộng tự động để xử lý các tập dữ liệu lớn mà không cần quản lý cơ sở hạ tầng.
- Google VM (Virtual Machine) là một dịch vụ máy ảo trên đám mây của GCP. Nó cho phép bạn tạo và quản lý các máy ảo linh hoạt trên nền tảng điện toán đám mây của Google. Bằng cách sử dụng Google VM, bạn có thể triển khai các hệ thống và ứng dụng phức tạp, cung cấp khả năng mở rộng và độ tin cậy cao. Google VM cung cấp nhiều tùy chọn máy ảo với các hệ điều hành khác nhau và khả năng tùy chỉnh tài nguyên phù hợp với yêu cầu của bạn.

2.2.4: dbt (Data Build Tool)

- dbt (data build tool) là một công cụ mã nguồn mở được thiết kế để xây dựng, quản lý và triển khai quy trình xây dựng dữ liệu (data pipelines). Nó giúp tạo ra các quy trình xử lý dữ liệu tự động và tái sử dụng được trong môi trường phát triển dữ liệu.

- dbt sử dụng ngôn ngữ SQL để xác định các bước xử lý dữ liệu và biến chúng thành quy trình xây dựng dữ liệu. Nó cung cấp các tính năng như truy vấn, biến đổi, kết hợp và tạo lại cấu trúc dữ liệu. Một trong những đặc điểm quan trọng của dbt là khả năng phân tách quy trình xây dựng dữ liệu thành các module nhỏ, tái sử dụng và dễ dàng kiểm tra.
- dbt cũng hỗ trợ kiểm tra chất lượng dữ liệu và quản lý phiên bản. Nó cho phép bạn viết các kiểm tra để đảm bảo tính nhất quán và độ tin cậy của dữ liệu. Đồng thời, dbt tích hợp với các công cụ quản lý phiên bản như Git để theo dõi và quản lý các thay đổi trong quy trình xây dựng dữ liệu.
- Với dbt, bạn có thể tạo ra quy trình xây dựng dữ liệu linh hoạt, có khả năng kiểm tra và dễ dàng triển khai. Nó giúp tăng tính nhất quán và hiệu suất trong việc xây dựng và quản lý dữ liệu, đồng thời cung cấp một quy trình phát triển dữ liệu mạnh mẽ và linh hoạt.

2.2.5: Google Data Studio

- Google Data Studio là một công cụ trực quan hóa dữ liệu trực tuyến miễn phí được cung cấp bởi Google. Nó cho phép bạn tạo ra bảng điều khiển và báo cáo tùy chỉnh để trực quan hóa dữ liệu từ các nguồn khác nhau một cách dễ dàng.
- Với Google Data Studio, bạn có thể kết nối và tổ chức dữ liệu từ nhiều nguồn khác nhau như Google Analytics, Google Sheets, Google BigQuery và nhiều nguồn dữ liệu khác. Bạn có thể trực quan hóa dữ liệu bằng cách sử dụng các biểu đồ, bản đồ, bảng và hình ảnh để hiển thị thông tin một cách trực quan và dễ hiểu.
- Với Google Data Studio, ta có thể tạo ra các báo cáo chuyên nghiệp và trực quan từ dữ liệu của mình một cách dễ dàng. Nó là một công cụ hữu ích cho việc trực quan hóa dữ liệu và chia sẻ thông tin với đồng nghiệp, khách hàng và nhóm làm việc.

Chương 3: Chi tiết cấu trúc Project

3.1 Thu thập dữ liệu

Dữ liệu được thu thập từ 3 subreddit là Data Engineering, Data Science, Data Analyst qua Reddit API và thư viện PRAW

Để bắt đầu thu thập, ta cần tạo tài khoản reddit, sau đó vào phần pref app để xác thực tài khoản và sử dụng Reddit API. Từ đó ta sẽ lấy ra được các thông tin sau:

- App name
- App Id
- API Secret Key

-- Data post:

- author: tác giả bài post

- author_flair_text: huy hiệu của tác giả
- clicked: tài khoản đã click vào bài viết chưa
- distinguished: đây có phải là bài đăng đặc biệt không
- edited: bài viết đã được chỉnh sửa chưa
- post_id: id của bài post
- is_original_content: đây có phải là bài đăng gốc không
- locked: bài viết có bị khóa không
- post_fullname: name của bài viết
- post_title: tiêu đề bài viết
- post_text: phần thân bài viết
- num_comments: tổng số comments của bài viết
- post_score: score của bài viết
- post_url: link đến bài viết
- saved: bài viết đã được lưu chưa
- created_at: thời điểm tạo bài post
- over_18: bài viết có chứa nội dung người lớn không
- spoiler: bài viết có chứa phần spoiler không ?
- Stickied: bài post có được dính ở đầu subreddit không ?
- upvote_ratio: tỉ lệ upvote

-- Comment data:

- author: tác giả comment
- comment_id: id của comments
- post_url: url của bài post chứa comment đó
- body: nội dung của comments
- created_at: thời điểm comment được tạo
- distinguished: đây có phải comment đặc biệt không
- edited: comment đã bị chỉnh sửa chưa
- is_author_submitter: đây có phải comment của tác giả bài viết chưa
- post_id: id của bài post
- link_comment: đường dẫn đến comment
- comment_score: điểm của comment đó

3.2 Cấu hình GCP với Terraform

Google Cloud được em cấu hình với:

- 1 máy ảo

Basic information

Name	huythai-de
Instance Id	8860057406362648457
Description	None
Type	Instance
Status	Running
Creation time	Dec 16, 2023, 10:25:30 PM UTC+07:00
Zone	asia-southeast1-b
Instance template	None
In use by	None
Reservations	Automatically choose
Labels	None
Tags ?	<div>—<div></div></div>
Deletion protection	Disabled
Confidential VM service ?	Disabled
Preserved state size	0 GB

Machine configuration

This instance is underutilized. You can save an estimated \$30 per month by switching to the machine type: e2-medium. [Learn more](#)

RESIZE

DISMISS

Machine type	e2-standard-2
CPU platform	Intel Broadwell
Minimum CPU platform	None
Architecture	x86_64
vCPUs to core ratio ?	—
Custom visible cores ?	—
Display device	Disabled
	Enable to use screen capturing and recording tools

- 1 Bucket Google Cloud Storage:

huythai-zc-1

Location

asia (multiple regions in Asia)

Storage class

Standard

Public access

Not public

Protection

None

OBJECTS

CONFIGURATION

PERMISSIONS

PROTECTION

LIFECYCLE

OBSERVABILITY

INVENTORY REPORTS

Buckets

huythai-zc-1

data

UPLOAD FILES

UPLOAD FOLDER

CREATE FOLDER

TRANSFER DATA

MANAGE HOLDS




DOWNLOAD

DELETE

Filter by name prefix only

Filter

filter objects and folders

<input type="checkbox"/>	Name	Size	Type	Created ?	Storage class	Last modified	Public access ?	Version history ?	Encryption ?
<input type="checkbox"/>	 cmt_ld.parquet	354.5 KB	application/octet-stream	Jan 19, 2024, 7:01:12 AM	Standard	Jan 19, 2024, 7:01:12 AM	Not public	—	Google-managed
<input type="checkbox"/>	 img/	—	Folder	—	—	—	—	—	—
<input type="checkbox"/>	 rde.parquet	214.4 KB	application/octet-stream	Jan 19, 2024, 12:00:26 PM	Standard	Jan 19, 2024, 12:00:26 PM	Not public	—	Google-managed

- 2 Dataset Google BigQuery:



3.3 Đưa dữ liệu lên Google Cloud Storage và chuyển qua Google BigQuery

a. Post

- Upload data lên Google Cloud Storage:
 - o Tải file rde.parquet từ GCS về máy (đây là file chứa thông tin các bài post đã thu thập được từ trước)
 - o Chuyển data trong file rde.parquet thành dạng dataframe pandas (df1)
 - o Thu thập data các bài post reddit bằng PRAW và lưu dưới dạng dataframe (df2)
 - o Chỉnh sửa trường created_at thành dạng datetime
 - o Nối df2 vào df1 với logic là:
 - Các bản ghi của df2 có id đã tồn tại trong df1 => cập nhật các id đó
 - Các bản ghi mới => thêm vào df1
 - o Chuyển df1 thành dạng file parquet (rde.parquet)
 - o Upload file này lên GCS bucket đã chỉ định sẵn
- Chuyển data qua Google BigQuery
 - o Extract data từ file rde.parquet
 - o Lưu data này vào table Bigquery đã được chỉ định

b. Comments

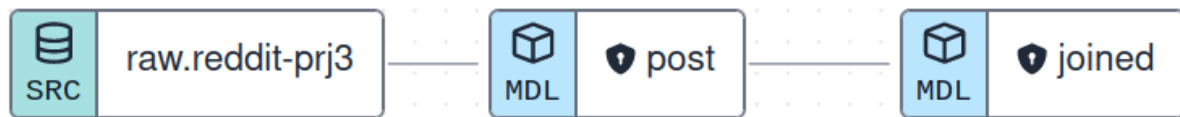
- Upload data lên Google Cloud Storage:
 - o Tải file rde.parquet từ GCS về (file chứa thông tin các bài post đã thu thập được)

- Tải file cmt_rd.parquet từ GCS về (file chứa thông tin các cmt đã thu thập được)
- Chuyển data trong file cmt_re.parquet thành dataframe pandas (df3)
- Tạo list chứa post_url đã thu thập được
- Tạo list chứa cmt_id đã thu thập được
- Duyệt vòng lặp với các post_url đã thu thập được, với từng post_url, thu thập các cmt nếu cmt_id chưa tồn tại trong list chứa cmt_id và lưu dưới dạng dataframe (df4)
- Chỉnh sửa trường created_at thành dạng datetime
- Nối df4 vào df3
- Chuyển df3 thành dạng file parquet (cmt_rd.parquet)
- Chuyển data qua Google BigQuery
 - Extract data từ file cmt_rd.parquet
 - Lưu data này vào table Bigquery đã được chỉ định

3.4 Transform data với dbt

- Tạo prj trên dbt cloud và kết nối với Bigquery để tiến hành transform data
- Kết nối prj với 1 repo trên trình quản lý git (GitHub, GitLab,...) để lưu code
- Cấu hình source data:
 - Tạo file mode/sources.yml
 - Thêm các thông tin vào file sources.yml:
 - Name: tên của sources
 - Database: tên project GCP
 - Schema: tên dataset cần dùng trong Google Bigquery
 - Tables / names : các bảng trong dataset cần sử dụng
- Cấu hình schema data output của dbt
 - Tạo file
 - Khai báo các bảng cùng với các trường cần thiết
- Tạo các hàm
 - Các hàm sẽ được tạo mới trong thư mục macros
 - Tạo hàm extract_hour chỉ lấy ra giờ trong timestamp
 - Tạo hàm normalize_timestamp để timestamp chỉ lấy đến giờ (phút và giây = 0)
- Tạo bảng post
 - tạo file post.sql trong thư mục model
 - bảng post mới được lấy nguồn từ bảng reddit-prj3 với một số thay đổi
 - các trường của bảng post được thêm ‘_post’ để dễ nhận biết
 - num_comments chuyển thành kiểu int
 - post_score chuyển thành kiểu int
 - upvote_ratio chuyển thành kiểu float
 - tạo trường mới hour_post_created_at: giờ mà post được tạo

- `normalize_post_created_at`: thời điểm post được tạo theo kiểu timestamp được chuẩn hóa như trên
- chỉ lấy các post có thời điểm tạo sau ngày 2023-01-01
- chỉ lấy các post có phần `post_text` (những post không có phần `post_text` là các bài quảng cáo)



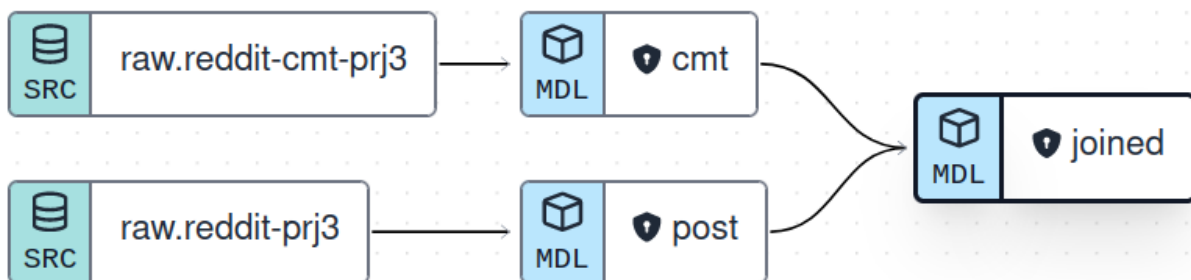
- Tạo bảng cmt

- tạo file `cmt.sql` trong thư mục `model`
- bảng `cmt` mới được lấy nguồn từ bảng `reddit-cmt-prj3` với một số thay đổi:
 - các trường của bảng `cmt` được thêm `'_comment'`
 - tạo trường mới `hour_post_created_at`: giờ mà post được tạo
 - `normalize_post_created_at`: thời điểm post được tạo theo kiểu timestamp được chuẩn hóa như trên
 - thêm `'reddit.com'` vào trước của `link_comment`



- Tạo bảng join

- Join 2 bảng `post` và `cmt` vừa tạo để tạo thành bảng mới `joined`



• Deployment và Schedule

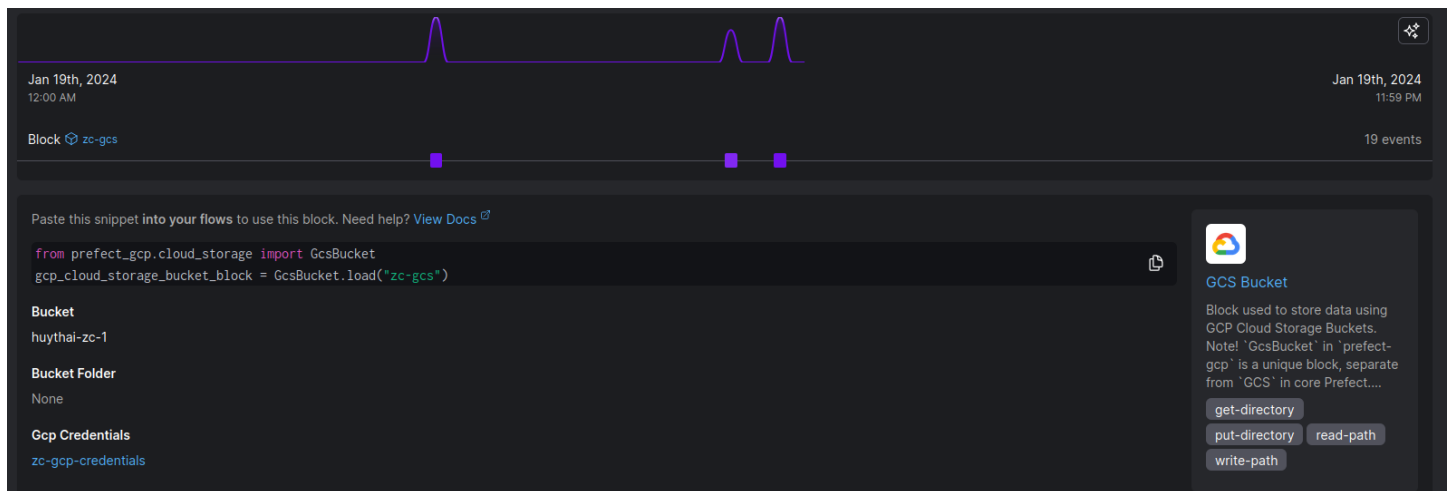
- Em đã deploy phần transform dbt và lập lịch cho nó tự động chạy vào lúc 11h30 pm hàng ngày.
- Mỗi lần chạy thì nó sẽ lấy data từ 2 bảng BigQuery đang có, transform và lưu vào 3 bảng mới (`cmt`, `post`, `joined`)

3.5 Tạo luồng và lập lịch với Prefect

- Set up

- tải xuống thư viện prefect
- tạo tài khoản prefect cloud (app.prefect.cloud)
- connect với Prefect API (lệnh `prefect cloud login` trên terminal)
- chạy lệnh `prefect block register -m prefect_gcp` để khởi tạo công kết nối prefect với Google Cloud
- Tạo Google Cloud Credentials để xác thực tài khoản google cloud (key services account)

- Tạo block GCS bucket để kết nối prefect với bucket GCS mà ta cần



- Chi tiết luồng

Trong Prefect thì mỗi luồng (Flow) được tập hợp từ nhiều task nhỏ

Trong project này em có làm 5 luồng:

- Luồng thu thập dữ liệu từ reddit và đẩy dữ liệu lên google cloud storage rde.parquet
- Luồng lấy data từ rde.parquet đưa lên google BigQuery
- Luồng thu thập dữ liệu từ reddit và đẩy dữ liệu lên google cloud storage cmt_rd.parquet
- Luồng lấy data từ cmt_rd.parquet đưa lên google BigQuery
- Luồng lấy data từ post text, post title và body từ 2 file rde.parquet và cmt_rd.parquet để tạo wordcloud
 - Sử dụng thư viện nltk, wordcloud
 - Lấy data từ GCS và chuyển thành dataframe pandas
 - Tạo 3 wordcloud cho mỗi trường trong 3 trường bên trên
 - Lấy các chữ tạo thành 1 python list
 - loại bỏ các stopword có trong stopwords English của nltk và bổ sung 1 số stopwords do em liệt kê
 - Tạo wordcloud với các chữ còn lại
 - upload ảnh wordcloud lên GCS

- Deploy và schedule

Các luồng trên đều được deploy lên prefect cloud và chạy với word pool là VM Google Cloud
4 luồng chính được schedule như trong hình:

<input type="checkbox"/>	Deployment name	Flow name	Schedule	Tags	Activity
<input type="checkbox"/>	cmt to bq ● Created 2024/01/10 01:16:02 AM	cmt-gcs-to-bq	At 10:30 PM every day		
<input type="checkbox"/>	posts_scraper ● Created 2024/01/09 11:24:07 PM	scrape-reddit	At 06:00 AM, 12:00 PM and 06:00 PM every day		
<input type="checkbox"/>	post to bq ● Created 2024/01/10 01:18:38 AM	post-gcs-to-bq	At 10:00 PM every day		
<input type="checkbox"/>	Scrape reddit cmt ● Created 2024/01/10 01:10:34 AM	scrape-reddit-comments	At 07:00 AM, 01:00 PM and 07:00 PM every day		
<input type="checkbox"/>	wordcloud plot ● Created 2024/01/19 12:33:22 PM	create-plots			

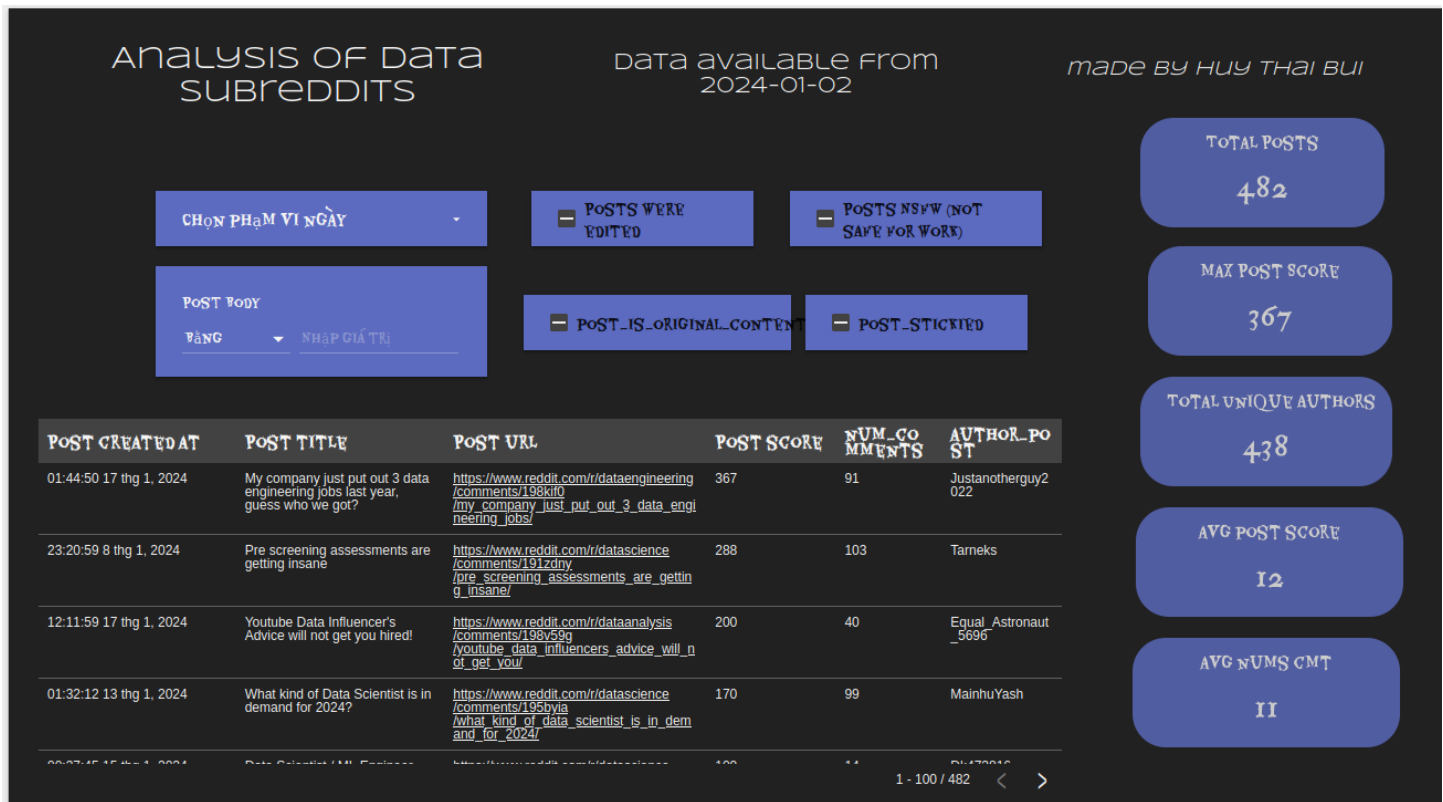
3.6 Phân tích dữ liệu với Google Data Studio

Trong project em tạo 2 dashboarch phân tích data với Google Data Studio (Looker):

a. Dashboard với data phân post (data sau khi xử lý với dbt):

Link tới dashboard:

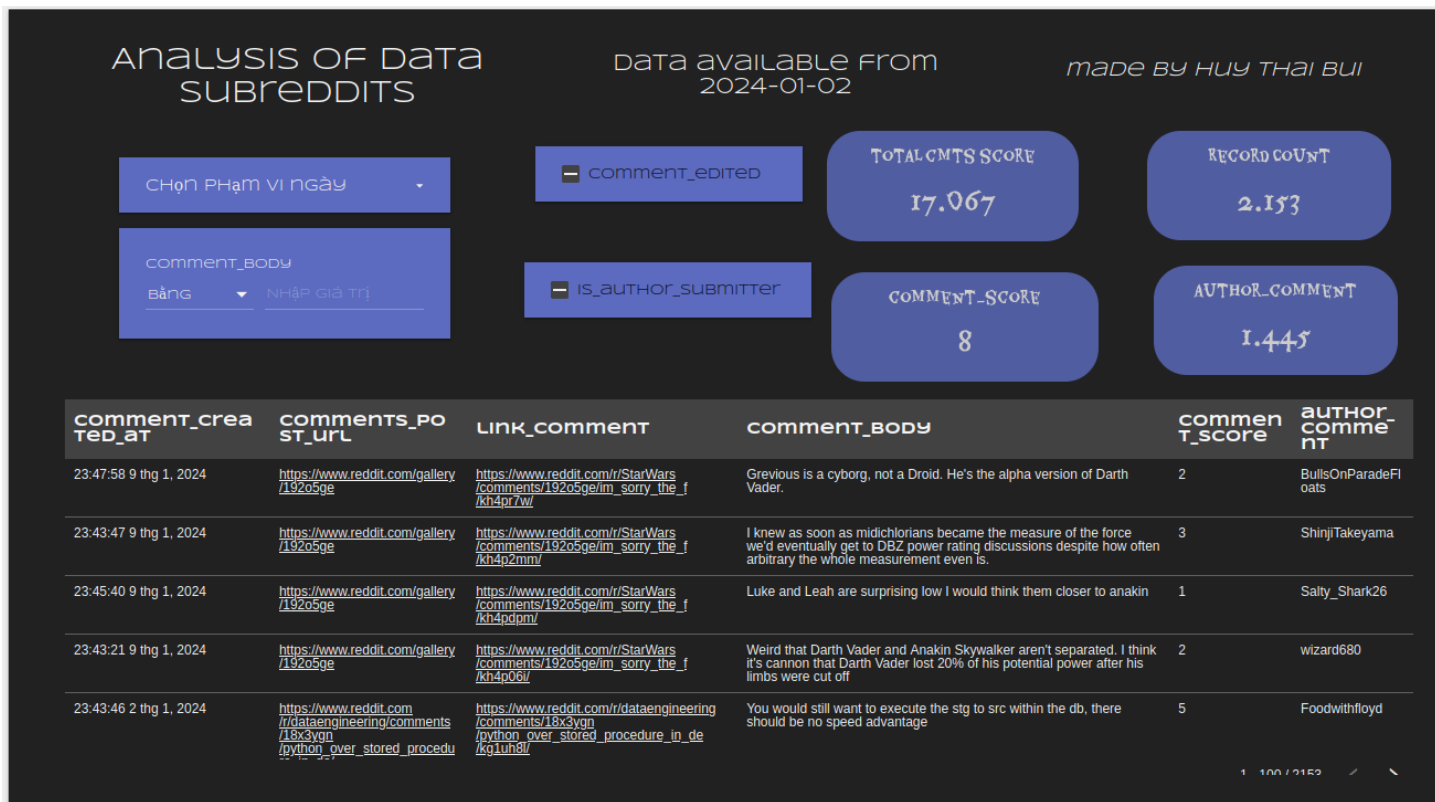
<https://lookerstudio.google.com/u/0/reporting/23452304-fc23-412e-aea7-ae8e7e1870eb/>



b. Dashboard với data comments (data sau khi xử lý với dbt)

Link tới dashboard:

<https://lookerstudio.google.com/u/0/reporting/8eb7f689-32fa-4c87-aa50-56f39015c59b/>



Chương 4: Khó khăn và kết quả

4.1 Khó khăn

- gặp vấn đề khi tạo schedule trên Prefect
 - o Nếu tạo schedule trên Prefect thì đến giờ đó Flow sẽ được chạy nhưng lúc đó máy cá nhân cần bật
 - o Tạo work pool trực tiếp với VM Google Cloud => không truy cập được file trên máy
 - o Đưa toàn bộ project lên VM => tạo work pool Hybrid => gặp vấn đề khi tắt terminal thì work pool bị terminate
 - o Chạy lệnh nohup để work pool tiếp tục chạy => giải quyết được vấn đề
- PRAW chỉ cho phép lấy data trong 1 thời gian ngắn và Reddit API bị giới hạn
- Lập lịch scrape data hàng ngày

4.2 Kết quả

- Tạo được thành công luồng dữ liệu như đã thiết kế
- Tạo được 2 dashboard phân tích dữ liệu cập nhật hàng ngày

4.3 Phát triển thêm

- Sử dụng dbt để tạo nhiều transform phức tạp hơn với data thu thập được

- Sử dụng data thu được để tạo model ML hoặc ứng dụng

Chương 5: Kết luận

Dự án xây dựng Data Pipeline Reddit đã thành công triển khai một quy trình tự động hoá và hiệu quả để thu thập dữ liệu từ Reddit, lưu trữ dữ liệu trên Google Cloud Storage và biến đổi dữ liệu sử dụng Google BigQuery và dbt. Sau đó, dữ liệu đã được trực quan hóa và khám phá thông qua Google Looker. Quy trình này cung cấp một hệ thống linh hoạt và mạnh mẽ cho việc phân tích dữ liệu Reddit.

Các bước chính trong dự án bao gồm:

1. Thu thập dữ liệu: Sử dụng Reddit API, dự án đã tạo một kết nối để truy xuất các bài đăng, bình luận và thông tin khác từ Reddit. Dữ liệu được thu thập và lưu trữ dưới dạng tệp tin trên Google Cloud Storage.
2. Lưu trữ dữ liệu: Google Cloud Storage đã được sử dụng làm nơi lưu trữ cho các tệp tin dữ liệu từ Reddit. Điều này đảm bảo tính bền vững và khả năng mở rộng trong việc lưu trữ dữ liệu.
3. Biến đổi dữ liệu: Sử dụng Google BigQuery và dbt, dữ liệu từ Google Cloud Storage đã được biến đổi và chuẩn hóa thành mô hình dữ liệu phù hợp. dbt đã giúp tạo và duy trì các phiên bản biến đổi dữ liệu, cung cấp một quy trình linh hoạt và kiểm soát cho việc phân tích dữ liệu.
4. Trực quan hóa dữ liệu: Với sự hỗ trợ của Google Looker, dữ liệu đã được trực quan hóa và khám phá thông qua các báo cáo, bảng điều khiển và biểu đồ. Người dùng có thể thực hiện phân tích tương quan và khám phá thông tin từ dữ liệu Reddit.
5. Quản lý luồng và lập lịch: Prefect đã được sử dụng để quản lý luồng công việc và lập lịch cho toàn bộ quy trình. Nó cung cấp khả năng kiểm soát và theo dõi công việc, tự động hóa quy trình và đảm bảo tính ổn định và tin cậy của hệ thống.

Tổng kết lại, dự án đã tạo ra một Data Pipeline Reddit toàn diện và tự động, từ việc thu thập dữ liệu từ Reddit, lưu trữ và biến đổi dữ liệu trên nền tảng Google Cloud, đến việc trực quan hóa và khám phá dữ liệu bằng Google Looker. Quy trình này mang lại giá trị phân tích và thông tin sâu sắc từ dữ liệu Reddit, cung cấp một cơ sở vững chắc cho quyết định và phân tích dựa trên dữ liệu.