

BÁO CÁO TUẦN

Tên nhân viên: Nguyễn Đức Anh

Chức vụ: Thực tập sinh

Phòng/ban: Phòng công nghệ thông tin

Tuần: Từ ngày 24/6/2024 đến 28/6/2024

1. Mục tiêu

- Tạo metrics trên Streamlit bằng cách lấy dữ liệu từ cloud function `getFunctionLogs`
- Tìm cách để gửi email lại cho user dựa trên error logs
- Tạo một rule để khi bị vi phạm thì function sẽ được trigger để gửi mail cho user
- Làm visualize trên looker

2. Tiến độ tuần trước

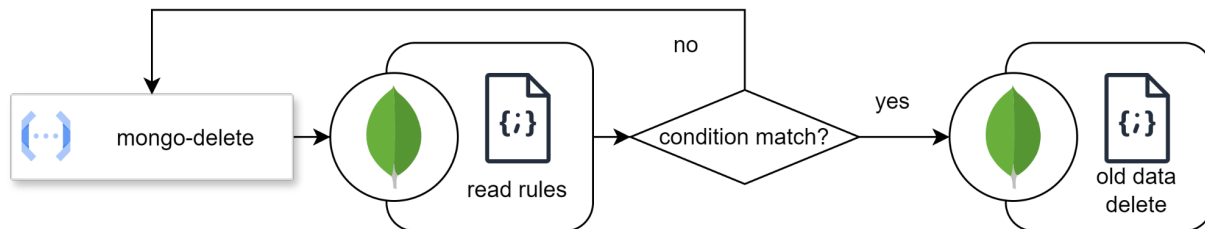


Fig 1.0: [Delete rule](#)

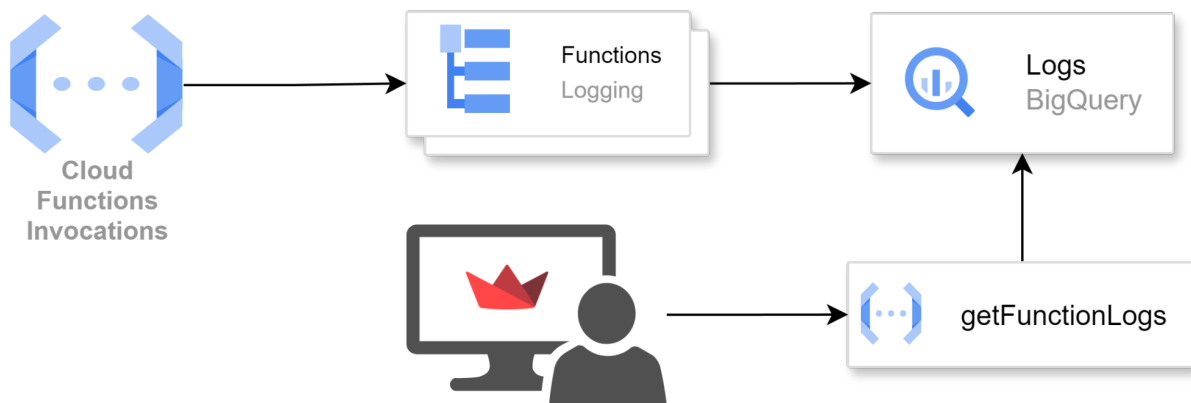


Fig 1.1: `getFunctionLogs` get metrics process



Fig 1.2: Streamlit MongoDB data lifecycle monitoring

3. Công việc hoàn thành trong tuần

- Terminal (streamlit) đã có thể coi được các metrics hoạt động của cloud function trên google cloud, bằng cách gọi cloud function `getFunctionLogs`, cloud function sẽ query data từ google big query(gbq), sau đó data sẽ được chuyển tới streamlit dưới định dạng JSON.

```
def getQuery(today,yesterday,rows):
    return f"""
    WITH raw AS(
      SELECT * FROM `project-finance-400806.logs.run_googleapis_com_requests_{today}`
      UNION ALL
      SELECT * FROM `project-finance-400806.logs.run_googleapis_com_requests_{yesterday}`
    ), sort_timestamp AS (
      SELECT * FROM raw
      ORDER BY timestamp DESC
    ), final_data as (
      SELECT
        resource.labels.service_name as function,
        round(httpRequest.latency,2) as latency,
        FORMAT_TIMESTAMP("%c",timestamp,"Asia/Ho_Chi_Minh") as time
      FROM sort_timestamp
    )
    SELECT * FROM final_data
    WHERE function != "getfunctionlogs"
    LIMIT {rows}
    """
```

Fig 1.3: `getFunctionLogs` querying data

Sau khi data được chuyển vào dataframe của streamlit, nó sẽ được chuyển hóa thành các live dashboard như sau:



Fig 1.4: Cloud function metrics (Invocations/Minutes & Latency)

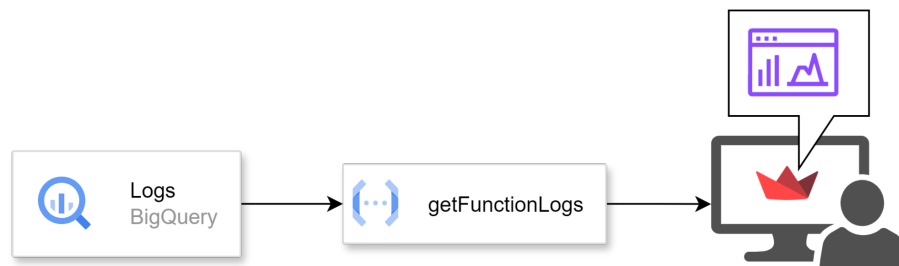


Fig 1.5: Dataflow

- b. Cloud function `emailHandler` được thiết kế để được gọi qua yêu cầu POST. Dữ liệu đầu vào cho yêu cầu POST bao gồm địa chỉ email người nhận (email khách hàng), chủ đề của email, và nội dung của email.

Khi được gọi qua yêu cầu POST, hàm cloud sẽ tạo một bản nháp email bao gồm địa chỉ email của khách hàng làm người nhận và kết hợp chủ đề cùng nội dung được cung cấp vào phần thân của email. Bản nháp email sẽ được định dạng để bao gồm một phần HTML với tiêu đề và nội dung để trình bày tốt hơn.

Dữ liệu đầu vào cho yêu cầu POST:

- `email`: Địa chỉ email của người nhận.
- `subject`: Chủ đề của email.
- `title`: Tiêu đề được bao gồm trong phần HTML của email.
- `content`: Nội dung chính của email.

```
def getDraft(email,subject,title,content):  
    return {  
        'FromEmail': "gauakanguyen@gmail.com",  
        'FromName': "Email Handler",  
        'Recipients': [  
            {  
                "Email": f"{email}",  
                "Name": "Client"  
            }  
        ],  
        'Subject': f"{subject}",  
        'Html-part': f"<h3>{title}</h3>{content}"  
    }  
}
```

Fig 1.6: Mail Draft



Email Handler gauakanguyen@gmail.com via bnc3.mailjet.com

to me ▾

Dear client,here's the error report!

col1	col2
1	4
2	3

Fig 1.7: Mail Sample