|  |
| --- |
|  |
| NodeRED Expansion |
| Reviewing possible addons with NodeRED |

|  |
| --- |
| Tran Nguyen Hoang Huy  8-7-2025 |

# Mục lục

[Mục lục 1](#_Toc205448924)

[1. Giới thiệu 2](#_Toc205448925)

[1.1. Mục tiêu nghiên cứu 2](#_Toc205448926)

[1.2. Phạm vi nghiên cứu 2](#_Toc205448927)

[1.3. Phương pháp nghiên cứu 2](#_Toc205448928)

[1.4. Công cụ nghiên cứu 2](#_Toc205448929)

[2. Hệ thống Grafana Loki 3](#_Toc205448930)

[2.1. Giới thiệu 3](#_Toc205448931)

[2.2. Phương pháp Logging 3](#_Toc205448932)

[3. Node Database 4](#_Toc205448933)

[3.1. Giới thiệu 4](#_Toc205448934)

[3.2. Phương pháp logging 4](#_Toc205448935)

[4. So sánh giữa phương thức Database và Loki 5](#_Toc205448936)

[5. Kết luận 5](#_Toc205448937)

[Reference 6](#_Toc205448938)

# 1. Giới thiệu

## 1.1. Mục tiêu nghiên cứu

Nghiên cứu này tập trung vào việc tìm hiểu cách tích hợp logging trong Node-RED thông qua một vài phương pháp. So sánh hai phương pháp lưu trữ log phổ biến: database và hệ thống quan sát log (Grafana Loki). Để đánh giá tính phù hợp của từng phương pháp trong ngữ cảnh xây dựng hệ thống cảnh báo theo luồng.

## 1.2. Phạm vi nghiên cứu

Phạm vi nghiên cứu bao gồm:

* Cơ chế ghi log trong Node-RED bằng cách sử dụng các node mặc định hoặc custom function.
* Thiết kế bảng ghi log trong MySQL để lưu lại thông tin như cấp độ cảnh báo, thời gian kích hoạt, mã sự kiện.
* Cấu hình logging handler tùy chỉnh để gửi log từ Node-RED đến Grafana Loki.
* So sánh hai phương pháp dựa trên yếu tố: thời gian thực, khả năng lưu trữ lâu dài, tính trực quan và mức độ tích hợp hệ thống.

## 1.3. Phương pháp nghiên cứu

Nghiên cứu sử dụng phương pháp thực hành triển khai và phân tích hệ thống. Tham khảo tài liệu chính thức của NodeRED, xây dựng một mô hình thử nghiệm gồm các flow trong NodeRED, mô phỏng việc lưu trữ và truy xuất log.

## 1.4. Công cụ nghiên cứu

* NodeRED
* MySQL
* SendGrid
* Postman
* ChatGPT
* Google
* Visual Studio Code

# 2. Hệ thống Grafana Loki

## 2.1. Giới thiệu

Grafana Loki là một hệ thống logging thời gian thực mã nguồn mở, được phát triển bởi Grafana Labs.

* Log được đẩy vào Loki theo định dạng dòng thời gian (time-series)
* Có thể được hiển thị trên giao diện Grafana
* Hỗ trợ tìm kiếm nhanh, lọc theo nhãn (labels)
* Dễ tích hợp với Promtail hoặc trực tiếp qua API

## 2.2. Phương pháp Logging

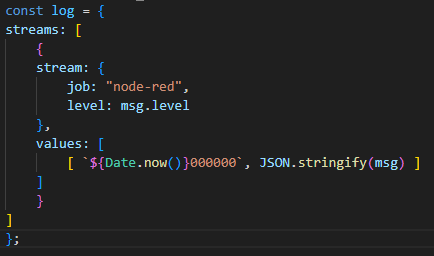
Vì Node-RED cho phép ta cấu hình thêm một hệ thống logging bên ngoài, khi NodeRED ghi log trong console thì các log này đồng thời có thể được gửi qua API đến **Grafana Loki**.

Tuy nhiên thì không phải mọi thứ đều được log lại nên ta cần sử dụng các phương thức dưới để đẩy thông tin đến log:

* **node.log():** Gứi log ở mức độ thông thường, được gán thẻ **info**
* **node.warn():** Gửi log ở mức độ cảnh báo, được gán thẻ **warn**
* **node.error():** Gửi log ở mức độ lỗi, được gán thẻ **error**

Ngoài ra thì khi log được chuyển sang dạng số với **log, warn, error** sẽ tương đương với level là **40, 30, 20**.

Ở phần cấu hình logger cho NodeRED, tùy theo thông tin ta cần log lại, còn có để đặt các label để khi cần truy vấn qua giao diện Grafana thì có thể sử dụng được, như ở đây sẽ có **label level** sẽ có mức độ log là **info, warn, error** như trên.



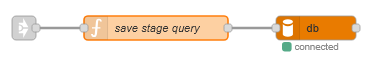
# 3. Node Database

## 3.1. Giới thiệu

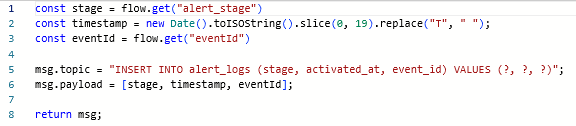
**Node MySQL** trong Node-RED cho phép kết nối đến hệ quản trị cơ sở dữ liệu MySQL. Qua đó, hệ thống có thể thực hiện các thao tác truy vấn, ghi dữ bằng cách gửi các câu query và thông tin cần thiết.

## 3.2. Phương pháp logging

* Tạo luồng có trigger, đến query, query sẽ được đẩy đến node DB



* Sử dụng Function node để tạo truy vấn SQL và dữ liệu log cần ghi.



* Dữ liệu được gửi tới node MySQL để ghi vào bảng **alert\_logs**.
* Mỗi bản ghi lưu trạng thái (**stage**), thời gian kích hoạt (**activated\_at**) và mã sự kiện (**event\_id**) tương ứng với cảnh báo.

# 4. So sánh giữa phương thức Database và Loki

|  |  |  |
| --- | --- | --- |
|  | **MySQL (Database Logging)** | **Grafana Loki (Real-time Logging)** |
| **Lưu trữ** | Local hoặc Sử dụng cloud database | Local hoặc Sử dụng cloud storage như Amazon S3 |
| **Gửi dữ liệu** | Cấu hình dữ liệu cần gửi + Một câu query thông qua node Function | Cấu hình logger trong NodeRED, tự động gửi khi có log. |
| **Quản lý** | Truy vấn dữ liệu qua query SQL | Truy vấn dữ liệu qua các label, hoặc điều kiện trong Grafana |

# 5. Kết luận

Trong quá trình nghiên cứu và triển khai hai phương pháp logging trong NodeRED sử dụng 2 phương pháp Logging là **Node Database** **(MySQL)** và **Grafana Loki.** Mỗi phương pháp đều thể hiện những ưu điểm và hạn chế riêng, phù hợp với các nhu cầu và mục tiêu khác nhau của hệ thống.

* **MySQL** mang lại khả năng lưu trữ có cấu trúc, dễ dàng truy vấn với ngôn ngữ SQL, phù hợp cho các hệ thống yêu cầu lưu trữ lâu dài, phân tích dữ liệu định kỳ, và có độ tin cậy cao trong việc bảo tồn dữ liệu. Tuy nhiên, nó yêu cầu thiết kế bảng dữ liệu chặt chẽ và không có khả năng xử lý log theo thời gian thực.
* **Grafana Loki**, ngược lại, tốt hơn khi hiển thị log theo thời gian thực với. Việc tích hợp với Node-RED thông qua custom logger cấu hình sẵn giúp đơn giản hóa quy trình triển khai. Tuy nhiên, Loki không phải là lựa chọn tối ưu nếu cần lưu trữ log lâu dài dưới dạng có cấu trúc hoặc tích hợp với các báo cáo dữ liệu dạng bảng.

Tùy vào **mục đích sử dụng**, ta có thể lựa chọn:

* **MySQL** phù hợp hơn cho viết quản lý, lưu trữ log.
* **Grafana Loki** cho các việc giám sát thời gian thực và cảnh báo.

Trong một số trường hợp, việc **kết hợp cả hai** giải pháp có thể mang lại hiệu quả toàn diện hơn: sử dụng Loki để xử lý log thời gian thực và ghi nhận sự kiện, đồng thời lưu các sự kiện quan trọng vào MySQL để phục vụ thống kê, báo cáo hoặc khôi phục dữ liệu.

# Reference

[NodeRED Logging](https://nodered.org/docs/user-guide/runtime/logging)

[Install Grafana Loki with Docker](https://grafana.com/docs/loki/latest/setup/install/docker/)

[Grafana Explore Logs](https://grafana.com/docs/grafana/latest/explore/logs-integration/)

[NodeRED MySQL node](https://flows.nodered.org/node/node-red-node-mysql)