TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN &  
TRUYỀN THÔNG VIỆT - HÀN

**KHOA KỸ THUẬT MÁY TÍNH & ĐIỆN TỬ**



ĐỒ ÁN CHUYÊN NGÀNH 2

**Xây dựng hệ thống giám sát và cảnh báo an toàn mạng sử dụng Prometheus và Grafana trong doanh nghiệp**

Sinh viên: **NGUYỄN AN PHÚC** Mã: 21IT639

**TRẦN VŨ LẬP THI** Mã: 21IT448

Giảng viên hướng dẫn: **THS.ĐẶNG QUANG HIỂN**

Đà Nẵng, tháng 8 năm 2025

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN &  
TRUYỀN THÔNG VIỆT - HÀN

**KHOA KỸ THUẬT MÁY TÍNH & ĐIỆN TỬ**

ĐỒ ÁN CHUYÊN NGÀNH 2

**XÂY DỰNG HỆ THỐNG GIÁM SÁT VÀ CẢNH BÁO AN TOÀN MẠNG SỬ DỤNG PROMETHEUS VÀ GRAFANA TRONG DOANH NGHIỆP**

Sinh viên: **NGUYỄN AN PHÚC** Mã: 21IT639

**TRẦN VŨ LẬP THI** Mã: 21IT448

Giảng viên hướng dẫn: **THS. ĐẶNG QUANG HIỂN**

Đà Nẵng, tháng 8 năm 2025

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

**…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………**

LỜI CẢM ƠN

Để đề tài đạt được kết quả như hiện tại, chúng em đã nhận được sự hỗ trợ, giúp đỡ, chỉ bảo tận tâm của thầy. Với tình cảm sâu sắc, chân thành, cho phép chúng em được bày tỏ lòng biết ơn sâu sắc đến tất cả quý thầy cô đã tạo điều kiện giúp đỡ chúng em trong quá trình học tập và nghiên cứu.

Trước hết, chúng em xin gửi tới đội ngũ giảng viên trường Đại học Công nghệ thông tin và truyền thông Việt – Hàn lời chào trân trọng, lời chúc sức khỏe và lời cảm ơn sâu sắc nhất. Với sự quan tâm, chỉ bảo tận tình chu đáo, cũng như vốn kiến thức về nhiều lĩnh vực của mình, đội ngũ giảng viên của trường đã truyền tải cho em được nhiều kiến thức, cũng như kỹ năng cần thiết, để đến ngày hôm nay, chúng em đã có thể hoàn thành được đề tài của mình.

Đặc biệt, em xin gửi lời cảm ơn chân thành tới thầy THS.ĐẶNG QUANG HIỂN, giảng viên đã quan tâm, giúp đỡ, chỉ bảo tận tình để chúng em có thể hoàn thành tốt đề tài này trong suốt thời gian vừa qua.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế, đề tài này không thể tránh được những thiếu sót. Chúng em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các thầy/cô, để chúng em có điều kiện bổ sung, nâng cao kiến thức, kỹ năng của mình, phục vụ tốt hơn cho công tác thực tế sau này.

Chúng em xin trân trọng cảm ơn!

*Sinh viên,*

Nguyễn An Phúc

Trần Vũ Lập Thi

**MỤC LỤC**

[LỜI CẢM ƠN ii](#_Toc204709694)

[DANH MỤC HÌNH ẢNH vi](#_Toc204709695)

[DANH MỤC CÁC TỪ VIẾT TẮT viii](#_Toc204709696)

[MỞ ĐẦU xi](#_Toc204709697)

[1. Giới thiệu xi](#_Toc204709698)

[2. Mục tiêu của đề tài xi](#_Toc204709699)

[3. Đối tượng, phạm vi và phương pháp tiếp cận xii](#_Toc204709700)

[4. Đóng góp của đề tài xii](#_Toc204709701)

[5. Bố cục của đề tài xiii](#_Toc204709702)

[CHƯƠNG 1: TỔNG QUAN 1](#_Toc204709703)

[1.1. Giới thiệu tổng quan về giám sát hệ thống mạng 1](#_Toc204709704)

[1.1.1. Giám sát mạng là gì? 1](#_Toc204709705)

[1.1.2. Ai cần giám sát mạng? 2](#_Toc204709706)

[1.1.3. Vai trò và tầm quan trọng của giám sát mạng trong doanh nghiệp 2](#_Toc204709707)

[1.2. Các thành phần cơ bản trong hệ thống giám sát mạng 3](#_Toc204709708)

[1.2.1. Thành phần chính 3](#_Toc204709709)

[1.2.2. Các loại giám sát 3](#_Toc204709710)

[1.2.3. Khác biệt giữa giám sát mạng và hệ thống phát hiện tấn công 4](#_Toc204709711)

[1.2.4. Các mô hình triển khai giám sát mạng phổ biến 5](#_Toc204709715)

[1.3. Các công cụ sử dụng để mô phỏng giám sát 5](#_Toc204709716)

[1.3.1. Prometheus – hệ thống giám sát và cảnh báo mạnh mẽ 5](#_Toc204709717)

[1.3.2. Grafana – nền tảng hiển thị dữ liệu trực quan 6](#_Toc204709718)

[1.3.3. VMware – Công cụ ảo hóa mạnh mẽ 7](#_Toc204709719)

[1.3.4. GNS3 – Mô phỏng mạng chuyên sâu 7](#_Toc204709720)

[1.4. Kết chương 1 8](#_Toc204709721)

[CHƯƠNG 2: NGHIÊN CỨU CÁC CÔNG NGHỆ PROMETHEUS VÀ GRAFANA 9](#_Toc204709722)

[2.1. Giới thiệu tổng quan về prometheus và grafana 9](#_Toc204709723)

[2.1.1. Kiến trúc và cơ chế hoạt động của Prometheus 9](#_Toc204709724)

[2.1.2. Thành phần chính của Prometheus 9](#_Toc204709725)

[2.1.3. Kiến trúc hoạt động của Prometheus 10](#_Toc204709726)

[2.1.4. Giới thiệu chi tiết về Grafana 10](#_Toc204709727)

[2.2. So sánh Prometheus – Grafana với các công cụ giám sát khác 11](#_Toc204709728)

[2.2.1. So sánh với Zabbix 11](#_Toc204709729)

[2.2.2. So sánh với Nagios 13](#_Toc204709730)

[2.3. Ưu điểm và nhược điểm của Prometheus – Grafana 15](#_Toc204709731)

[2.3.1. Ưu điểm: 15](#_Toc204709732)

[2.3.2. Nhược điểm: 15](#_Toc204709733)

[2.4. Ứng dụng trong doanh nghiệp 15](#_Toc204709734)

[2.5. Cấu hình và cài đặt Prometheus và Grafana 15](#_Toc204709735)

[2.5.1. Phần cứng tối thiểu 15](#_Toc204709736)

[2.5.2. Phần mềm 16](#_Toc204709737)

[2.5.3. Cài đặt Prometheus 16](#_Toc204709738)

[2.5.4. Cài đặt Node Exporter (giám sát hệ thống) 18](#_Toc204709739)

[2.5.5. Cài đặt Grafana 19](#_Toc204709740)

[2.6. Kết chương 2 20](#_Toc204709741)

[CHƯƠNG 3: XÂY DỰNG LAB TRIỂN KHAI 21](#_Toc204709742)

[3.1. Thiết kế mô hình phòng lab 21](#_Toc204709743)

[3.1.1. Mục tiêu triển khai 21](#_Toc204709744)

[3.1.2. Mô hình hệ thống thực nghiệm 21](#_Toc204709745)

[3.1.3. Phân vùng mạng chi tiết: 22](#_Toc204709746)

[3.2. Cấu hình triển khai mô phỏng 22](#_Toc204709747)

[3.2.1. Cấu hình Prometheus thu thập metric 22](#_Toc204709748)

[3.2.2. Tích hợp hệ thống và tạo Dashboard 27](#_Toc204709749)

[3.2.3. Cấu hình tích hợp alertmanager gửi cảnh báo 32](#_Toc204709750)

[3.3. Kết luận chương 39](#_Toc204709751)

[KẾT LUẬN 41](#_Toc204709752)

[1. Kết quả đạt được 41](#_Toc204709753)

[2. Những mặt đạt được và chưa đạt được 41](#_Toc204709755)

[3. Hướng nghiên cứu 42](#_Toc204709756)

[TÀI LIỆU THAM KHẢO 44](#_Toc204709757)

DANH MỤC HÌNH ẢNH

[Hình 1.1. Giám sát mạng là gì? 1](#_Toc204709503)

[Hình 1.2. Đối tượng giám sát mạng 2](#_Toc204709504)

[Hình 1.3. Các loại giám sát 3](#_Toc204709505)

[Hình 1.4. Mô hình triển khai giám sát 5](#_Toc204709506)

[Hình 1.5. Logo Prometheus 5](#_Toc204709507)

[Hình 1.6. Grafana 6](#_Toc204709508)

[Hình 1.7. VMware 7](#_Toc204709509)

[Hình 1.8. GNS3 chạy trên nền VMware 8](#_Toc204709510)

[Hình 2.1. Kiến trúc hoạt động của Prometheus 10](#_Toc204709511)

[Hình 2.2. Grafana DashBoard 11](#_Toc204709512)

[Hình 2.3. So sánh giữa Prometheus và Zabbix 12](#_Toc204709513)

[Hình 2.4. So sánh giữa Nagios và Prometheus 13](#_Toc204709514)

[Hình 2.5. ELK Stack (Elasticsearch – Logstash – Kibana) 14](#_Toc204709515)

[Hình 2.6. Tải Prometheus 16](#_Toc204709516)

[Hình 2.7. Tạo file cấu hình 17](#_Toc204709517)

[Hình 2.8. Tạo file service cho Prometheus 17](#_Toc204709518)

[Hình 2.9. Truy cập giao diện web 18](#_Toc204709519)

[Hình 2.10. Tạo service Node exporter 18](#_Toc204709520)

[Hình 2.11. Kiểm tra thu thập metrics 19](#_Toc204709521)

[Hình 2.12. Giao diện Grafana 19](#_Toc204709522)

[Hình 3.1. Sơ đồ mô hình hệ thống 21](#_Toc204709523)

[Hình 3.2. Kiểm tra trạng thái hoạt động của node\_exporter 23](#_Toc204709524)

[Hình 3.3. Kiểm tra cấu hình 24](#_Toc204709525)

[Hình 3.4. Cấu hình SNMP trên Pfsense 24](#_Toc204709526)

[Hình 3.5. Gom các metrics thành các module phù hợp 25](#_Toc204709527)

[Hình 3.6. Cấu hình file Prometheus.yml 26](#_Toc204709528)

[Hình 3.7. Kiểm tra trạng thái hoạt động của từng thiết bị 26](#_Toc204709529)

[Hình 3.8. Tích hợp Prometheus vào Grafana 27](#_Toc204709530)

[Hình 3.9. Import Dashboard có sẵn từ Grafana Lab 27](#_Toc204709531)

[Hình 3.10. Thông tin chung về hệ thống 28](#_Toc204709532)

[Hình 3.11. Hiệu suất 28](#_Toc204709533)

[Hình 3.12. Network traffic trên từng cổng 29](#_Toc204709534)

[Hình 3.13: Xem giám sát thông qua Grafana 30](#_Toc204709535)

[Hình 3.14. Giao diện tùy chỉnh 30](#_Toc204709536)

[Hình 3.15. Tổng lưu lượng mạng In/Out 31](#_Toc204709537)

[Hình 3.16. Bảng trạng thái từng cổng 31](#_Toc204709538)

[Hình 3.17. Hệ thống gửi cảnh báo Alertmanager 32](#_Toc204709539)

[Hình 3.18. Cấu hình alertmanager.yml 33](#_Toc204709540)

[Hình 3.19. Trạng thái dịch vụ Alertmanager 33](#_Toc204709541)

[Hình 3.20. Rule cảnh báo trạng thái cổng mạng 34](#_Toc204709542)

[Hình 3.21. Rule cảnh báo trạng backup khi không backup thành công 35](#_Toc204709543)

[Hình 3.22. Rule cảnh báo khi có lưu lượng In/Out tăng cao 35](#_Toc204709544)

[Hình 3.23. Rule gửi cảnh báo theo tình trạng mức sử dụng bộ nhớ 36](#_Toc204709545)

[Hình 3.24. Ví dụ về template mẫu của cảnh báo Up/Down trên cổng mạng 37](#_Toc204709546)

[Hình 3.25. Cảnh báo khi trạng thái cổng down 37](#_Toc204709547)

[Hình 3.26. Cảnh báo khi trạng thái cổng được khôi phục trở lại 38](#_Toc204709548)

[Hình 3.27. Cảnh báo khi backup failed 38](#_Toc204709549)

[Hình 3.28. Cảnh báo khi lượng memory đạt 74.55% 39](#_Toc204709550)

DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| TỪ VIẾT TẮT | ĐẦY ĐỦ | NỘI DUNG |
| IP | Internet Protocol | Giao thức định địa chỉ và truyền dữ liệu giữa các thiết bị mạng. |
| LAN | Local Area Network | Mạng cục bộ – kết nối các thiết bị trong phạm vi gần (như văn phòng, nhà ở). |
| WAN | Wide Area Network | Mạng diện rộng – kết nối các mạng LAN ở khoảng cách lớn. |
| NAT | Network Address Translation | Cơ chế chuyển đổi địa chỉ IP trong mạng nội bộ ra địa chỉ công cộng. |
| SNMP | Simple Network Management Protocol | Giao thức giám sát và quản lý thiết bị mạng. |
| MIB | Management Information Base | Cơ sở dữ liệu quản lý dùng trong SNMP để lưu thông tin thiết bị. |
| pfSense | Packet Filter Sense | Hệ điều hành tường lửa mã nguồn mở dùng để định tuyến và bảo mật mạng. |
| DNS | Domain Name System | Hệ thống phân giải tên miền thành địa chỉ IP. |
| DHCP | Dynamic Host Configuration Protocol | Giao thức cấp phát địa chỉ IP động cho các thiết bị. |
| GUI | Graphical User Interface | Giao diện người dùng đồ họa, dễ sử dụng. |
| RRD | Round Robin Database | Cơ sở dữ liệu dạng vòng dùng để lưu trữ số liệu theo thời gian (dùng cho biểu đồ). |
| ACL | Access Control List | Danh sách điều khiển truy cập để giới hạn quyền của người dùng hoặc dịch vụ. |
| IDS | Intrusion Detection System | Hệ thống phát hiện xâm nhập mạng. |
| IPS | Intrusion Prevention System | Hệ thống ngăn chặn xâm nhập mạng. |
| HA | High Availability | Độ sẵn sàng cao – đảm bảo hệ thống luôn hoạt động ngay cả khi có sự cố. |
| CPU | Central Processing Unit | Bộ xử lý trung tâm – thành phần chính xử lý dữ liệu. |
| RAM | Random Access Memory | Bộ nhớ truy xuất ngẫu nhiên – lưu trữ dữ liệu tạm thời. |
| VM | Virtual Machine | Máy ảo – mô phỏng máy tính chạy trên phần mềm ảo hóa. |
| API | Application Programming Interface | Giao diện lập trình ứng dụng – cho phép các phần mềm giao tiếp với nhau. |
| TLS | Transport Layer Security | Giao thức mã hóa bảo mật cho truyền thông mạng. |
| HTTP | HyperText Transfer Protocol | Giao thức truyền tải siêu văn bản – nền tảng của World Wide Web. |
| HTTPS | HyperText Transfer Protocol Secure | Phiên bản bảo mật của HTTP với mã hóa TLS/SSL. |
| Prometheus | Prometheus Monitoring System | Hệ thống giám sát mã nguồn mở thu thập và lưu trữ dữ liệu dạng chuỗi thời gian. |
| Grafana | Grafana Visualization Tool | Nền tảng trực quan hóa dữ liệu, xây dựng dashboard từ dữ liệu Prometheus và nhiều nguồn khác. |
| Alertmanager | Prometheus Alertmanager | Thành phần quản lý cảnh báo của Prometheus, hỗ trợ gửi thông báo đến nhiều kênh. |
| Exporter | Metric Exporter | Thành phần trung gian thu thập số liệu từ hệ thống hoặc thiết bị để gửi về Prometheus. |
| Node Exporter | Prometheus Node Exporter | Exporter phổ biến dùng để giám sát tài nguyên máy chủ (CPU, RAM, Disk). |
| SNMP Exporter | Prometheus SNMP Exporter | Exporter hỗ trợ thu thập số liệu từ thiết bị mạng qua SNMP. |
| Webhook | Webhook | Cơ chế gửi dữ liệu tự động từ hệ thống này sang hệ thống khác qua HTTP POST. |
| Backup | Backup Process | Quá trình sao lưu dữ liệu hoặc cấu hình hệ thống để phục hồi khi xảy ra sự cố. |
| Load Balancing | Load Balancing | Cân bằng tải – phân phối lưu lượng giữa nhiều máy chủ để tối ưu hiệu suất. |
| QoS | Quality of Service | Chính sách quản lý băng thông và ưu tiên dữ liệu trong mạng. |

MỞ ĐẦU

1. Giới thiệu

Trong thời đại số hóa và kết nối toàn cầu hiện nay, an toàn thông tin là một trong những vấn đề quan trọng hàng đầu đối với cá nhân, doanh nghiệp và tổ chức. Với sự phát triển mạnh mẽ của công nghệ thông tin, các mối đe dọa an ninh mạng ngày càng gia tăng cả về quy mô lẫn mức độ tinh vi, đòi hỏi các hệ thống mạng phải có những biện pháp bảo vệ và theo dõi liên tục để kịp thời phát hiện và xử lý các sự cố bảo mật.

Bên cạnh các giải pháp phòng thủ chủ động như tường lửa, hệ thống phát hiện xâm nhập (IDS/IPS), thì việc giám sát và cảnh báo an toàn mạng theo thời gian thực là yếu tố then chốt giúp đảm bảo tính sẵn sàng và ổn định của hệ thống CNTT. Trong đó, các công cụ mã nguồn mở như Prometheus và Grafana đã và đang được ứng dụng rộng rãi nhờ khả năng thu thập dữ liệu linh hoạt, trực quan hóa mạnh mẽ và dễ dàng mở rộng, phù hợp với nhiều quy mô doanh nghiệp.

Nhận thấy tầm quan trọng của việc xây dựng hệ thống giám sát an toàn mạng, chúng em đã lựa chọn thực hiện đề tài: "Xây dựng hệ thống giám sát và cảnh báo an toàn mạng sử dụng Prometheus và Grafana trong doanh nghiệp". Mục tiêu của đề tài là nghiên cứu, thiết kế và triển khai mô hình giám sát mạng với khả năng thu thập dữ liệu, theo dõi trạng thái hệ thống và gửi cảnh báo khi xảy ra sự cố. Qua đó, chúng em mong muốn nâng cao kiến thức về giám sát hệ thống, bảo mật mạng cũng như góp phần cung cấp giải pháp hiệu quả và chi phí thấp cho doanh nghiệp trong công tác đảm bảo an toàn thông tin.

1. Mục tiêu của đề tài

Nhằm hoàn tất chương trình học và mở rộng kiến thức thực tiễn trong lĩnh vực giám sát và bảo mật hệ thống mạng, em đã lựa chọn đề tài này với các mục tiêu cụ thể như sau:

* Tìm hiểu tổng quan về Prometheus và Grafana: Khảo sát kiến trúc, tính năng chính và khả năng ứng dụng của hai công cụ mã nguồn mở này trong giám sát hệ thống mạng và máy chủ.
* Nghiên cứu phương pháp thu thập dữ liệu và cảnh báo: Phân tích cách Prometheus thu thập số liệu từ các Exporter, tích hợp Alertmanager để xử lý cảnh báo, và khả năng gửi thông báo qua email, Telegram hoặc các nền tảng khác.
* Thực hành triển khai hệ thống giám sát: Cài đặt và cấu hình Prometheus, Grafana cùng với các Exporter (Node Exporter, Blackbox Exporter...) trong môi trường doanh nghiệp mô phỏng, từ đó xây dựng bảng điều khiển trực quan và thiết lập cảnh báo theo thời gian thực.
* Đánh giá hiệu quả hệ thống giám sát: Phân tích khả năng phát hiện sự cố, hiệu suất thu thập dữ liệu và mức độ hỗ trợ vận hành hệ thống mạng an toàn, ổn định khi sử dụng bộ công cụ Prometheus và Grafana.

1. Đối tượng, phạm vi và phương pháp tiếp cận

**Đối tượng nghiên cứu:** Đề tài tập trung nghiên cứu hệ thống giám sát và cảnh báo an toàn mạng sử dụng các công cụ mã nguồn mở **Prometheus** và **Grafana**, kết hợp với các thành phần hỗ trợ như **Alertmanager** và các loại **Exporter**. Mục tiêu là áp dụng các công cụ này để xây dựng một hệ thống giám sát hiệu quả cho hạ tầng mạng trong môi trường doanh nghiệp.

**Phạm vi nghiên cứu:**

* Tìm hiểu về Prometheus và Grafana: Nghiên cứu kiến trúc, cơ chế hoạt động và tính năng chính của hai công cụ.
* Triển khai hệ thống giám sát mạng: Cài đặt và cấu hình Prometheus, Grafana và các Exporter trong môi trường mạng doanh nghiệp mô phỏng.
* Thiết lập hệ thống cảnh báo: Tích hợp Alertmanager để gửi cảnh báo khi phát hiện các dấu hiệu bất thường trong hệ thống.
* Đánh giá hiệu quả hoạt động: Kiểm tra khả năng giám sát, cảnh báo và trực quan hóa dữ liệu của hệ thống qua các tình huống giả lập và thực tế.

1. Đóng góp của đề tài

Sau quá trình nghiên cứu và triển khai, đề tài dự kiến sẽ đạt được các kết quả sau:

* Hoàn thành báo cáo chi tiết về hệ thống giám sát và cảnh báo an toàn mạng sử dụng Prometheus và Grafana, bao gồm cả phần lý thuyết nền tảng lẫn triển khai thực tế.
* Xây dựng một hệ thống giám sát mạng hoàn chỉnh, có khả năng thu thập dữ liệu, trực quan hóa hiệu suất hoạt động của hệ thống, và gửi cảnh báo kịp thời khi phát hiện bất thường.
* Đánh giá hiệu quả của hệ thống giám sát trong việc hỗ trợ phát hiện sự cố, nâng cao khả năng vận hành ổn định và đảm bảo an toàn thông tin mạng trong doanh nghiệp.
* Cung cấp tài liệu hướng dẫn triển khai, cấu hình và sử dụng Prometheus – Grafana phục vụ cho các cá nhân, tổ chức đang quan tâm đến giải pháp giám sát và bảo mật hệ thống mạng với chi phí tối ưu.

1. Bố cục của đề tài

Sau phần Mở đầu, báo cáo được trình bày trong ba chương, cụ thể như sau:

***Chương 1: Tổng quan về hệ thống giám sát mạng***

**Chương này trình bày các khái niệm cơ bản và tổng quan về hệ thống giám sát và cảnh báo an toàn mạng. Nội dung bao gồm vai trò của giám sát trong bảo mật hệ thống, các mô hình giám sát phổ biến, so sánh một số công cụ mã nguồn mở và thương mại. Đặc biệt tập trung giới thiệu về Prometheus và Grafana – hai công cụ được lựa chọn triển khai trong đề tài.**

***Chương 2: Nghiên cứu kiến trúc và công nghệ Prometheus – Grafana***

**Chương này đi sâu vào nghiên cứu kiến trúc hoạt động của Prometheus và Grafana, các thành phần chính như Exporter, Alertmanager, cơ chế thu thập dữ liệu, tạo cảnh báo và trực quan hóa. Đồng thời trình bày cách thức tích hợp giữa các công cụ này trong hệ thống mạng doanh nghiệp, cùng với một số ví dụ triển khai thực tế.**

***Chương 3: Xây dựng và triển khai hệ thống giám sát mạng***

**Chương này trình bày quá trình xây dựng mô hình giám sát mạng hoàn chỉnh bằng Prometheus và Grafana trong môi trường mô phỏng. Bao gồm các bước cài đặt, cấu hình Exporter, thiết lập cảnh báo qua Alertmanager, tạo dashboard hiển thị trạng thái hệ thống. Ngoài ra, chương này cũng trình bày các tình huống thử nghiệm và đánh giá hiệu quả hoạt động của hệ thống giám sát.**

**Cuối cùng là phần Kết luận, Tài liệu tham khảo và Phụ lục phục vụ minh họa và hỗ trợ kỹ thuật cho quá trình triển khai.**

# TỔNG QUAN

## Giới thiệu tổng quan về giám sát hệ thống mạng

### Giám sát mạng là gì?

Giám sát mạng là một hệ thống quan trọng trong việc đảm bảo hiệu suất và hoạt động của mạng trong một doanh nghiệp. Nó cung cấp thông tin về tình trạng mạng, cho phép quản trị viên phân tích và giải quyết các vấn đề liên quan đến hiệu suất, hoạt động và khả dụng của mạng.



Hình 1.1. Giám sát mạng là gì?

Giám sát mạng được hiểu đơn giản là quá trình theo dõi và kiểm tra các hoạt động trên hệ thống mạng. Điều này bao gồm việc theo dõi lưu lượng mạng, thời gian truy cập và các hoạt động khác để đảm bảo an toàn cho dữ liệu và mạng. Trong bài viết này, chúng ta sẽ tìm hiểu chi tiết về những ưu điểm, nhược điểm, và cách triển khai nó hiệu quả.

Các thông số giám sát mạng: thông lượng, tỷ lệ lỗi, mất gói và độ trễ, cũng như tính khả dụng của các thiết bị như router và switch. Quản trị viên mạng nhận được thông báo về các lỗi thông qua biểu ngữ cảnh báo, email, điện thoại và các phương tiện khác để kịp thời phát hiện và giải quyết sự cố.

Nó có vai trò chiến lược trong các doanh nghiệp bằng cách tối ưu hóa luồng dữ liệu, phát hiện các thiết bị không đáng tin cậy và xác minh năng lực, điều kiện hoạt động của các thiết bị. Điều này giúp tối đa hoá hiệu suất mạng và giảm thiểu các lỗi tiềm ẩn.

Mạng được tối ưu hóa qua việc giám sát mang lại nhiều lợi ích cho doanh nghiệp. Nó giúp giảm chi phí cấu trúc cơ sở hạ tầng, tăng năng suất của nhân viên và đảm bảo sự nhanh chóng và đáng tin cậy của luồng dữ liệu.

### Ai cần giám sát mạng?

Giám sát mạng là cần thiết đối với tất cả các doanh nghiệp và tổ chức sử dụng hệ thống mạng. Bất kỳ doanh nghiệp nào muốn bảo vệ dữ liệu của họ khỏi những nguy cơ mạng như tấn công phần mềm độc hại, tin tặc hoặc vi phạm bảo mật phải có một giải pháp để đảm bảo an toàn cho thông tin của họ.



Hình 1.2. Đối tượng giám sát mạng

### ****Vai trò và tầm quan trọng của giám sát mạng trong doanh nghiệp****

Phát hiện bất thường, sự cố hệ thống nhanh chóng.

Giảm thời gian gián đoạn dịch vụ.

Cải thiện hiệu suất vận hành và bảo mật.

Là công cụ không thể thiếu trong quản trị hệ thống CNTT hiện đại.

## Các thành phần cơ bản trong hệ thống giám sát mạng

### Thành phần chính

Một hệ thống giám sát mạng thường bao gồm các thành phần chính sau:

* Tác nhân giám sát (Agent): Được cài đặt trên các máy chủ hoặc thiết bị mạng để thu thập dữ liệu (CPU, RAM, Disk, Network...).
* Máy chủ giám sát (Monitoring Server): Là nơi lưu trữ, xử lý dữ liệu thu thập và hiển thị thông tin trên giao diện quản trị.
* Cơ sở dữ liệu (Database): Lưu trữ dữ liệu thời gian thực hoặc dữ liệu lịch sử phục vụ truy vấn và phân tích.
* Hệ thống cảnh báo (Alerting): Gửi cảnh báo qua email, tin nhắn, webhook… khi phát hiện sự cố hoặc điều kiện vượt ngưỡng.
* Giao diện trực quan (Dashboard): Hiển thị các thông số dưới dạng biểu đồ, đồ thị giúp người quản trị dễ dàng theo dõi.

### Các loại giám sát

Giám sát dựa trên SNMP: Sử dụng giao thức SNMP (Simple Network Management Protocol) để thu thập thông tin từ các thiết bị mạng như router, switch, máy chủ. Các công cụ này giúp giám sát, quản lý hiệu suất và sự hoạt động của hệ thống.



Hình 1.3. Các loại giám sát

Giám sát dựa trên luồng (Flow-based): Sử dụng thông tin từ luồng dữ liệu mạng như NetFlow, sFlow hoặc IPFIX để phân tích và giám sát lưu lượng mạng. Các công cụ này giúp xác định nguồn gốc và đích của lưu lượng, phân tích tài nguyên mạng, đồng thời phát hiện, ngăn chặn các cuộc tấn công mạng.

Active network monitoring: Sử dụng các công cụ và kỹ thuật như ping, traceroute, SNMP polling để giám sát trạng thái, hoạt động của các thiết bị mạng và dịch vụ. Các công cụ này thường kiểm tra định kỳ, tự động thông báo về sự cố và điều kiện không bình thường trong mạng.

Ngoài ra còn có cả phương pháp giám sát mạng có Agent và không có Aquan đến việc cài đặt một Agent, một ứng dụng hoặc một phần mềm trên thiết bị được giám sát. Giám sát không cần Agent (sử dụng giao thức SNMP) không cần cài đặt; thay vào đó, phần mềm giám sát sẽ ghi trực tiếp vào thiết bị được giám sát.

Các loại công cụ này đều có vai trò quan trọng trong việc giám sát, quản lý mạng và hệ thống. Tùy thuộc vào mục tiêu và yêu cầu cụ thể, người dùng có thể lựa chọn công cụ phù hợp cho nhu cầu của mình.

### Khác biệt giữa giám sát mạng và hệ thống phát hiện tấn công

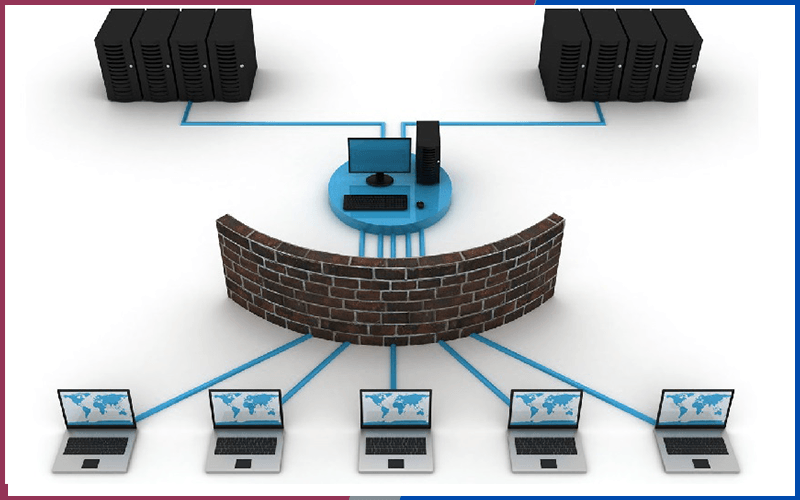
|  |  |  |
| --- | --- | --- |
| **Tiêu chí** | **Giám sát mạng (NMS)** | **Phát hiện tấn công (IDS/IPS)** |
| Mục tiêu chính | Theo dõi hiệu suất, cảnh báo sự cố | Phát hiện và ngăn chặn truy cập trái phép |
| Dữ liệu phân tích | Thông số hệ thống, lưu lượng mạng | Mẫu tấn công, hành vi bất thường |
| Tác động | Không can thiệp trực tiếp vào luồng dữ liệu | Có thể chặn kết nối khi cần thiết |
| Vị trí triển khai | Máy chủ, router, switch, dịch vụ | Cổng mạng, tường lửa, gateway |

Cả hai hệ thống đều đóng vai trò quan trọng và thường được kết hợp để tạo nên một nền tảng giám sát – bảo vệ toàn diện.

### Các mô hình triển khai giám sát mạng phổ biến

Tùy theo quy mô và kiến trúc hệ thống, có thể lựa chọn các mô hình triển khai như:

* Triển khai tập trung: Một máy chủ giám sát chính thu thập toàn bộ dữ liệu từ các agent.
* Triển khai phân tán: Nhiều máy chủ giám sát hoạt động tại từng khu vực/chi nhánh, đồng bộ về máy chủ trung tâm.
* Triển khai hybrid (lai): Kết hợp giữa tập trung và phân tán, phù hợp cho hệ thống nhiều tầng, đa đám mây.



Hình 1.4. Mô hình triển khai giám sát

Các mô hình này có thể được tích hợp cùng công cụ cảnh báo, tự động hóa phản ứng sự cố, hoặc phân tích dữ liệu nâng cao.

## Các công cụ sử dụng để mô phỏng giám sát

### Prometheus – hệ thống giám sát và cảnh báo mạnh mẽ



Hình 1.5. Logo Prometheus

Prometheus là một hệ thống giám sát mã nguồn mở, ban đầu được phát triển bởi SoundCloud và hiện là một phần của Cloud Native Computing Foundation (CNCF). Nó hỗ trợ:

Thu thập dữ liệu theo mô hình pull từ các endpoint.

Lưu trữ chuỗi thời gian (time series) hiệu quả.

Cung cấp ngôn ngữ truy vấn PromQL mạnh mẽ.

Tích hợp hệ thống cảnh báo Alertmanager để gửi email, Slack, webhook...

Prometheus đặc biệt thích hợp với hệ thống có số lượng lớn dịch vụ và yêu cầu giám sát linh hoạt, mở rộng.

### Grafana – nền tảng hiển thị dữ liệu trực quan

Grafana là công cụ trực quan hóa dữ liệu phổ biến, có thể kết nối với nhiều nguồn dữ liệu như Prometheus, InfluxDB, MySQL,... Các tính năng nổi bật:

Hiển thị dữ liệu thời gian thực với biểu đồ linh hoạt.

Tùy chỉnh dashboard theo từng đối tượng quản trị.

Hỗ trợ cảnh báo trực quan ngay trên giao diện đồ họa.

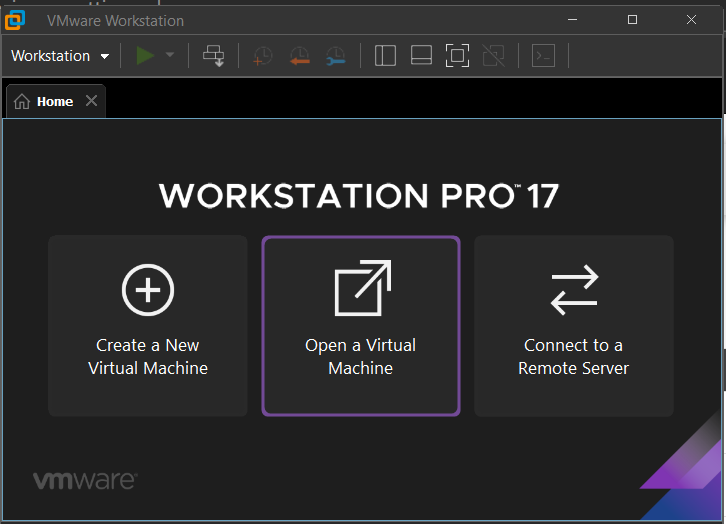


Hình 1.6. Grafana

Sự kết hợp giữa Prometheus và Grafana tạo nên một nền tảng giám sát mạng hiệu quả: thu thập – phân tích – hiển thị – cảnh báo, tất cả đều trên mã nguồn mở, miễn phí và có khả năng mở rộng cao.

### VMware – Công cụ ảo hóa mạnh mẽ

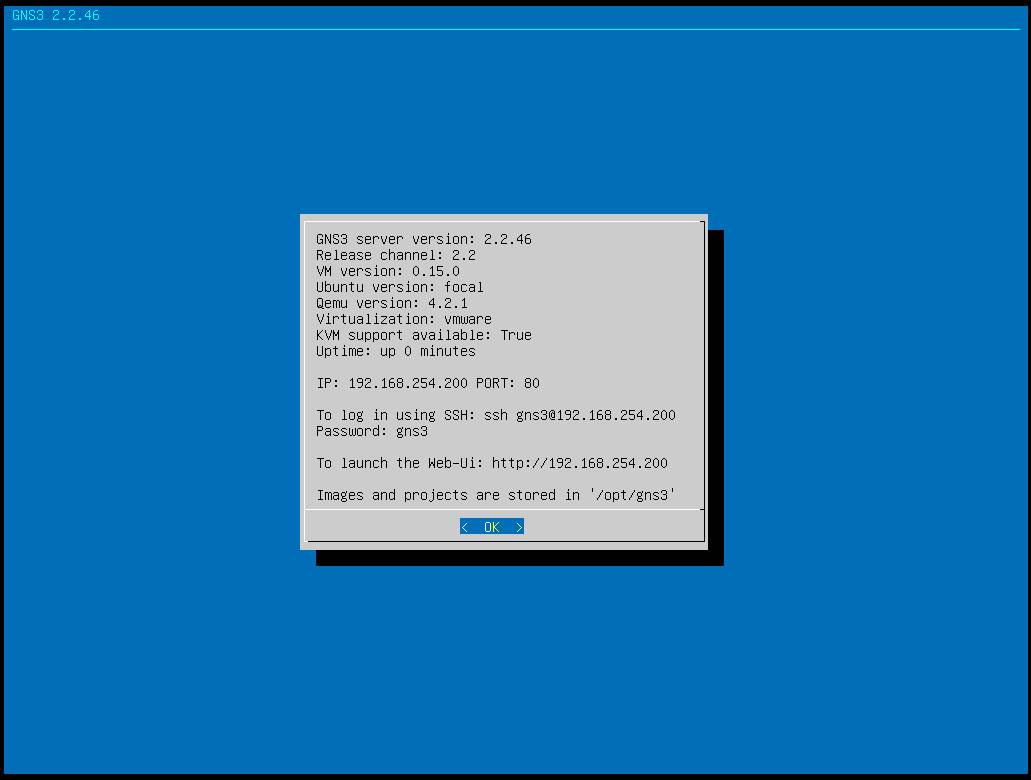
VMware là phần mềm ảo hóa chuyên nghiệp, cho phép tạo máy ảo chạy GNS3 để kiểm thử và triển khai trong môi trường mạng thực tế.



Hình 1.7. VMware

### GNS3 – Mô phỏng mạng chuyên sâu

GNS3 là công cụ mô phỏng mạng mạnh mẽ, giúp mô phỏng các thiết bị mạng và kiểm thử pfSense trong môi trường giả lập.



Hình 1.8. GNS3 chạy trên nền VMware

## Kết chương 1

Chương 1 đã trình bày cái nhìn tổng quan về hệ thống giám sát và cảnh báo mạng trong doanh nghiệp, làm rõ tầm quan trọng của việc giám sát hiệu suất và trạng thái hạ tầng mạng để đảm bảo hoạt động ổn định và an toàn. Nội dung đã phân tích các thành phần cơ bản của hệ thống giám sát, sự khác biệt với hệ thống phát hiện xâm nhập, cũng như các mô hình triển khai phổ biến hiện nay.

Đặc biệt, chương này đã giới thiệu hai công cụ nổi bật là Prometheus và Grafana, được đánh giá cao nhờ khả năng thu thập dữ liệu thời gian thực, trực quan hóa linh hoạt và hỗ trợ cảnh báo hiệu quả. Sự kết hợp của hai công cụ này mang lại giải pháp mạnh mẽ và phù hợp cho nhiều môi trường doanh nghiệp.

Những kiến thức nền tảng này sẽ là cơ sở quan trọng để đi sâu vào nghiên cứu kiến trúc, công nghệ và quá trình triển khai hệ thống giám sát – cảnh báo cụ thể bằng Prometheus và Grafana, được trình bày ở các chương tiếp theo.

# NGHIÊN CỨU CÁC CÔNG NGHỆ PROMETHEUS VÀ GRAFANA

## Giới thiệu tổng quan về prometheus và grafana

Prometheus và Grafana là hai công cụ mã nguồn mở phổ biến được sử dụng trong giám sát hạ tầng CNTT và cảnh báo mạng.

Prometheus đóng vai trò thu thập và lưu trữ dữ liệu dạng chuỗi thời gian (time series), đồng thời cung cấp cơ chế cảnh báo linh hoạt thông qua Alertmanager.

Grafana đảm nhiệm việc trực quan hóa dữ liệu, hiển thị biểu đồ và dashboard, giúp người quản trị dễ dàng theo dõi tình trạng hệ thống.

Sự kết hợp của hai công cụ này mang đến giải pháp giám sát hiện đại, nhẹ, linh hoạt và dễ mở rộng, phù hợp cho cả doanh nghiệp vừa và lớn.

### ****Kiến trúc và cơ chế hoạt động của Prometheus****

Prometheus được dùng để giám sát hệ thống thông qua các daemon cài sẵn trên các node, qua đó thu thập các thông tin cần thiết. Prometheus giao tiếp với node qua giao thức http/https và lưu trữ data theo dạng time-series database (TSDB).

Time series database (TSDB) là những database được lưu trữ theo các mốc thời gian. Nghĩa là mỗi dữ liệu luôn được gắn với một mốc thời gian nhất đinh từ đó tạo thành chuỗi dữ liêu theo thời gian từ đó giúp chúng ta có thể xem lại dữ liệu cho một khoảng thời gian 1 giờ hay 1 tiếng.

### ****Thành phần chính của Prometheus****

Prometheus bao gồm các thành phần cốt lõi sau:

**Prometheus Server:** Thành phần trung tâm, chịu trách nhiệm thu thập, lưu trữ dữ liệu và xử lý truy vấn PromQL.

**Exporter:** Ứng dụng hoặc plugin giúp xuất dữ liệu từ hệ thống hoặc dịch vụ (như Node Exporter cho máy chủ, Blackbox Exporter cho HTTP/TCP ping, v.v.).

**Alertmanager:** Thành phần quản lý cảnh báo, gửi thông báo qua email, Slack, webhook,… khi xảy ra sự cố.

**Pushgateway (tùy chọn):** Cho phép nhận dữ liệu từ các ứng dụng không thể bị Prometheus “pull” trực tiếp, ví dụ job chạy ngắn hạn.

**PromQL (Prometheus Query Language):** Ngôn ngữ truy vấn dữ liệu mạnh mẽ để phân tích và lọc thông tin theo yêu cầu.

### ****Kiến trúc hoạt động của Prometheus****



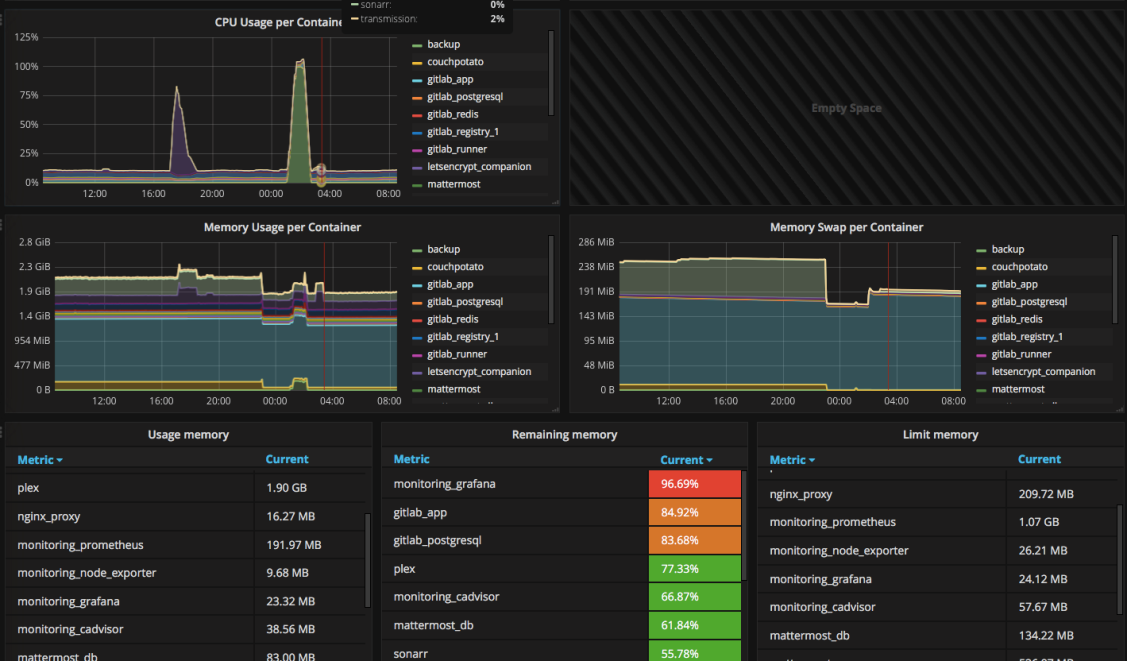
Hình 2.1. Kiến trúc hoạt động của Prometheus

Kiến trúc của Prometheus hoạt động theo mô hình **pull-based**:

* Prometheus Server định kỳ “kéo” dữ liệu từ các exporter thông qua HTTP endpoint.
* Dữ liệu được lưu trữ dưới dạng chuỗi thời gian trong cơ sở dữ liệu nội bộ TSDB (Time Series Database).
* Khi xảy ra sự kiện vượt ngưỡng đã cấu hình, Alertmanager sẽ gửi cảnh báo tới người quản trị.
* Người dùng sử dụng PromQL để truy vấn hoặc kết hợp với Grafana để hiển thị trực quan.

### ****Giới thiệu chi tiết về Grafana****

Grafana là một vizualizer hiển thị các metric dưới dạng các biểu đồ (chart) hoặc đồ thị (graph), được tập hợp lại thành dashboard có tính tùy biến cao, giúp dễ dàng theo dõi tình trạng của node. Đơn giản cho các bạn dễ hiểu là sau khi lấy được metric từ các thiết bị, grafana sẽ sử dụng metric đó để phân tích và tạo ra dashboard mô tả trực quan các metric cần thiết cho việc monitoring như CPU, RAM, disks, IO operations...



Hình 2.2. Grafana DashBoard

#### **Chức năng chính**

Tích hợp nhiều nguồn dữ liệu: Prometheus, InfluxDB, MySQL, ElasticSearch,…

Trực quan hóa dữ liệu bằng dashboard tùy chỉnh.

Cung cấp cơ chế cảnh báo ngay trong giao diện đồ họa.

Hỗ trợ quản lý người dùng, phân quyền và chia sẻ dashboard.

#### **Vai trò trong hệ thống giám sát**

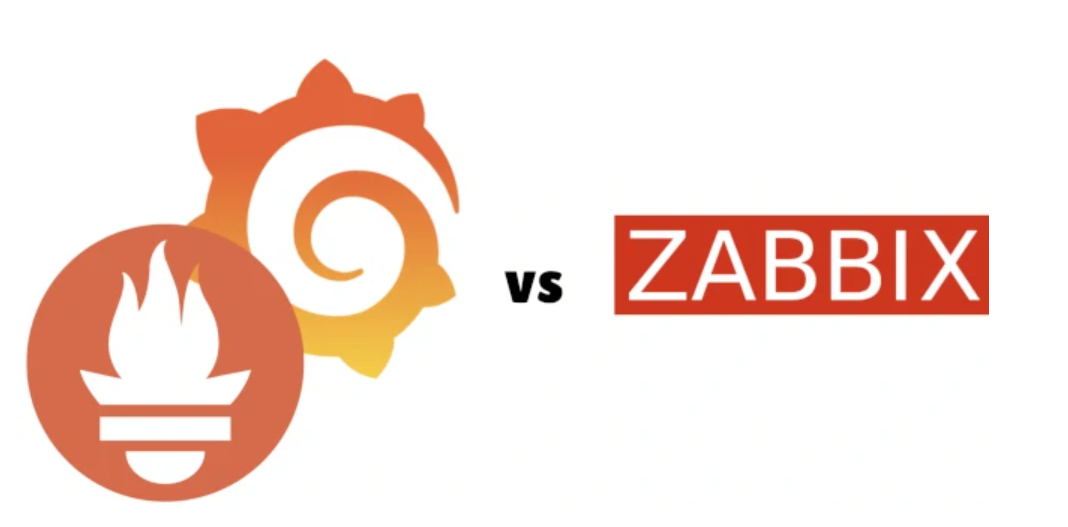
Grafana không thực hiện việc thu thập dữ liệu, mà đóng vai trò hiển thị thông tin từ các nguồn dữ liệu khác nhau. Trong mô hình Prometheus – Grafana:

* Prometheus chịu trách nhiệm lưu trữ và cung cấp dữ liệu.
* Grafana truy vấn dữ liệu từ Prometheus và hiển thị biểu đồ trực quan.

## ****So sánh Prometheus – Grafana với các công cụ giám sát khác****

### ****So sánh với Zabbix****

Zabbix là nền tảng giám sát truyền thống, hỗ trợ cả agent-based và agentless (SNMP, IPMI). Nó tích hợp đầy đủ tính năng từ thu thập dữ liệu, cảnh báo đến trực quan hóa mà không cần công cụ bổ trợ.



Hình 2.3. So sánh giữa Prometheus và Zabbix

**Thu thập dữ liệu:**

* **Prometheus sử dụng cơ chế pull (server chủ động thu thập dữ liệu từ exporter).**
* **Zabbix chủ yếu dùng cơ chế push (agent gửi dữ liệu về server).**

**Trực quan hóa:**

* **Grafana cung cấp dashboard cực kỳ linh hoạt và đẹp, hỗ trợ nhiều plugin.**
* **Zabbix có giao diện tích hợp nhưng ít tùy biến và kém hiện đại.**

**Cảnh báo:**

* **Prometheus cần Alertmanager để quản lý cảnh báo.**
* **Zabbix tích hợp sẵn cảnh báo trong cùng hệ thống.**

**Triển khai:**

* **Prometheus – Grafana phù hợp môi trường cloud-native (Docker, Kubernetes).**
* **Zabbix thích hợp cho hệ thống mạng truyền thống (LAN, server vật lý).**

**Khả năng mở rộng:**

* **Prometheus dễ mở rộng theo chiều ngang (scale-out).**
* **Zabbix mở rộng nhưng khó hơn, phụ thuộc nhiều vào cấu hình server.**

### ****So sánh với Nagios****

Nagios là một trong những công cụ giám sát lâu đời, tập trung vào kiểm tra trạng thái dịch vụ (Service Monitoring). Nagios nổi tiếng với khả năng phát hiện sự cố nhưng hạn chế trong trực quan hóa.



Hình 2.4. So sánh giữa Nagios và Prometheus

**Công nghệ cốt lõi:**

* Nagios chủ yếu dùng phương pháp kiểm tra “ping” hoặc “service check”.
* Prometheus giám sát theo thời gian thực với dữ liệu chuỗi thời gian.

**Khả năng mở rộng:**

* Nagios gặp khó khăn khi giám sát hàng ngàn node.
* Prometheus hỗ trợ scale dễ dàng và hoạt động tốt với microservices.

**Trực quan hóa:**

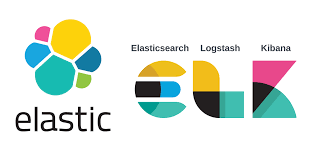
* Nagios có giao diện cơ bản, muốn nâng cấp phải tích hợp thêm plugin (NagVis).
* Grafana cung cấp trực quan hóa mạnh mẽ và realtime.

**Cảnh báo:**

* Nagios có cơ chế cảnh báo tích hợp nhưng khá thủ công.
* Prometheus kết hợp Alertmanager mạnh mẽ, dễ tích hợp hệ thống chat, email.

#### So sánh với ELK Stack (Elasticsearch – Logstash – Kibana)

ELK Stack tập trung vào phân tích log (logging), tìm kiếm dữ liệu lớn và trực quan hóa log với Kibana. ELK không phải công cụ giám sát thuần túy nhưng thường được dùng kết hợp để phân tích sự cố.



Hình 2.5. ELK Stack (Elasticsearch – Logstash – Kibana)

Chức năng chính:

* ELK thiên về lưu trữ, tìm kiếm và phân tích log.
* Prometheus chuyên về giám sát chỉ số hiệu năng (metrics).

Dữ liệu xử lý:

* ELK xử lý dữ liệu phi cấu trúc (log text).
* Prometheus xử lý dữ liệu chuỗi thời gian (time-series).

Trực quan hóa:

* Kibana mạnh trong phân tích log.
* Grafana mạnh trong giám sát số liệu thời gian thực.

Tích hợp:

* ELK yêu cầu nhiều tài nguyên và phức tạp hơn khi triển khai.
* Prometheus – Grafana nhẹ, dễ triển khai hơn cho hệ thống giám sát thời gian thực.

## ****Ưu điểm và nhược điểm của Prometheus – Grafana****

### ****Ưu điểm:****

Mã nguồn mở, miễn phí và cộng đồng hỗ trợ lớn.

Khả năng mở rộng cao, phù hợp hệ thống từ nhỏ đến lớn.

Hỗ trợ trực quan hóa mạnh mẽ, dashboard tùy chỉnh dễ dàng.

Thu thập dữ liệu thời gian thực với độ trễ thấp.

### ****Nhược điểm:****

Không lưu trữ dữ liệu dài hạn mặc định (cần tích hợp thêm như Thanos hoặc Cortex).

Cấu hình cảnh báo phức tạp hơn Zabbix hoặc Nagios.

Yêu cầu kiến thức chuyên sâu về PromQL và cấu trúc hệ thống để khai thác tối đa.

## ****Ứng dụng trong doanh nghiệp****

Prometheus và Grafana có thể được triển khai trong nhiều tình huống:

**Giám sát máy chủ:** CPU, RAM, Disk, I/O, Network,…

**Giám sát dịch vụ mạng:** HTTP, DNS, Database, API,…

**Giám sát container & microservices:** Kubernetes, Docker Swarm,…

**Giám sát bảo mật:** Phát hiện bất thường về lưu lượng, phát hiện sự cố dịch vụ.

**Cảnh báo thời gian thực:** Tích hợp email, Slack, Telegram để thông báo cho quản trị viên.

## Cấu hình và cài đặt Prometheus và Grafana

### ****Phần cứng tối thiểu****

**CPU:** 2 core trở lên

**RAM:** 4 GB trở lên

**Ổ cứng:** 20 GB (ưu tiên SSD để tối ưu hiệu năng)

**Mạng:** Kết nối Internet ổn định

### ****Phần mềm****

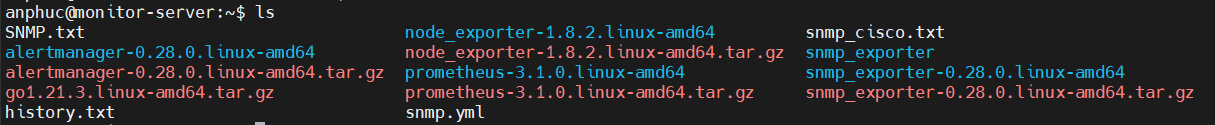
**Hệ điều hành:** Ubuntu Server 22.04 LTS hoặc CentOS 7/8

**Quyền truy cập:** Quyền sudo hoặc root

**Công cụ cần thiết:** wget, tar, systemctl, firewalld/ufw

### Cài đặt Prometheus

Bước 1: Tải Prometheus và các gói cần thiết



Hình 2.6. Tải Prometheus

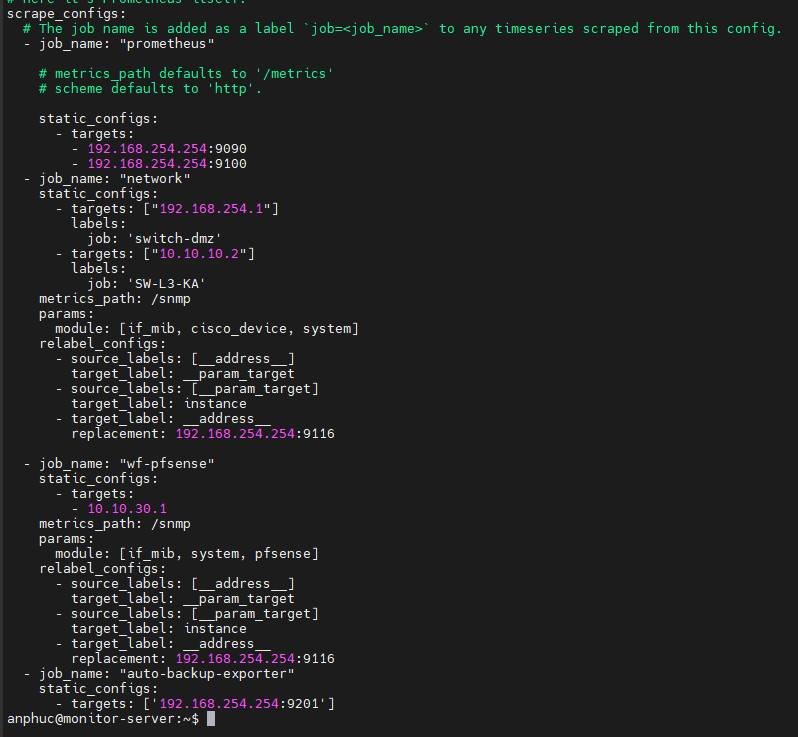
* cd /tmp
* wget https://github.com/prometheus/prometheus/releases/download/v2.53.0/prometheus-2.53.0.linux-amd64.tar.gz
* tar xvf prometheus-2.53.0.linux-amd64.tar.gz
* sudo mv prometheus-2.53.0.linux-amd64 /usr/local/Prometheus

Bước 2: Tạo user Prometheus

* sudo useradd --no-create-home --shell /bin/false prometheus
* sudo mkdir /etc/prometheus
* sudo mkdir /var/lib/prometheus
* sudo chown prometheus:prometheus /etc/prometheus /var/lib/Prometheus

Bước 3: Cấu hình Prometheus

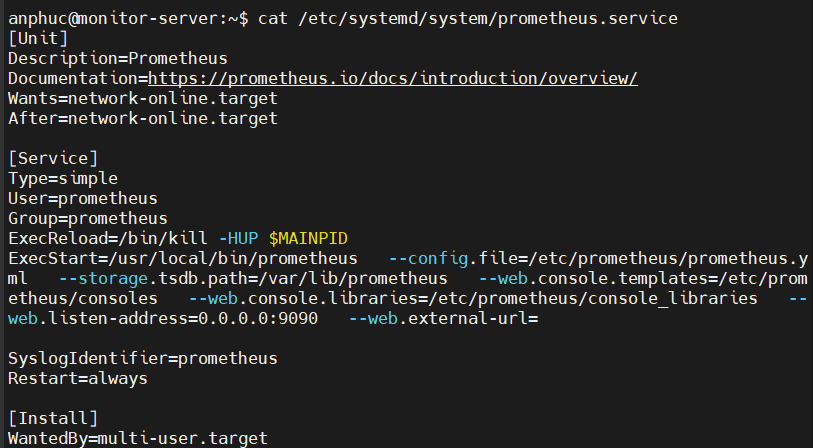
Tạo file /etc/prometheus/prometheus.yml:



Hình 2.7. Tạo file cấu hình

Bước 4: Tạo service Prometheus

Tạo file /etc/systemd/system/prometheus.service:

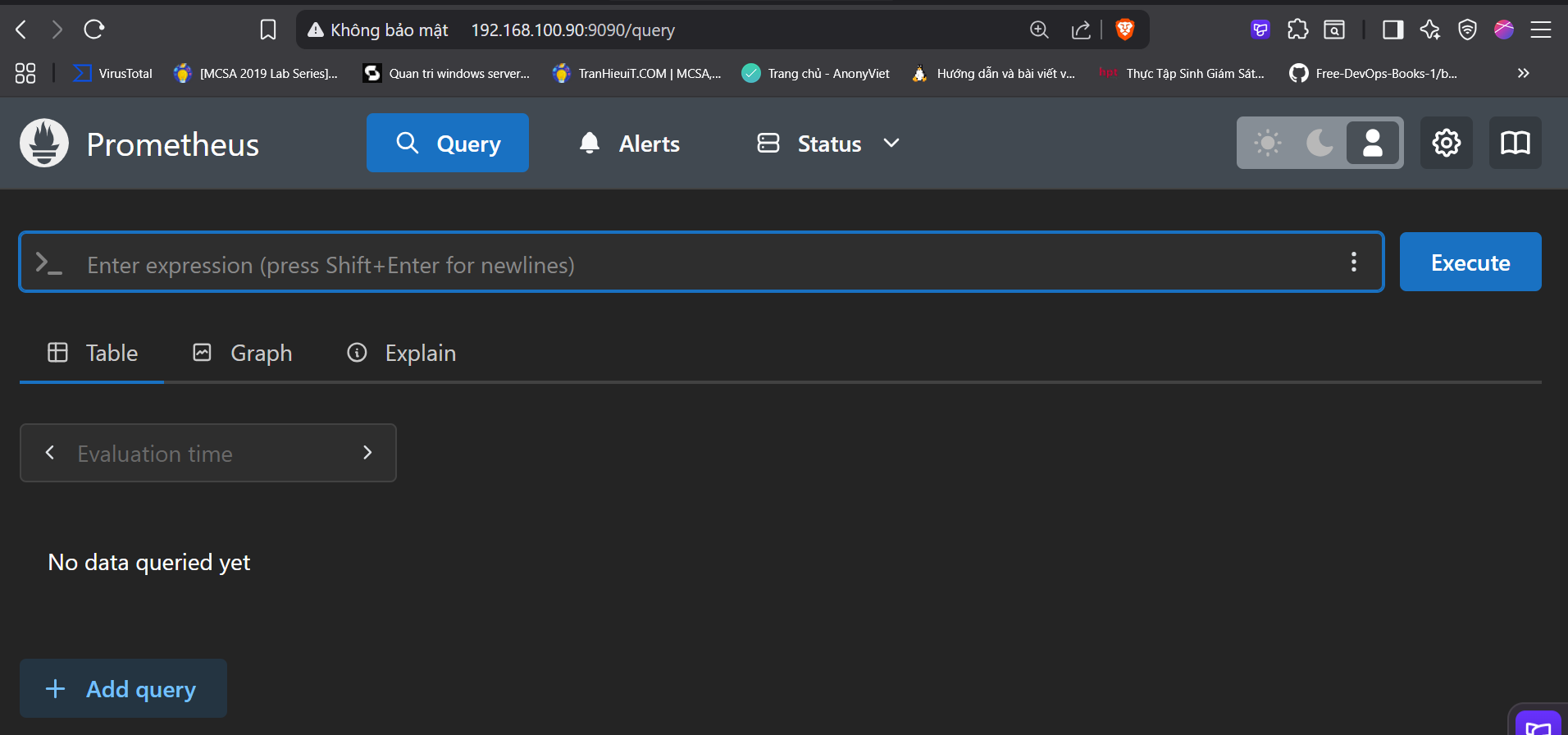


Hình 2.8. Tạo file service cho Prometheus

Và khởi động dịch vụ

* sudo systemctl daemon-reexec
* sudo systemctl enable --now Prometheus

Sau đó truy cập với địa chỉ ip của máy trên cổng 9090 để kiểm tra



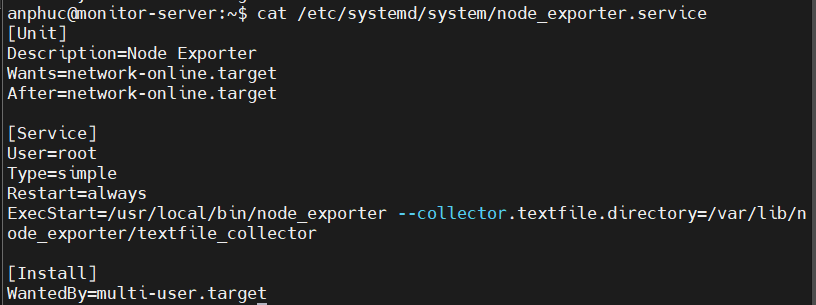
Hình 2.9. Truy cập giao diện web

### Cài đặt Node Exporter (giám sát hệ thống)

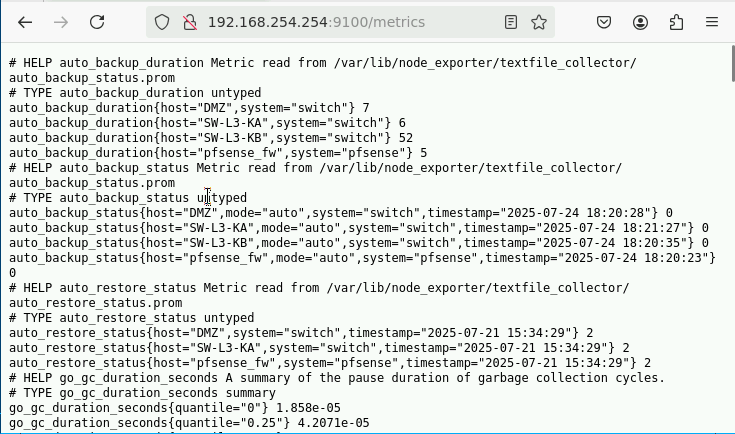
Node Exporter thu thập số liệu CPU, RAM, disk, network.

* cd /tmp
* wget https://github.com/prometheus/node\_exporter/releases/download/v1.7.0/node\_exporter-1.7.0.linux-amd64.tar.gz
* tar xvf node\_exporter-1.7.0.linux-amd64.tar.gz
* sudo mv node\_exporter-1.7.0.linux-amd64/node\_exporter /usr/local/bin/
* sudo useradd --no-create-home --shell /bin/false node\_exporter
* sudo chown node\_exporter:node\_exporter /usr/local/bin/node\_exporter

Sau đó tạo service để khởi chạy và kiểm tra metrics



Hình 2.10. Tạo service Node exporter



Hình 2.11. Kiểm tra thu thập metrics

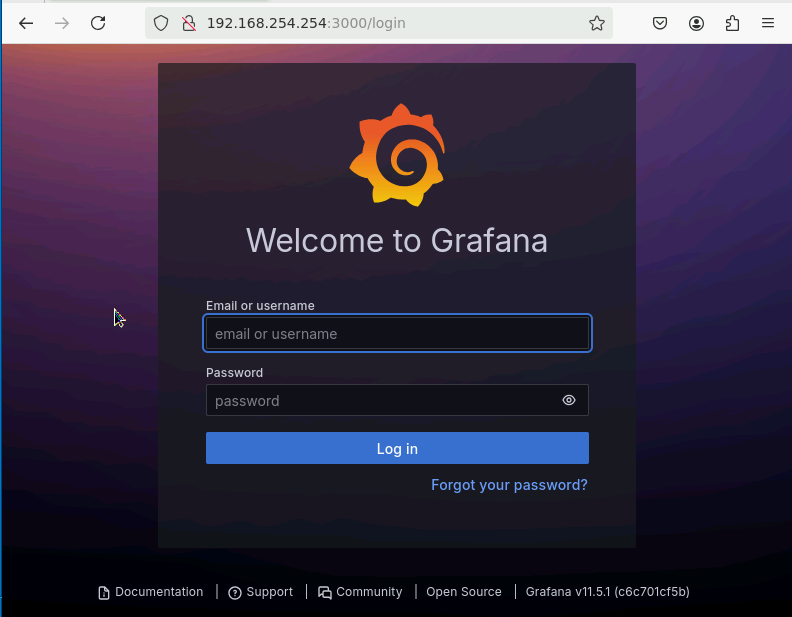
### Cài đặt Grafana

Bước 1: Thêm repository Grafana

* sudo apt-get install -y software-properties-common
* sudo add-apt-repository "deb https://packages.grafana.com/oss/deb stable main"
* wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -
* sudo apt-get update

Bước 2: Cài đặt Grafana

* sudo apt-get install grafana -y
* sudo systemctl enable --now grafana-server



Hình 2.12. Giao diện Grafana

Truy cập **http://<IP-server>:3000 để kiểm tra giao diện truy cập với mật khẩu mặc định là admin/admin**

## Kết chương 2

Trong chương 2, báo cáo đã trình bày và phân tích các công nghệ cốt lõi tạo nên hệ thống giám sát và cảnh báo an toàn mạng sử dụng Prometheus và Grafana. Nội dung bao gồm kiến trúc tổng thể, cơ chế hoạt động, các thành phần chính như Prometheus Server, Exporter, Alertmanager, và Grafana, cũng như quy trình thu thập – lưu trữ – trực quan hóa dữ liệu.

Bên cạnh đó, chương này cũng đã so sánh Prometheus – Grafana với các giải pháp giám sát phổ biến khác như Zabbix, Nagios, ELK Stack, qua đó làm rõ ưu điểm về khả năng mở rộng, tích hợp linh hoạt và phù hợp cho môi trường cloud-native. Đồng thời, phần hướng dẫn cài đặt và cấu hình cơ bản Prometheus – Grafana cung cấp nền tảng thực hành cho chương triển khai lab.

Những kiến thức trong chương này là cơ sở quan trọng để tiến hành triển khai hệ thống giám sát mạng thực tế, cấu hình cảnh báo và đánh giá hiệu quả trong chương tiếp theo. Qua đó, người đọc có thể hiểu rõ cách áp dụng Prometheus và Grafana vào thực tiễn doanh nghiệp để nâng cao khả năng bảo vệ và giám sát an toàn hệ thống mạng.

# ****XÂY DỰNG LAB TRIỂN KHAI****

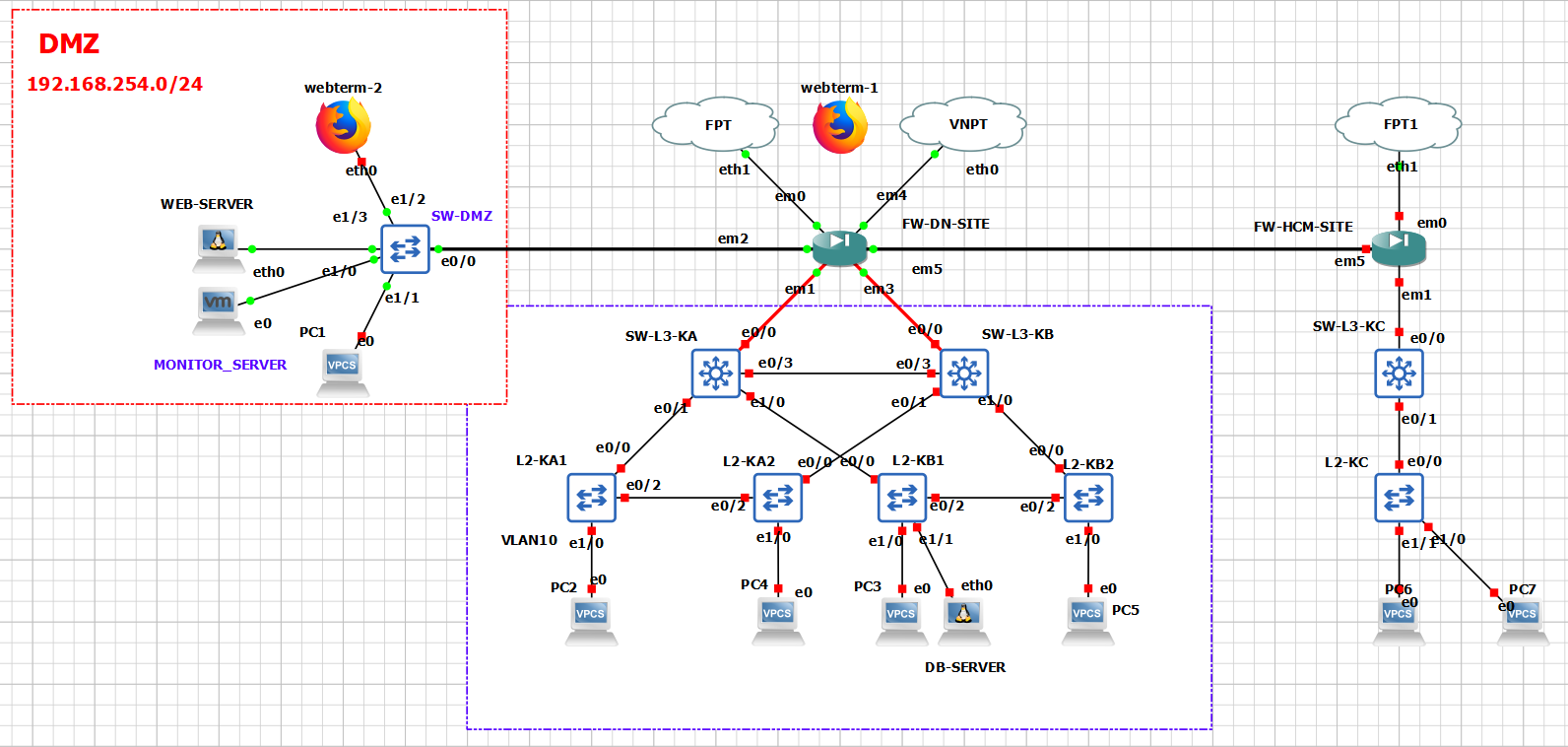
## ****Thiết kế mô hình phòng lab****

### Mục tiêu triển khai

Mục tiêu chính của chương này là xây dựng một môi trường thí nghiệm (lab) sử dụng Prometheus và Grafana để mô phỏng việc giám sát và bảo vệ hệ thống mạng. Các mục tiêu cụ thể bao gồm:

* Cài đặt và cấu hình **Node Exporter** để giám sát hiệu suất hệ thống (CPU, RAM, Disk, Network).
* Cài đặt và tích hợp **Alertmanager và Telegram** để gửi cảnh báo khi có sự cố.
* Sử dụng **Grafana** để trực quan hóa dữ liệu và tạo dashboard giám sát theo thời gian thực.
* Kiểm thử khả năng cảnh báo khi vượt ngưỡng tài nguyên hoặc khi dịch vụ gặp sự cố.

### Mô hình hệ thống thực nghiệm



Hình 3.1. Sơ đồ mô hình hệ thống

DB Server & Client: Được phân bố trong các VLAN khác nhau.

Switch DMZ: Quản lý kết nối đến vùng DMZ, nơi chứa các dịch vụ công khai và giám sát.

Monitor Server (192.168.254.254/24): Đóng vai trò theo dõi hoạt động mạng và tài nguyên hệ thống, được đặt trong vùng DMZ.

Web server chỉ phục vụ nội bộ – người dùng trong VLAN 10–40 có thể truy cập

### Phân vùng mạng chi tiết:

#### Vùng LAN (quản lý qua switch L3)

|  |  |  |
| --- | --- | --- |
| SWITCH | Địa chỉ IP | Mô tả |
| SW-L3-KA  SW-L3-KB | 10.10.20.1 | Switch core layer 3 khu A - B |
| SW-L2-KA1  SW-L2-KA2 | 192.168.10.2  192.168.20.2 | Switch layer 2 khu A |
| SW-L2-KB1  SW-L2-KB2 | 192.168.30.2  192.168.40.2 | Switch layer 2 khu B |

#### Vùng DMZ

|  |  |  |
| --- | --- | --- |
| Thành phần | Địa chỉ IP | Mô tả |
| Switch DMZ | 192.168.254.1 | Quản lý thiết bị vùng DMZ |
| Monitor Server | 192.168.254.254 | Giám sát hệ thống |
| Web Server | 192.168.254.100 | Web nội bộ phục vụ người dùng |

## Cấu hình triển khai mô phỏng

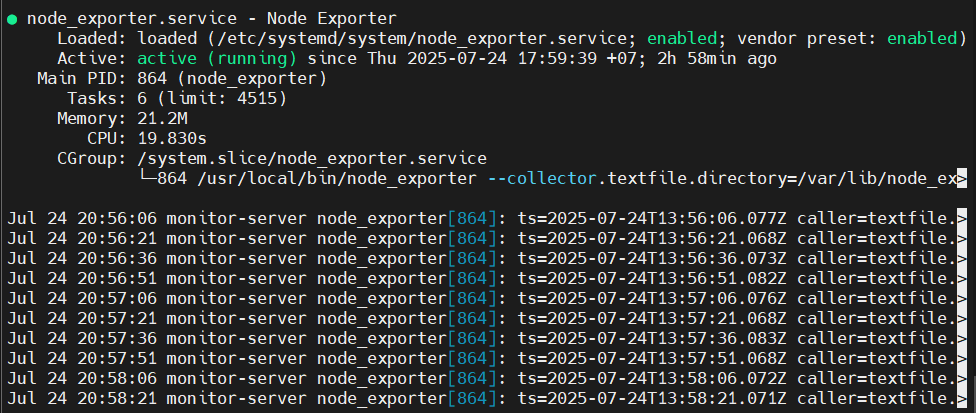
### Cấu hình Prometheus thu thập metric

#### Thu thập sử dụng node\_exporter

Cài đặt node\_exporter trên từng thiết bị cần được giám sát

* cd /tmp
* wget https://github.com/prometheus/node\_exporter/releases/download/v1.7.0/node\_exporter-1.7.0.linux-amd64.tar.gz
* tar xvf node\_exporter-1.7.0.linux-amd64.tar.gz
* sudo mv node\_exporter-1.7.0.linux-amd64/node\_exporter /usr/local/bin/
* sudo useradd --no-create-home --shell /bin/false node\_exporter
* sudo chown node\_exporter:node\_exporter /usr/local/bin/node\_exporter

Tạo file dịch vụ và kiểm tra trạng thái



Hình 3.2. Kiểm tra trạng thái hoạt động của node\_exporter

#### Thu thập sử dụng giao thức SNMP

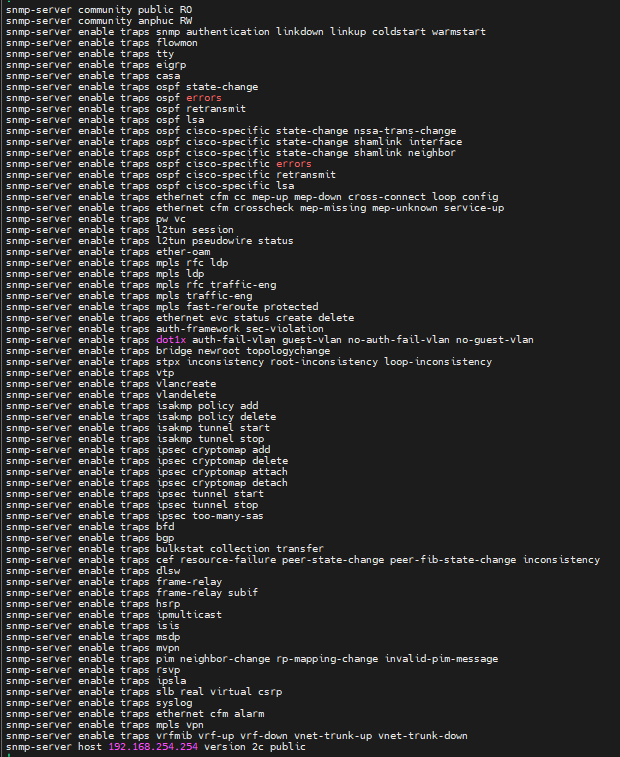
Xác định và cấu hình các thiết bị cần được cấu hình SNMP, như switch, firewall

1. Trên switch cisco

Cấu hình SNMP sử dụng các câu lệnh:

* snmp-server community public RO
* snmp-server community anphuc RW
* snmp-server host 192.168.254.254 public
* snmp-server enable traps

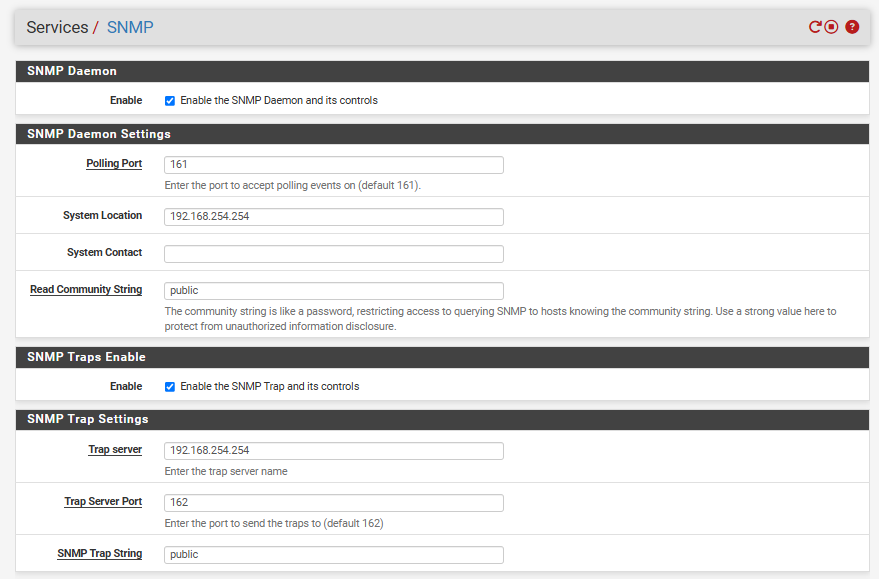
Kiểm tra thông tin:



Hình 3.3. Kiểm tra cấu hình

1. Trên firewall pfsense

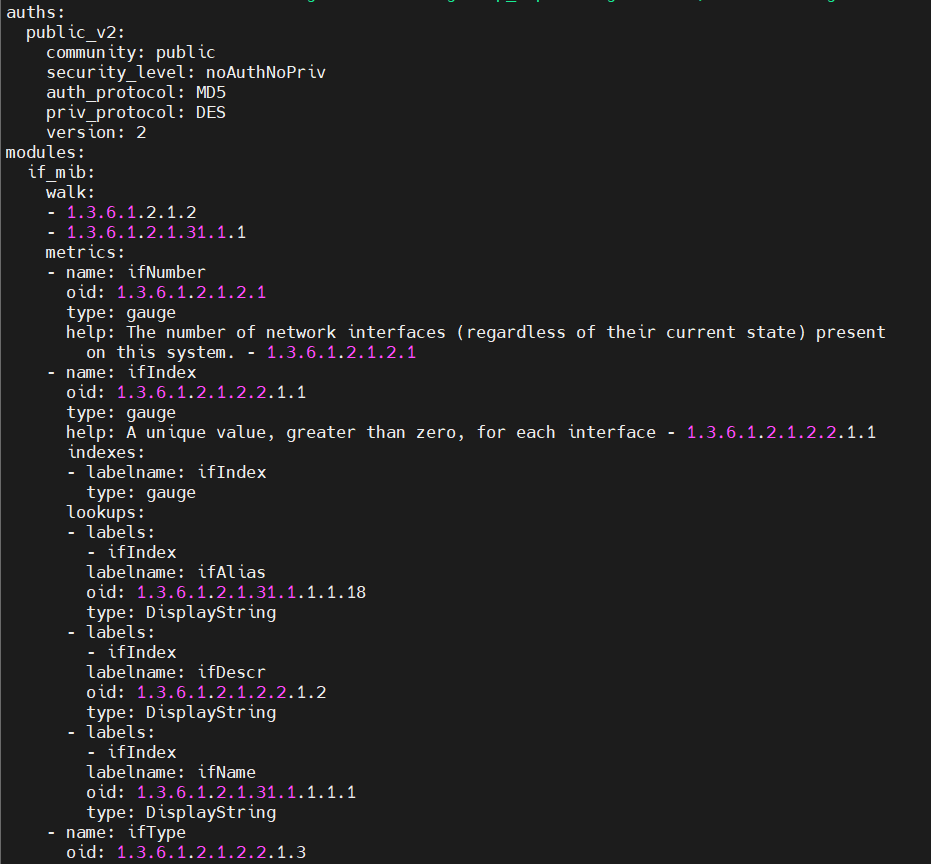
Vào Services -> SNMP



Hình 3.4. Cấu hình SNMP trên Pfsense

Bật trạng thái SNMP, sử dụng port 161 mặc định, snmp server 192.168.254.254 và community string là public

Sau khi đã cấu hình snmp xong thì ta cần tạo file /etc/prometheus/snmp.yml để lưu các metric thành 1 module để dễ dàng cấu hình



Hình 3.5. Gom các metrics thành các module phù hợp

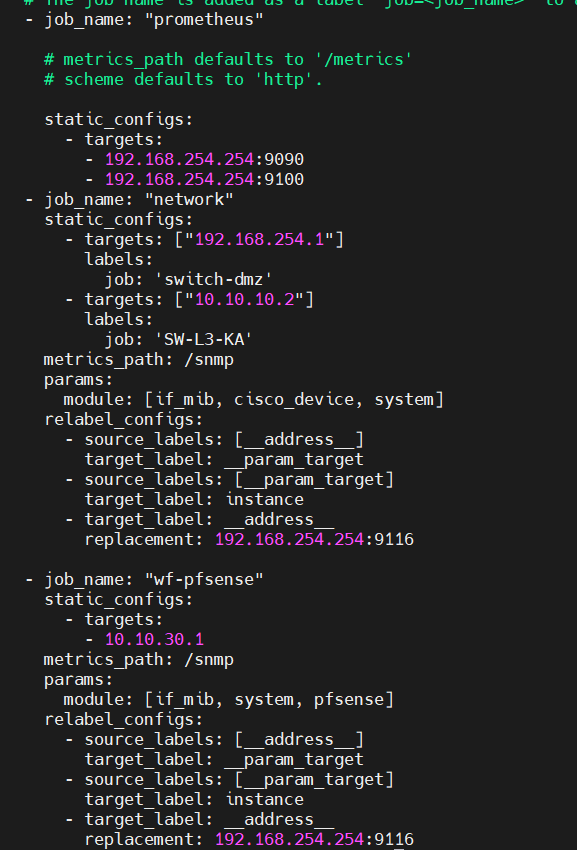
#### Cấu hình file Prometheus.yml

Sau khi ta đã xác định và thu thập được metrics ở các thiết bị ta bắt đầu tạo file /etc/prometheus/prometheus.yml để Prometheus xác định được chính xác các đối tượng cần thu thập thông tin và sử dụng modules phù hợp

Xác định các targets

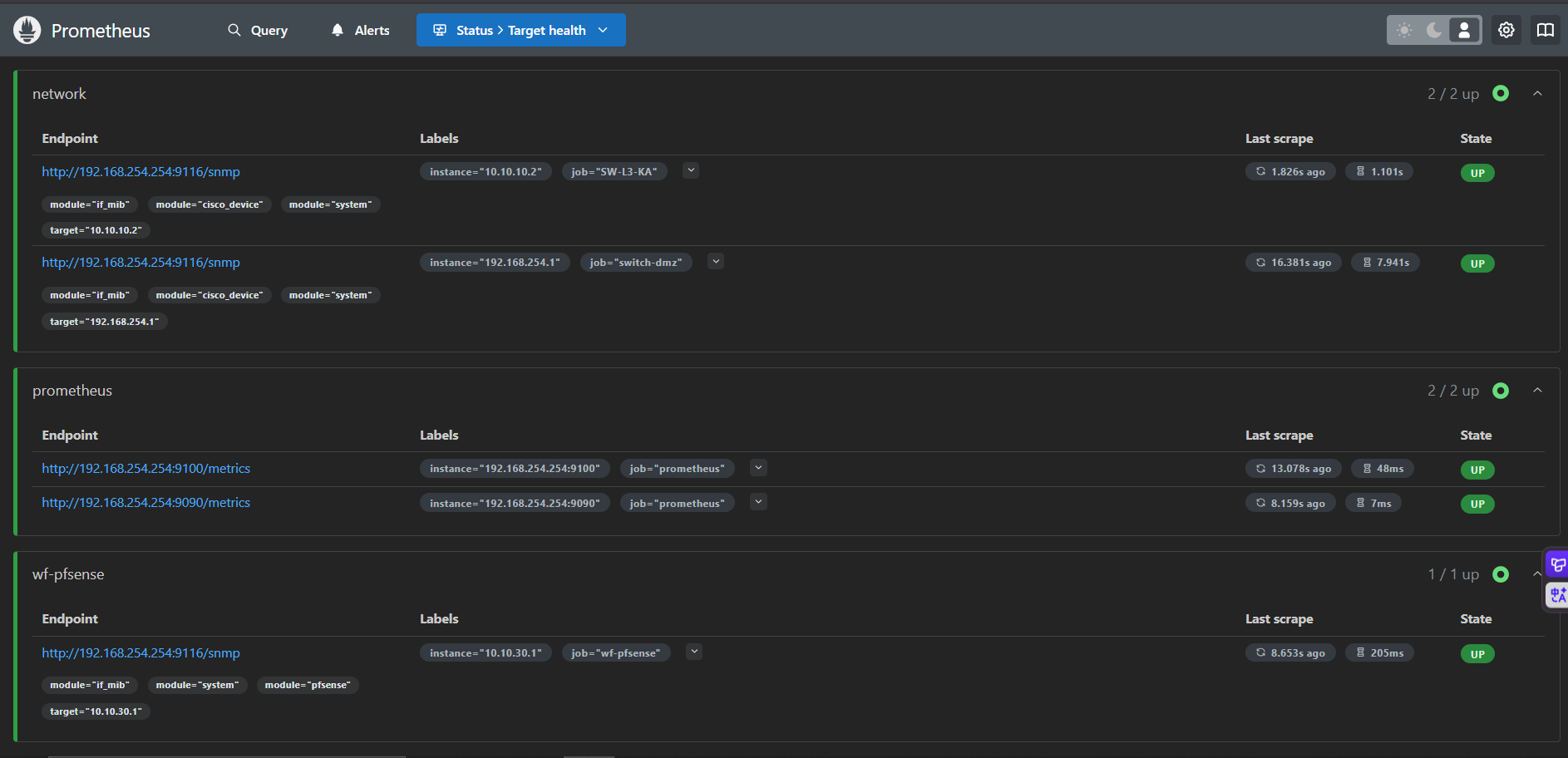
Quy định job name cho từng đối tượng

Đường dẫn đến file snmp và sử dụng các module nào



Hình 3.6. Cấu hình file Prometheus.yml

Sau khi đã cấu hình xong thì ta cần quay lại giao diện web Prometheus để kiểm tra trạng thái các node.



Hình 3.7. Kiểm tra trạng thái hoạt động của từng thiết bị

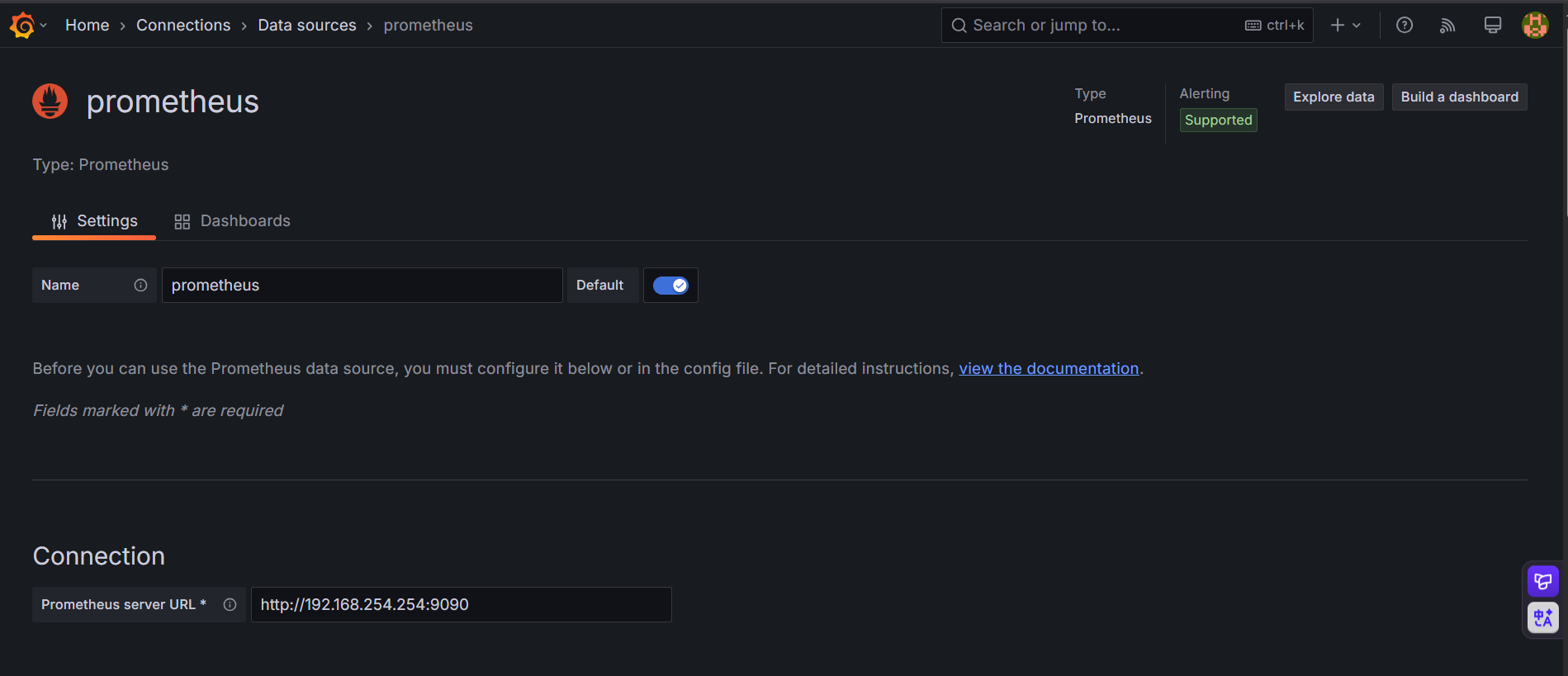
### Tích hợp hệ thống và tạo Dashboard

#### **Thêm Prometheus vào Grafana**

Vào **Configuration → Data Sources → Add Data Source**

Chọn **Prometheus →** nhập URL: http://192.168.254.254:9090

Save & Test.



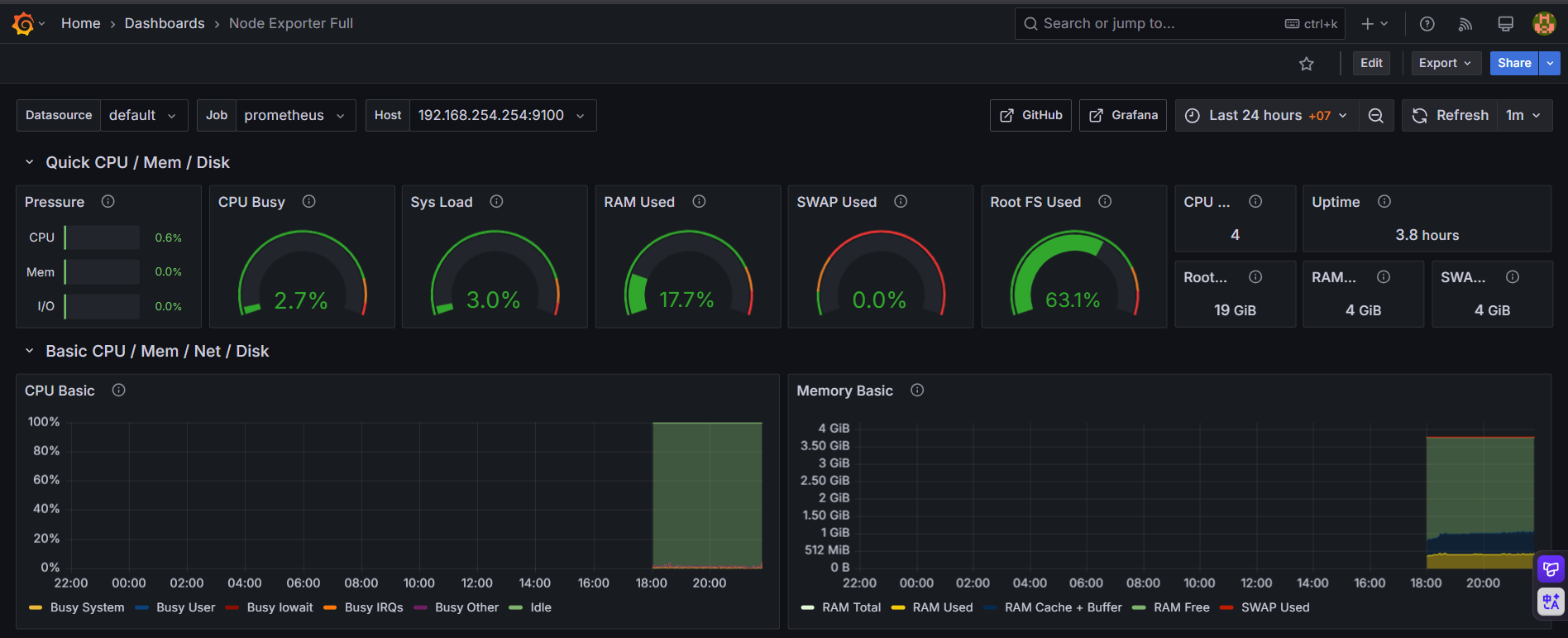
Hình 3.8. Tích hợp Prometheus vào Grafana

#### **Tạo Dashboard giám sát sử dung node\_exporter**

Import dashboard sẵn có từ **Grafana Labs (ID: 1860 – Node Exporter Full)**

Hoặc tự tạo panel hiển thị:

* **CPU Usage**
* **Memory Usage**
* **Disk I/O**
* **Network Traffic**

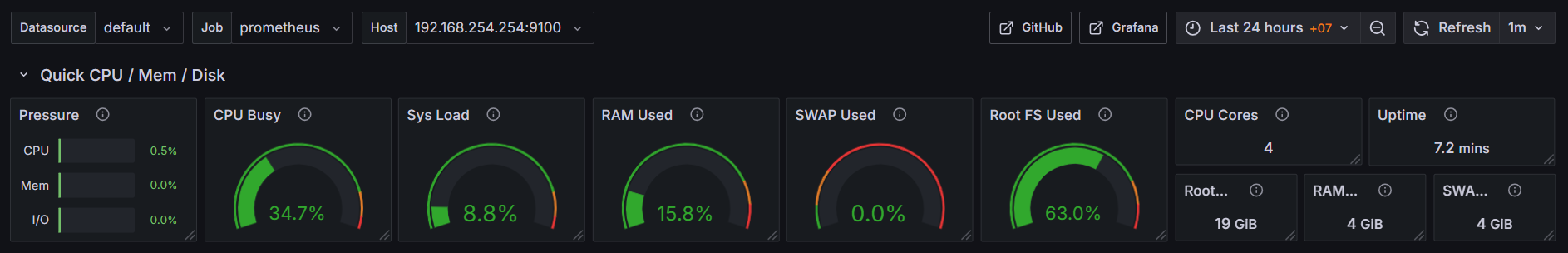


Hình 3.9. Import Dashboard có sẵn từ Grafana Lab

Qua dashboard trên ta thu thập được các thông tin như:

1. **Thông tin chung về hệ thống**

* Tên máy chủ, địa chỉ IP, uptime (thời gian hoạt động).
* Phiên bản kernel, phiên bản Node Exporter.
* Số CPU, tổng dung lượng RAM, dung lượng disk.



Hình 3.10. Thông tin chung về hệ thống

1. Hiệu suất CPU

* Tổng quan CPU usage (%): idle, user, system, iowait, steal…
* Tải trung bình (Load Average): 1 phút, 5 phút, 15 phút.
* Usage từng core CPU (nếu server đa nhân).



Hình 3.11. Hiệu suất

1. Bộ nhớ (Memory)

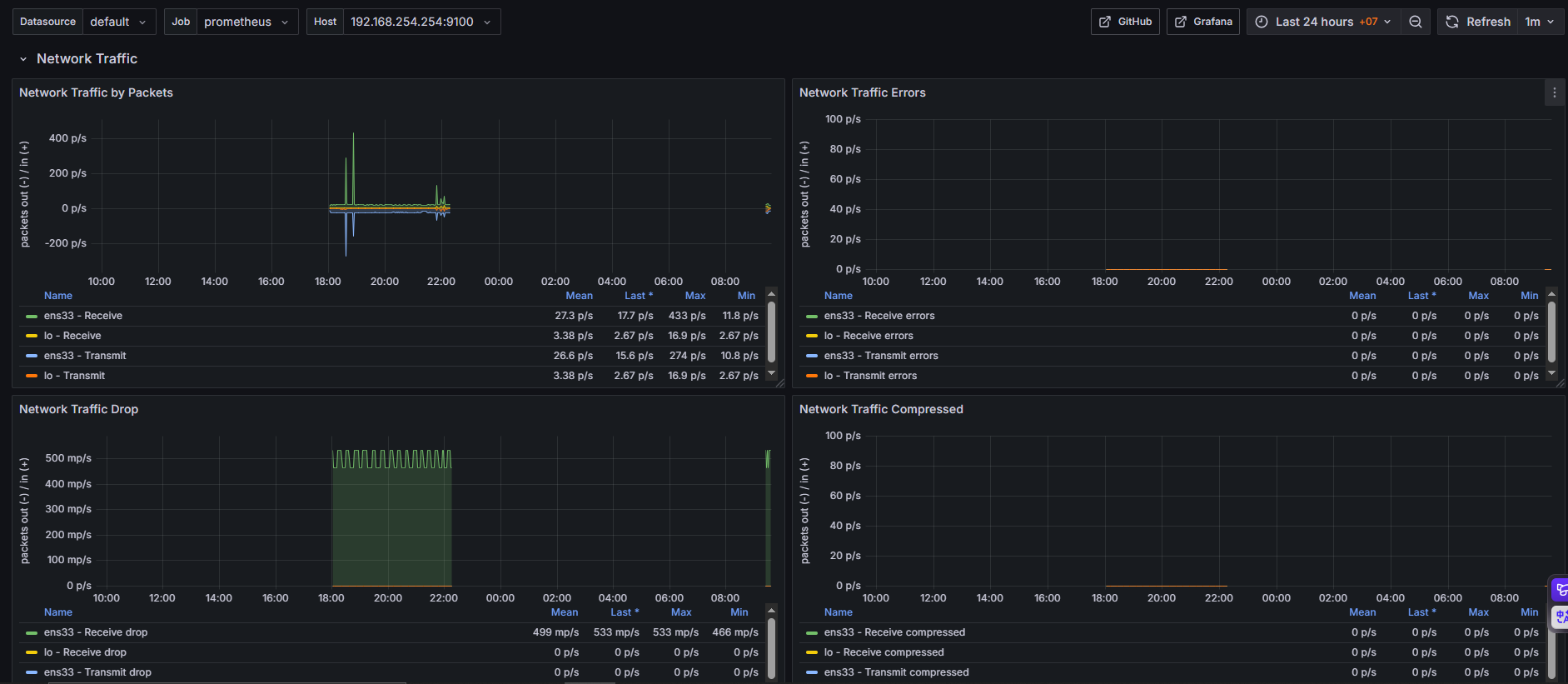
* RAM tổng, đã sử dụng, còn trống, buffers/cache.
* Swap memory: sử dụng bao nhiêu, còn trống bao nhiêu.
* Tỷ lệ phần trăm sử dụng RAM theo thời gian.

1. Disk (Ổ đĩa)

* Disk usage: dung lượng đã dùng / tổng dung lượng từng partition.
* Disk I/O: số lượng đọc/ghi, tốc độ đọc/ghi (MB/s), độ trễ.
* Inode usage: dung lượng inode còn trống / đã dùng.

1. Network (Mạng)

* Lưu lượng mạng In/Out theo interface (eth0, ens33, v.v…).
* Tốc độ (bytes per second hoặc Mbps).
* Tổng bytes đã truyền và nhận.
* Gói tin bị lỗi, dropped packets.



Hình 3.12. Network traffic trên từng cổng

1. File System & Mount Points

* Danh sách tất cả filesystem được mount.
* Tỷ lệ sử dụng từng phân vùng.

1. Process

* Số lượng tiến trình đang chạy, sleep, zombie.
* Tiến trình chiếm nhiều CPU/RAM nhất (có thể custom thêm).

1. Thông tin khác

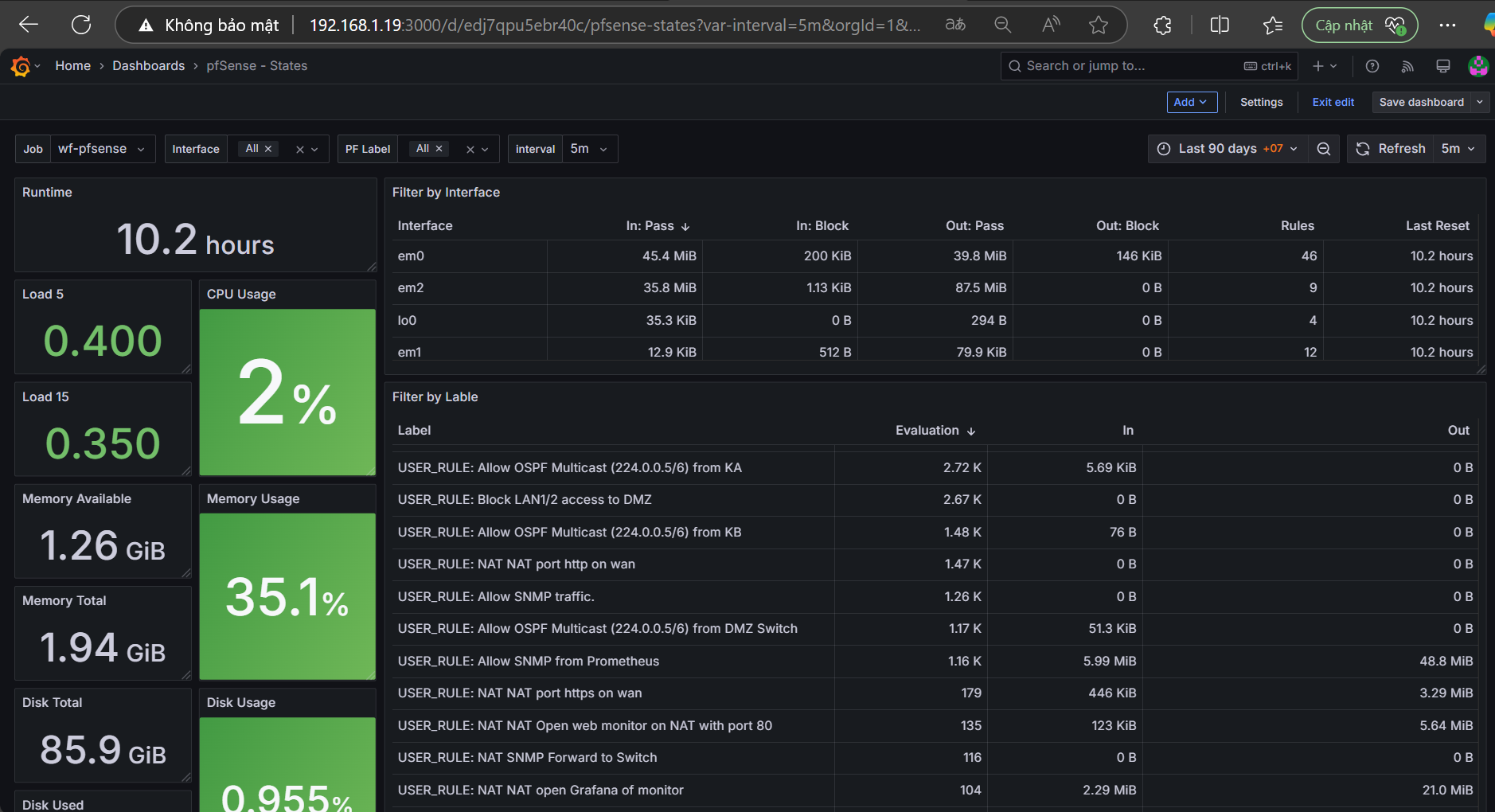
* Temperature (nếu server hỗ trợ sensor).
* Context switches, interrupts.
* Boot time (thời gian khởi động gần nhất).

#### Tạo dashboard giám sát sử dụng SNMP walk

Kích hoạt dịch vụ SNMP để cho phép phần mềm giám sát thu thập dữ liệu trạng thái hệ thống.

SNMP Daemon hoạt động với community string là public.

Dữ liệu như CPU, RAM, traffic interface được thu thập qua SNMP Walk/Prometheus

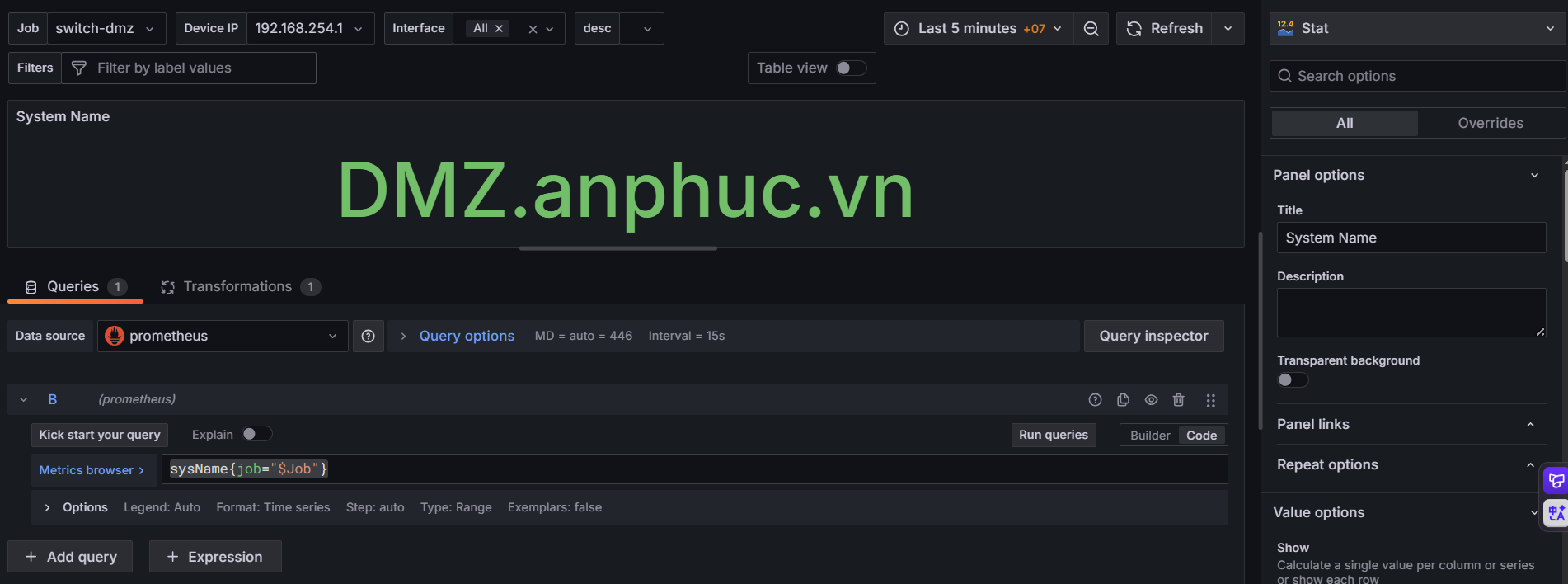


Hình 3.13: Xem giám sát thông qua Grafana

#### Tự tạo dashboard để giám sát các thiết bị switch

1. Cấu hình tên thiết bị

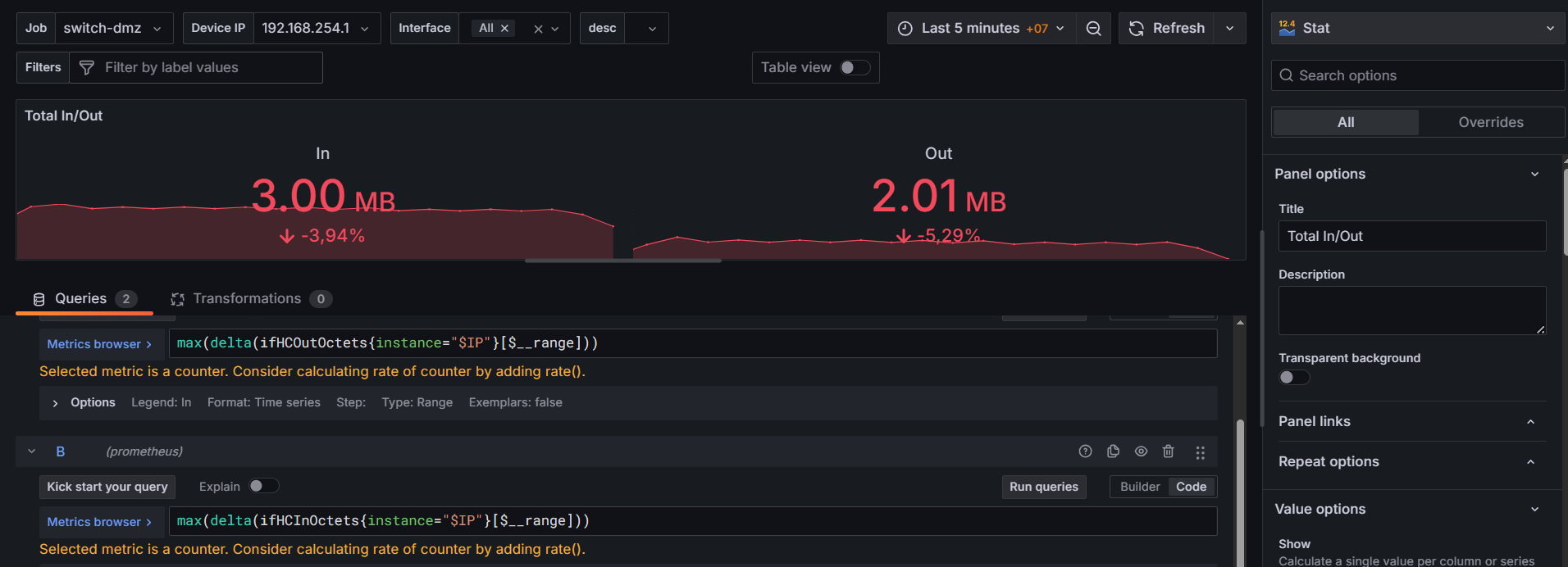
Vào new dashboard -> add visualization -> datasource: Prometheus



Hình 3.14. Giao diện tùy chỉnh

Vào Metric browsers: thêm queries “sysName{job="$Job"}” để lấy được hostname của thiết bị. Vào transformation thêm “Lables to fields” để chỉ hiển thị sysName, và sử dụng biểu đồ Stat để hiển thị

1. Cấu hình hiển thị tổng lưu lượng mạng In/Out



Hình 3.15. Tổng lưu lượng mạng In/Out

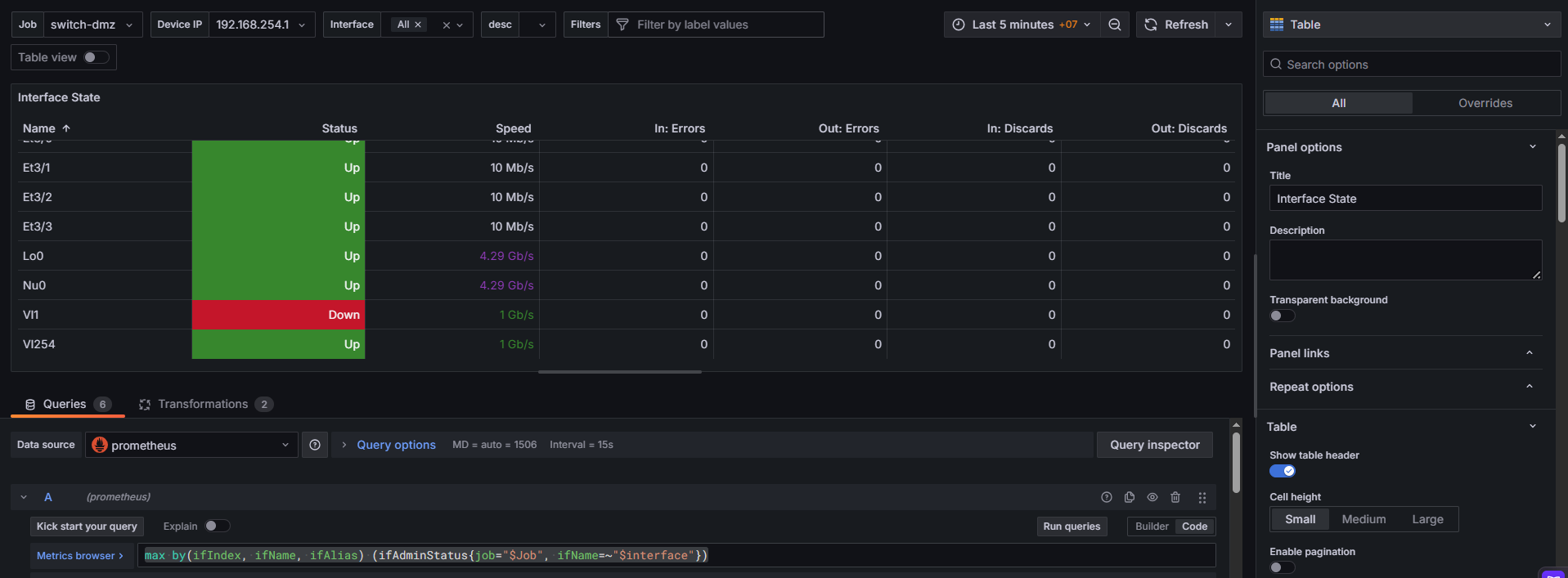
Sửu dụng các truy vấn

* ifHCInOctets và ifHCOutOctets là metric SNMP thể hiện tổng số byte nhận/gửi của interface.

Biểu đồ này cho biết lượng dữ liệu tổng cộng đã vào/ra thiết bị trong một khoảng thời gian nhất định.

* Dùng để kiểm tra nhanh tình trạng tải mạng của switch/router DMZ.
* Có thể dùng làm cơ sở đặt cảnh báo (ví dụ: In/Out vượt quá 100MB trong 5 phút → cảnh báo).

1. **Bảng trạng thái (Interface State):**

****

Hình 3.16. Bảng trạng thái từng cổng

Sử dụng query:

ifAdminStatus: Đây là metric thu thập từ SNMP Exporter, thể hiện trạng thái hành chính (administrative status) của các interface (cổng mạng) trên thiết bị (switch/router).

Giá trị thường là:

1: Up (được bật)

2: Down (được tắt)

3: Testing

max by(ifIndex, ifName, ifAlias)

* ifIndex: Chỉ số duy nhất cho mỗi interface.
* ifName: Tên interface (VD: Et3/0, Vi1).
* ifAlias: Mô tả/ghi chú interface nếu được cấu hình trong thiết bị.
* max by(...): Gom nhóm các metric theo các nhãn này và lấy giá trị cao nhất (thường là 1 hoặc 2).
* Mục đích: Hiển thị trạng thái hiện tại của từng interface.

Biến $Job và $interface

* $Job: Lọc theo tên job đã cấu hình trong Prometheus (ví dụ switch-dmz).
* $interface: Lọc theo tên interface cụ thể hoặc tất cả interface (All).

### Cấu hình tích hợp alertmanager gửi cảnh báo

Hệ thống Prometheus kết hợp Alertmanager được triển khai nhằm:

* Giám sát tình trạng mạng và tài nguyên hệ thống.
* Phát hiện sự cố tức thời (Interface Down/Backup Fail).
* Theo dõi xu hướng tăng tải (Traffic/Memory).
* Gửi cảnh báo đa kênh (Telegram + Email) giúp phản ứng nhanh.

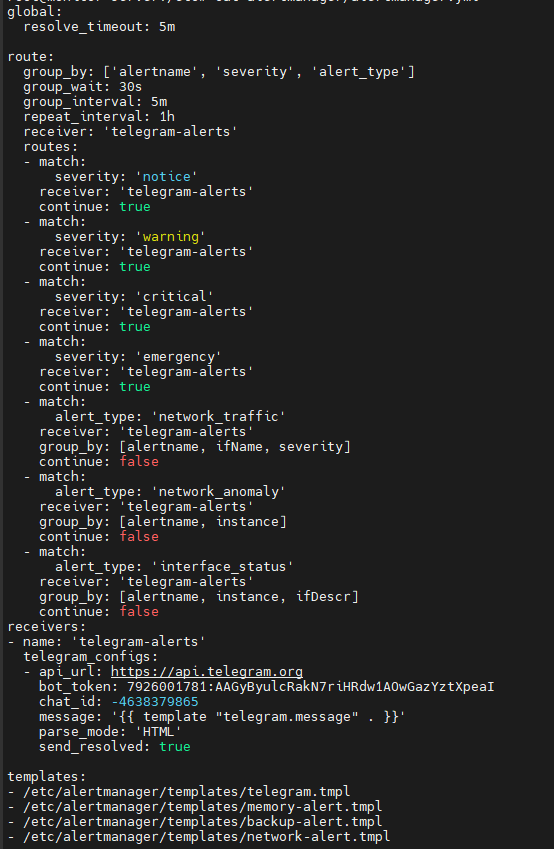


Hình 3.17. Hệ thống gửi cảnh báo Alertmanager

#### Kết nối cấu hình cảnh báo đến telegram

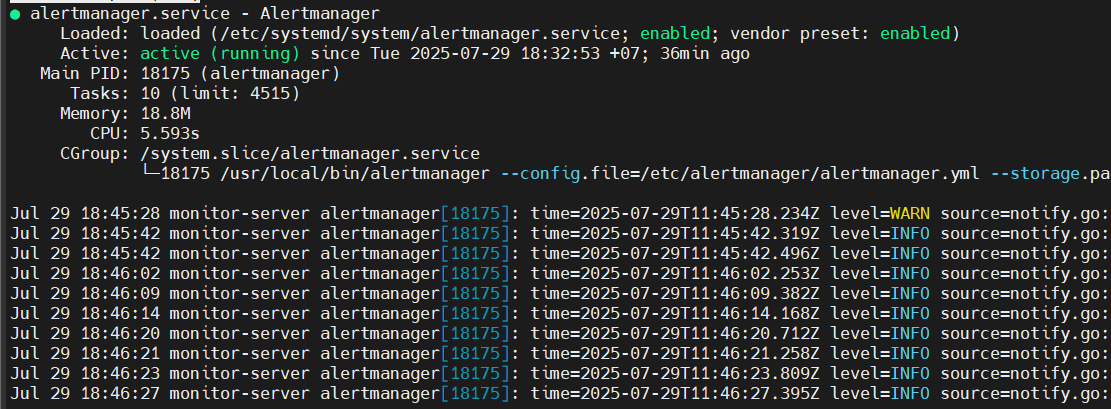
Thực hiện kết nối đến alertmanager bằng cách tạo file alertmanager.yml trong /etc/alertmanager/

Cấu hình token-id và chat-id để kết nối được đến với telegram



Hình 3.18. Cấu hình alertmanager.yml

Thực hiện restart và kiểm tra trạng thái hoạt động của alertmanager



Hình 3.19. Trạng thái dịch vụ Alertmanager

#### Quy tắc cảnh báo (Alert Rules)

1. Cảnh báo trạng thái cổng mạng (Interface)

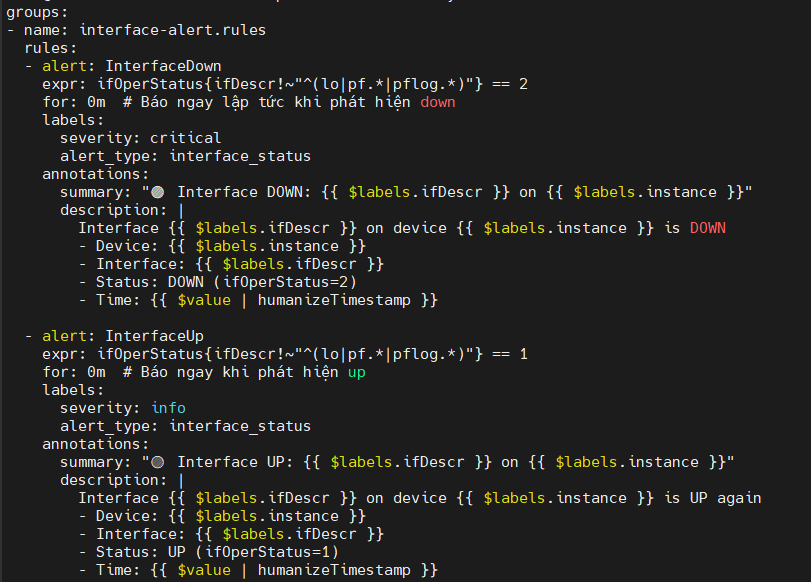
Mục tiêu: Phát hiện nhanh khi cổng mạng bị down hoặc up trở lại.

Rule:

* InterfaceDown: Nếu ifOperStatus == 2 → severity critical.
* InterfaceUp: Nếu ifOperStatus == 1 → severity info.

Tính năng:

* Bỏ qua loopback và interface ảo (lo, pf.\*, pflog.\*).
* Gửi ngay lập tức (for: 0m).

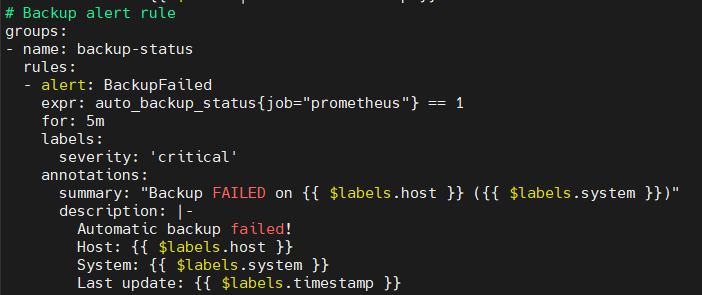


Hình 3.20. Rule cảnh báo trạng thái cổng mạng

1. Cảnh báo lỗi Backup

Mục tiêu: Đảm bảo sao lưu tự động không bị gián đoạn.

* Rule: auto\_backup\_status == 1 trong 5 phút → severity critical.
* Thông tin bổ sung: Hiển thị tên host, loại thiết bị, timestamp.



Hình 3.21. Rule cảnh báo trạng backup khi không backup thành công

1. Cảnh báo Traffic mạng

Mục tiêu: Giám sát lưu lượng In/Out bất thường hoặc vượt ngưỡng.

Rule:

* HighInboundTraffic > 100 Mbps (warning) hoặc > 150 Mbps (critical).
* HighOutboundTraffic > 50 Mbps (warning) hoặc > 75 Mbps (critical).
* NetworkTrafficAnomaly: Lưu lượng bất thường so với trung bình 7 ngày.
* Ý nghĩa: Hỗ trợ phát hiện tấn công DDoS hoặc lỗi cấu hình mạng.



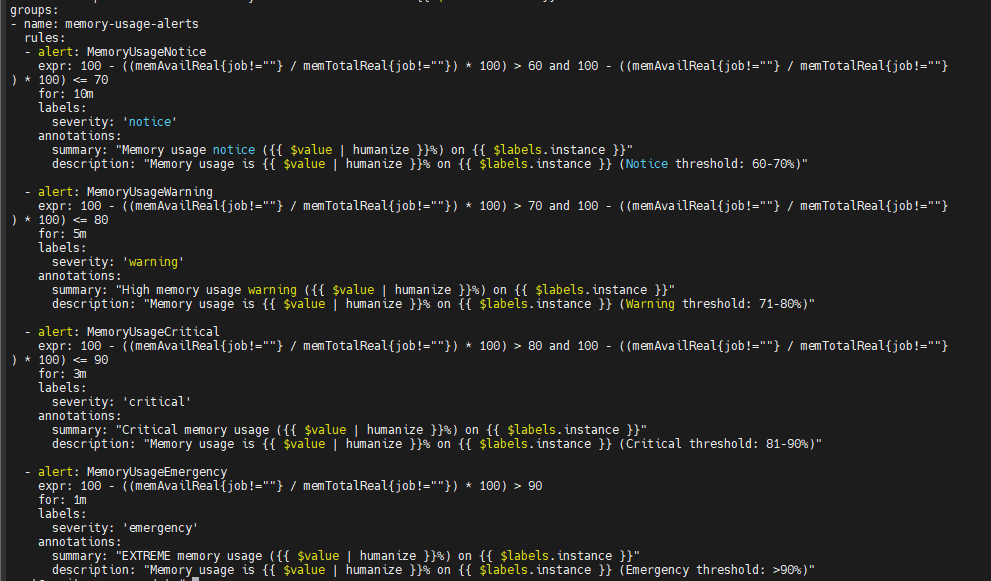
Hình 3.22. Rule cảnh báo khi có lưu lượng In/Out tăng cao

1. Cảnh báo sử dụng Memory

Mục tiêu: Giám sát mức sử dụng bộ nhớ và phân loại theo 4 mức:

* Notice: >60% đến 70%.
* Warning: >70% đến 80%.
* Critical: >80% đến 90%.
* Emergency: >90%.

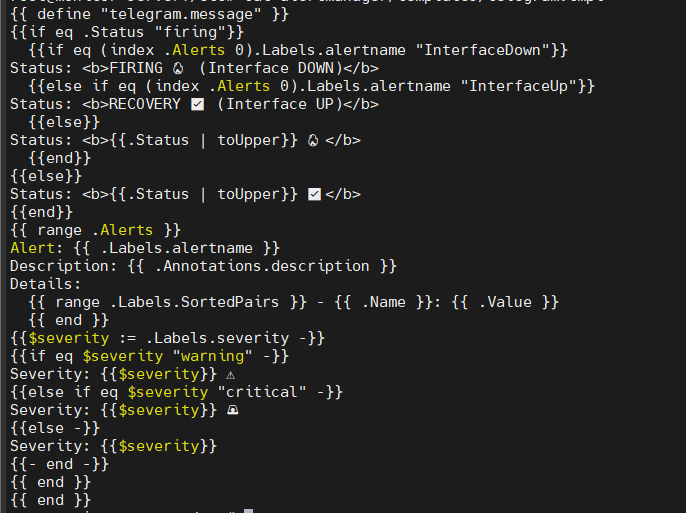
Rule: Sử dụng memAvailReal và memTotalReal để tính toán phần trăm bộ nhớ sử dụng.



Hình 3.23. Rule gửi cảnh báo theo tình trạng mức sử dụng bộ nhớ

#### Cơ chế gửi cảnh báo

Ta tạo các template theo từng rule cụ thể để biết được chính xác các thông tin mà ta nhận được. Ta sử dụng 2 kênh chính là telegram và email để thực hiện nhận thông tin

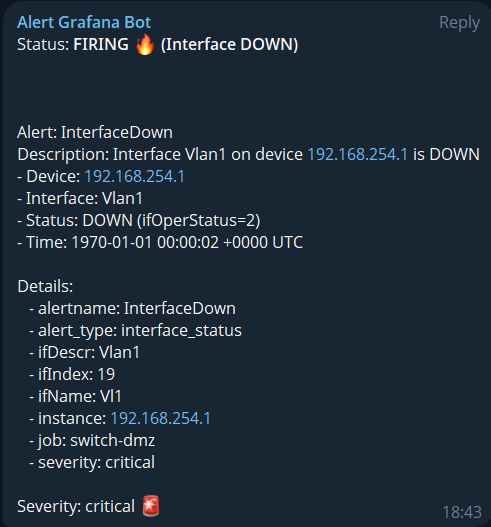


Hình 3.24. Ví dụ về template mẫu của cảnh báo Up/Down trên cổng mạng

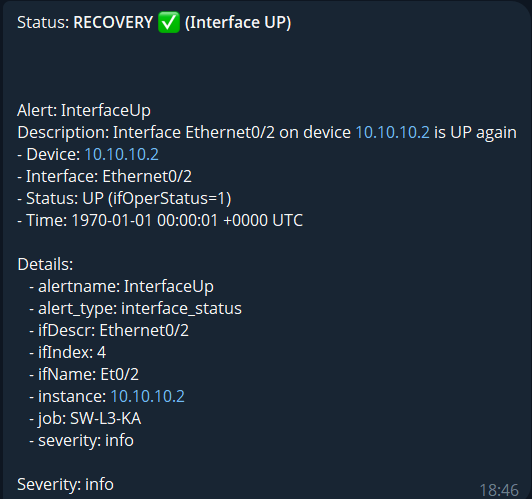
Kênh: Telegram và Email.

Định dạng thông báo (HTML):

* Kết quả trạng thái cổng mạng

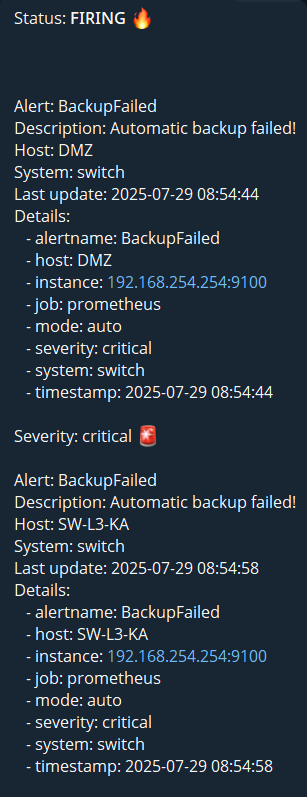


Hình 3.25. Cảnh báo khi trạng thái cổng down



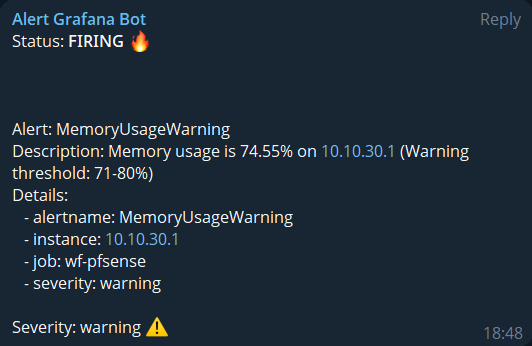
Hình 3.26. Cảnh báo khi trạng thái cổng được khôi phục trở lại

* Kết quả trạng thái khi backup failed



Hình 3.27. Cảnh báo khi backup failed

* Kết quả khi lượng memory sử dụng đạt đến ngưỡng cảnh báo



Hình 3.28. Cảnh báo khi lượng memory đạt 74.55%

## Kết luận chương

Trong chương này, tôi đã triển khai hoàn chỉnh hệ thống giám sát và cảnh báo an toàn mạng trong doanh nghiệp dựa trên Prometheus và Grafana. Quá trình triển khai được thực hiện theo các bước rõ ràng và logic, bao gồm:

* Cài đặt và cấu hình Prometheus để thu thập số liệu từ nhiều nguồn, chủ yếu thông qua SNMP Exporter và các exporter khác nhằm giám sát tình trạng phần cứng, dịch vụ và lưu lượng mạng.
* Xây dựng bộ luật cảnh báo (Alert Rules) chi tiết với nhiều kịch bản:
* Cảnh báo khi cổng mạng bị mất kết nối hoặc hoạt động trở lại (Interface Down/Up).
* Cảnh báo khi quy trình sao lưu tự động thất bại (Backup Failed).
* Cảnh báo khi lưu lượng mạng inbound/outbound tăng cao bất thường hoặc vượt ngưỡng quy định.
* Cảnh báo khi mức sử dụng bộ nhớ vượt các ngưỡng Notice, Warning, Critical, Emergency để phản ánh mức độ nghiêm trọng khác nhau.
* Tích hợp Alertmanager với các kênh thông báo Telegram và Email, bảo đảm cảnh báo được gửi tức thì, chứa đầy đủ thông tin về thiết bị, mức độ và thời gian xảy ra sự cố.
* Xây dựng dashboard trên Grafana để hiển thị số liệu và trạng thái cảnh báo theo thời gian thực, hỗ trợ đội ngũ vận hành theo dõi toàn cảnh hệ thống một cách trực quan.

Kết quả kiểm thử

* Cảnh báo được kích hoạt chính xác và nhanh chóng khi các điều kiện trong luật cảnh báo được đáp ứng. Ví dụ, khi một cổng mạng bị down, thông báo được gửi ngay lập tức đến Telegram với đầy đủ thông tin: tên thiết bị, giao diện, trạng thái, thời gian.
* Tích hợp Telegram và Email hoạt động ổn định, đảm bảo các sự kiện quan trọng được truyền tải đồng thời đến nhiều kênh liên lạc.
* Dashboard Grafana trực quan, dễ sử dụng: biểu đồ và trạng thái cảnh báo giúp người quản trị dễ dàng phân tích xu hướng lưu lượng, kiểm tra sức khỏe hệ thống, và phát hiện bất thường.
* Bộ luật cảnh báo phân cấp mức độ (Notice → Warning → Critical → Emergency) giúp đội vận hành ưu tiên xử lý các sự cố quan trọng trước, tối ưu hóa quy trình phản ứng sự cố.

Đánh giá hệ thống

* Việc sử dụng Prometheus và Grafana mang lại nhiều ưu điểm nổi bật:
* Hệ thống chi phí thấp, mã nguồn mở, dễ triển khai và tùy chỉnh theo nhu cầu thực tế của doanh nghiệp.
* Khả năng mở rộng cao: dễ dàng thêm luật cảnh báo mới hoặc tích hợp thêm exporter cho các dịch vụ khác.
* Cảnh báo đa kênh giúp giảm thiểu thời gian phản hồi khi sự cố xảy ra, nâng cao độ tin cậy cho hạ tầng mạng.
* Dashboard Grafana cung cấp cái nhìn toàn diện và trực quan về trạng thái mạng, hỗ trợ việc giám sát liên tục.

Tuy nhiên, hệ thống vẫn còn một số hạn chế cần cải thiện ở các chương sau, chẳng hạn như tối ưu hóa biểu thức cảnh báo phức tạp, giảm nhiễu (alert noise) và bổ sung cơ chế tự động khắc phục (auto-remediation).

KẾT LUẬN

* 1. Kết quả đạt được

Sau quá trình thực hiện đề tài “Xây dựng hệ thống giám sát và cảnh báo an toàn mạng bằng Prometheus và Grafana trong doanh nghiệp”, em xin rút ra một số kết luận như sau:

**THUẬN LỢI**

**Kiến thức chuyên môn: Đề tài giúp em củng cố và mở rộng kiến thức về hệ thống giám sát hiện đại, hiểu rõ cơ chế hoạt động của Prometheus, Grafana và Alertmanager, cũng như cách tích hợp SNMP, Node Exporter và các rule cảnh báo phức tạp trong môi trường mạng doanh nghiệp.**

**Tài liệu tham khảo phong phú: Prometheus và Grafana có cộng đồng người dùng lớn, tài liệu chính thức rõ ràng, cùng nhiều blog, video và diễn đàn hỗ trợ giúp quá trình triển khai và khắc phục sự cố thuận lợi.**

**Mô phỏng linh hoạt: Việc triển khai hệ thống trong môi trường ảo hóa (VMware, lab thử nghiệm) giúp kiểm thử các rule cảnh báo (interface down, backup failed, memory usage, traffic anomaly) mà không ảnh hưởng đến hạ tầng thật.**

**KHÓ KHĂN**

**Yêu cầu kiến thức mạng và hệ thống: Việc thiết lập metric SNMP, viết PromQL, và cấu hình rule cảnh báo đòi hỏi nền tảng kiến thức vững về mạng và quản trị hệ thống.**

**Tích hợp nhiều thành phần: Quá trình đồng bộ Prometheus – Alertmanager – Grafana và các exporter cần nhiều bước cấu hình, dễ phát sinh lỗi như định dạng dữ liệu, cảnh báo trùng lặp hoặc gửi sai nội dung.**

**Kiểm thử trong môi trường giới hạn: Một số kịch bản cảnh báo nâng cao (traffic anomaly dài hạn, backup lỗi định kỳ) chưa được kiểm thử đầy đủ trên hạ tầng thật, mới dừng ở môi trường lab.**

2. Những mặt đạt được và chưa đạt được

**Mặt đạt được:**

Triển khai thành công hệ thống giám sát: Bao gồm cài đặt Prometheus, cấu hình các Node Exporter/SNMP Exporter, thiết lập Alertmanager và tích hợp với Telegram/email.

Xây dựng dashboard trực quan: Sử dụng Grafana để hiển thị lưu lượng, trạng thái bộ nhớ, cổng mạng, và các chỉ số quan trọng theo thời gian thực.

Viết rule cảnh báo đa dạng: Hoàn thiện các rule như InterfaceDown/Up, BackupFailed, MemoryUsage (4 mức cảnh báo), High/Abnormal Traffic và kiểm thử gửi cảnh báo chính xác.

Quy trình cảnh báo rõ ràng: Khi sự cố xảy ra, cảnh báo hiển thị trạng thái (FIRING), thông tin chi tiết thiết bị, mức độ nghiêm trọng (critical, emergency), hỗ trợ phản ứng nhanh.

**Mặt chưa đạt:**

Chưa triển khai trên hạ tầng thật quy mô lớn: Mới kiểm thử trong lab, chưa đánh giá hiệu năng khi giám sát hàng trăm thiết bị.

Chưa tích hợp IDS/IPS: Mô hình hiện tại chưa kết hợp với Suricata/Snort để bổ sung phát hiện xâm nhập nâng cao.

Chưa tối ưu rule giảm cảnh báo giả: Một số rule cần tinh chỉnh thêm ngưỡng và điều kiện để tránh cảnh báo không cần thiết (false positive).

1. Hướng nghiên cứu

Trong tương lai, để hoàn thiện và mở rộng mô hình giám sát – cảnh báo mạng, em đề xuất:

Triển khai trong môi trường doanh nghiệp thực: Đánh giá hiệu năng và độ ổn định của hệ thống với thiết bị thật và lưu lượng thực tế.

Tích hợp IDS/IPS: Bổ sung Suricata hoặc Snort vào pipeline giám sát và cảnh báo qua Prometheus/Grafana.

Tối ưu hiệu năng và cảnh báo thông minh: Ứng dụng phân tích hành vi (anomaly detection) để tự động nhận diện sự cố thay vì chỉ dựa ngưỡng cố định.

Tự động hóa triển khai: Sử dụng Ansible/Docker để triển khai và quản lý Prometheus + Grafana + Alertmanager nhanh chóng, dễ mở rộng.

Kết nối với hệ thống giám sát tổng hợp: Tích hợp với Zabbix, ELK để tạo hệ sinh thái giám sát – phân tích log toàn diện hơn.

TÀI LIỆU THAM KHẢO

[1] Prometheus Documentation – Tài liệu chính thức của Prometheus, hướng dẫn cài đặt, cấu hình và viết rule cảnh báo - https://prometheus.io/docs/introduction/overview/

[2] Grafana Documentation – Hướng dẫn sử dụng Grafana để xây dựng dashboard và trực quan hóa dữ liệu giám sát mạng - https://grafana.com/docs/grafana/latest/

[3] Alertmanager Documentation – Tài liệu cấu hình Alertmanager để quản lý cảnh báo và tích hợp gửi thông báo qua các kênh (Telegram, Email) - https://prometheus.io/docs/alerting/latest/alertmanager/

[4] SNMP Exporter Documentation – Hướng dẫn triển khai SNMP Exporter để thu thập thông tin thiết bị mạng qua SNMP- https://github.com/prometheus/snmp\_exporter

[5] pfSense Documentation – Tài liệu cấu hình pfSense làm tường lửa và gateway, tích hợp SNMP để giám sát - https://docs.netgate.com/pfsense/en/latest/

[6] Suricata IDS/IPS Documentation – Hướng dẫn triển khai và cấu hình Suricata trong hệ thống giám sát bảo mật mạng - https://suricata.io/documentation/

[7] Telegram Bot API – Tài liệu về API của Telegram để gửi cảnh báo tự động từ Alertmanager - https://core.telegram.org/bots/api

[8] Node Exporter Documentation – Hướng dẫn triển khai Node Exporter để thu thập metric hệ thống (CPU, RAM, Disk) - https://prometheus.io/docs/guides/node-exporter/