



BÁO CÁO THỰC HÀNH

Bài thực hành 03: Tìm hiểu cơ sở lý thuyết và triển khai thực nghiệm hệ thống giám sát “Prometheus và Grafana”

Môn học: Công nghệ DevOps và Ứng dụng

Lớp: NT548.P11.MMCL

GVHD: ThS. Lê Anh Tuấn

THÀNH VIÊN THỰC HIỆN (Nhóm 14):

STT	Họ và tên	MSSV	Điểm
1	Lâm Bảo Duy	20521231	
2	Hồ Hải Dương	21520202	10

ĐÁNH GIÁ KHÁC:

Tổng thời gian thực hiện	2 tuần
Link GitHub	DevOps-Technology-Application/Lab03 at main · 13octt/DevOps-Technology-Application · GitHub
Phân chia công việc	Hồ Hải Dương: Tìm hiểu lý thuyết Lâm Bảo Duy: Triển khai Demo
Ý kiến (nếu có) + Khó khăn + Đề xuất, kiến nghị	

Phần bên dưới của báo cáo này là báo cáo chi tiết của nhóm thực hiện.

MỤC LỤC

A. CƠ SỞ LÝ THUYẾT	4
1. Tổng quan	4
2. Prometheus	4
2.1. Tổng quan	4
2.2. Kiến trúc của Prometheus.....	5
2.3. Các thành phần của Prometheus.....	5
2.4. Cơ chế hoạt động.....	5
2.5. Use cases phổ biến.....	6
3. Exporter trong hệ sinh thái Prometheus	6
3.1. Khái niệm về Exporter.....	6
3.2. Cách hoạt động của Exporter	6
3.3. Các loại Exporter phổ biến	7
3.4. Lợi ích của Exporter	7
3.5. Node Exporter.....	7
3.5.1. Tổng quan	7
3.5.2. Cách hoạt động của Node Exporter.....	7
3.5.3. Các loại metrics được Node Exporter thu thập	7
3.5.4. Cách triển khai	8
4. Grafana	8
4.1. Tổng quan	8
4.2. Đặc điểm nổi bật.....	8
4.3. Use cases phổ biến.....	9
B. TRIỂN KHAI HỆ THỐNG	10
1. Docker Compose và Cấu hình Prometheus	10
1.1 Docker Compose	10
1.1.1 services.prometheus	10
1.1.2. services.node-exporter	10
1.1.3. services.grafana.....	11
1.2. Prometheus Config	12
1.2.1. Global Configuration	12
1.2.2. Scrape Configurations.....	13
1.3. Run docker compose	13
1.3.1. Tổng quát	13
1.3.2. Các bước thực hiện	13
2. Prometheus	14
2.1. Target: node-exporter	14
2.2. Target: prometheus	14
2.3. Service Discovery Prometheus.....	15



2.4. Runtime & build information prometheus	15
2.5. Promethes Config	16
2.6. Các thông tin mà Node exporter thu được	17
3. Các câu query Prometheus thông dụng cho Node Exporter:.....	17
3.1. Tổng CPU Usage	17
3.2. CPU Usage per Core	17
3.3. Tổng Memory Usage.....	18
3.4. Memory Usage per Node.....	18
3.5. Disk Space Usage	18
3.6. Disk I/O Rate (Read + Write).	19
3.7. Disk Write Speed.....	19
3.8. Filesystem Free Space	19
3.9. Network Receive Rate.....	20
3.10. Load Average (1 minute).....	20
3.11. Uptime	20
3.12. Thông tin về hệ điều hành (OS)	21
3.13. Thông tin về bộ nhớ (Memory)	21
3.13.1. Tổng dung lượng bộ nhớ.....	21
3.13.2. Bộ nhớ còn trống.....	21
3.13.3. Bộ nhớ khả dụng	22
3.14. Thông tin về đĩa (Filesystem).....	22
3.14.1. Dung lượng đĩa tổng cộng.....	22
3.14.2. Dung lượng đĩa còn trống	22
3.14.3. Dung lượng đĩa khả dụng.....	23
3.15. Thông tin về mạng (Network)	23
3.15.1. Tốc độ nhận dữ liệu qua mạng (bytes per second):.....	23
3.15.2. Tốc độ truyền dữ liệu qua mạng (bytes per second):	23
4. Grafana	24
4.1. Tổng quát.....	24
4.2. Query Prometheus trong Grafna	24
4.2.1. Memory Usage (Sử dụng bộ nhớ).....	24
4.2.2. Network Traffic (Lưu lượng mạng)	25
4.3. Import Dashboar	26
TÀI LIỆU THAM KHẢO	31



BÁO CÁO CHI TIẾT

A. CƠ SỞ LÝ THUYẾT

1. Tổng quan

Hệ thống giám sát giúp theo dõi toàn bộ hạ tầng CNTT một cách liên tục. Điều này cho phép phát hiện sớm các sự cố, lỗi, hoặc rủi ro có thể ảnh hưởng đến hoạt động của tổ chức. Khi xảy ra bất kỳ vấn đề nào, hệ thống sẽ ngay lập tức đưa ra cảnh báo để đội ngũ kỹ thuật kịp thời xử lý.

Một hệ thống giám sát hiệu quả giúp đảm bảo rằng các dịch vụ và ứng dụng luôn trong trạng thái sẵn sàng, giảm thiểu tối đa thời gian gián đoạn, từ đó cải thiện trải nghiệm người dùng và duy trì lòng tin của khách hàng.

Qua việc thu thập và phân tích dữ liệu từ các thành phần hệ thống, giám sát giúp tối ưu hóa việc sử dụng tài nguyên, đảm bảo hiệu suất tối ưu và giảm lãng phí.

Hệ thống giám sát không chỉ phát hiện lỗi mà còn giúp nhận diện các hoạt động bất thường hoặc tiềm ẩn nguy cơ tấn công mạng, góp phần bảo vệ an toàn cho dữ liệu và hệ thống của tổ chức. Việc phát hiện và xử lý các vấn đề ngay từ sớm giúp giảm thiểu thiệt hại tiềm tàng, từ đó tiết kiệm chi phí và thời gian khắc phục sự cố.

Hệ thống giám sát không chỉ là công cụ kỹ thuật mà còn là yếu tố chiến lược giúp tổ chức đảm bảo sự ổn định, an toàn và phát triển bền vững trong môi trường công nghệ không ngừng thay đổi.

2. Prometheus

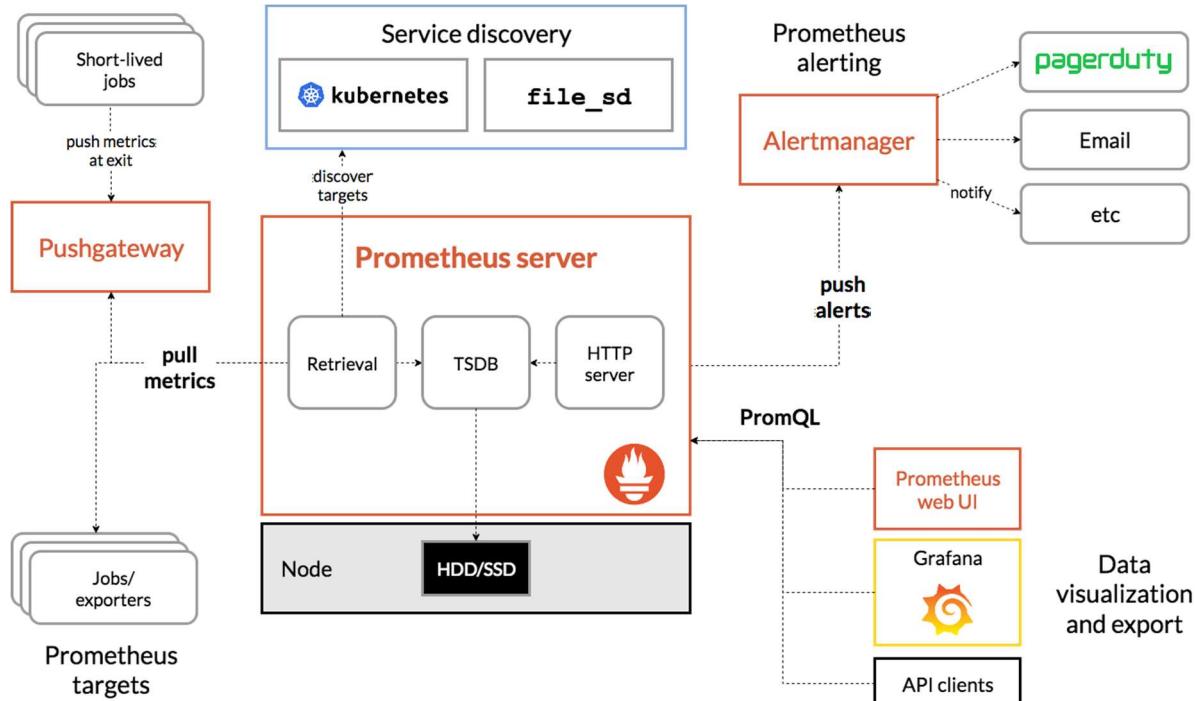


Hình 1 – Logo của Prometheus

2.1. Tổng quan

Prometheus là một hệ thống giám sát và cảnh báo mã nguồn mở, được thiết kế để xử lý dữ liệu dạng chuỗi thời gian (time-series) từ các dịch vụ và hạ tầng công nghệ thông tin.

2.2. Kiến trúc của Prometheus



Hình 2 – Kiến trúc Prometheus và hệ sinh thái

2.3. Các thành phần của Prometheus

1. Prometheus Server: Là trung tâm của hệ thống, chịu trách nhiệm thu thập, xử lý và lưu trữ dữ liệu từ các endpoint (exporter). Lưu trữ dữ liệu dạng time-series cục bộ.
2. Exporters: Là các ứng dụng hoặc agent thu thập metric từ hạ tầng hoặc dịch vụ. Ví dụ:
 - Node Exporter: Thu thập dữ liệu về hệ thống (CPU, RAM, Disk).
 - cAdvisor: Thu thập dữ liệu từ container.
 - Blackbox Exporter: Thu thập dữ liệu từ các endpoint HTTP/HTTPS.
3. Alertmanager: Quản lý và gửi cảnh báo dựa trên các rule được định nghĩa trong Prometheus. Hỗ trợ thông báo qua email, Slack, hoặc các công cụ khác.
4. PromQL (Prometheus Query Language): Ngôn ngữ mạnh mẽ dùng để truy vấn dữ liệu chuỗi thời gian, hỗ trợ phân tích và xây dựng biểu đồ.
5. Pushgateway: Được sử dụng để thu thập dữ liệu từ các job ngắn hạn (ephemeral jobs) hoặc các hệ thống không thể sử dụng cơ chế pull-based.
6. Service Discovery: Tự động phát hiện các endpoint mới trong môi trường động như Kubernetes, AWS, hoặc Docker Swarm.
7. Visualization Tools: Prometheus cung cấp các endpoint để tích hợp với Grafana, cho phép tạo dashboard và hiển thị dữ liệu trực quan.

2.4. Cơ chế hoạt động

Prometheus sử dụng cơ chế pull-based monitoring, trong đó Prometheus Server tự động và định kỳ truy xuất (pull) dữ liệu từ các endpoint được cấu hình trước.

Cách thức hoạt động:

- Endpoint Exposing Metrics: Các dịch vụ hoặc ứng dụng cần phải phơi bày (expose) các metric qua HTTP/HTTPS dưới dạng một endpoint, thường là /metrics.
- Pull Interval: Prometheus định kỳ lấy dữ liệu từ các endpoint theo khoảng thời gian được cấu hình (scrape interval).
- Centralized Control: Prometheus kiểm soát hoàn toàn quá trình thu thập dữ liệu, không yêu cầu các ứng dụng phải đẩy (push) dữ liệu đến hệ thống.

Lợi ích của Pull-based Monitoring:

- Prometheus chủ động lấy dữ liệu, đảm bảo tính ổn định và tránh lỗi từ cấu hình push endpoint không chính xác.
- Dữ liệu từ endpoint có thể được kiểm tra trực tiếp qua trình duyệt hoặc công cụ dòng lệnh.
- Phù hợp với các hệ thống phân tán hoặc containerized.
- Theo dõi nhiều endpoint mà không cần cơ chế push phức tạp.

Nhược điểm của Pull-based: Đối với các công việc ngắn hạn hoặc không liên tục (ephemeral jobs), cần sử dụng thêm công cụ Pushgateway để lưu trữ tạm thời các metric.

2.5. Use cases phổ biến

Prometheus được sử dụng rộng rãi trong các tình huống:

- Giám sát hệ thống hạ tầng: Theo dõi tài nguyên máy chủ, dịch vụ mạng và container với Node Exporter, cAdvisor, hoặc Blackbox Exporter.
- Ứng dụng cloud-native: Giám sát các ứng dụng microservices chạy trên Kubernetes và các nền tảng container khác.
- Cảnh báo thông minh: Sử dụng Alertmanager để gửi thông báo qua email, Slack, hoặc các kênh khác khi xảy ra sự cố.
- Tích hợp DevOps: Phân tích hiệu suất và độ ổn định của pipeline CI/CD, hỗ trợ tối ưu hóa quá trình phát triển và vận hành.
- Theo dõi hiệu suất ứng dụng: Tích hợp với các thư viện client để thu thập dữ liệu từ các ngôn ngữ lập trình phổ biến (Go, Java, Python,...).

3. Exporter trong hệ sinh thái Prometheus

3.1. Khái niệm về Exporter

Exporter là các thành phần trong hệ sinh thái Prometheus, được thiết kế để thu thập dữ liệu từ các hệ thống, ứng dụng, hoặc phần cứng và phơi bày dữ liệu dưới dạng metric. Các metric này thường được cung cấp qua một endpoint HTTP/HTTPS, cho phép Prometheus Server định kỳ truy cập và thu thập dữ liệu (scrape).

3.2. Cách hoạt động của Exporter

Exporter hoạt động như một "agent" trên hệ thống hoặc dịch vụ cần giám sát.

Nó truy vấn hoặc thu thập thông tin từ nguồn dữ liệu nội bộ (như CPU, RAM, Disk, hoặc Log).

Bài thực hành 03: Tìm hiểu cơ sở lý thuyết và triển khai thực nghiệm hệ thống giám sát “Prometheus và Grafana”



Dữ liệu sau đó được định dạng thành các metric phù hợp với Prometheus và được cung cấp qua một endpoint (thường là /metrics).

Prometheus Server truy cập endpoint này theo tần suất định kỳ để lấy dữ liệu.

3.3. Các loại Exporter phổ biến

Có nhiều exporter được thiết kế cho các ứng dụng hoặc hạ tầng như:

- cAdvisor: Theo dõi container.
- Blackbox Exporter: Kiểm tra tính khả dụng của endpoint qua HTTP, HTTPS, ICMP.
- MySQL Exporter: Giám sát cơ sở dữ liệu MySQL.
- Redis Exporter: Theo dõi hiệu suất của Redis.

3.4. Lợi ích của Exporter

Đa dạng dữ liệu: Cung cấp thông tin chi tiết về hạ tầng, ứng dụng và dịch vụ.

Dễ triển khai: Hầu hết các exporter đều dễ cài đặt và cấu hình.

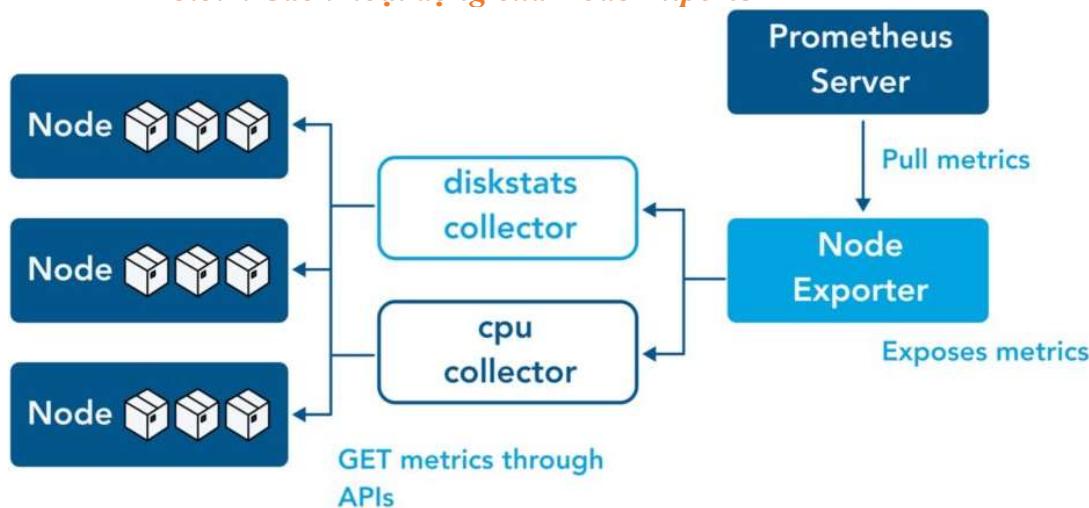
Khả năng mở rộng: Prometheus hỗ trợ nhiều exporter, giúp giám sát nhiều thành phần trong hệ thống.

3.5. Node Exporter

3.5.1. Tổng quan

Node Exporter là một trong những exporter phổ biến và quan trọng nhất trong hệ sinh thái Prometheus, được thiết kế để thu thập các chỉ số (metrics) về hệ thống máy chủ. Nó cung cấp thông tin chi tiết về trạng thái và hiệu suất của tài nguyên hệ thống như CPU, RAM, Disk, và Network. Node Exporter chủ yếu được sử dụng trên các hệ điều hành dựa trên Linux, nhưng cũng có thể hoạt động trên các nền tảng khác với mức độ hỗ trợ hạn chế.

3.5.2. Cách hoạt động của Node Exporter



Hình 3 – Cách hoạt động của Node Exporter trong hệ thống Prometheus

3.5.3. Các loại metrics được Node Exporter thu thập

- CPU Metrics: Thời gian sử dụng CPU (user, system, idle). Số lượng CPU đang hoạt động. Tỷ lệ tải CPU trung bình (load average).
- Memory Metrics: Bộ nhớ khả dụng (available memory). Bộ nhớ đang sử dụng (used memory). Số lượng swap đã sử dụng và khả dụng.



- Disk Metrics: Dung lượng đã sử dụng và còn trống của các phân vùng. Tốc độ đọc/ghi đĩa. Số lượng thao tác I/O đang thực hiện.
- Filesystem Metrics: Thông tin về dung lượng các file system. Trạng thái các điểm mount.
- Network Metrics: Lưu lượng dữ liệu gửi và nhận qua các giao diện mạng (interfaces). Số lượng lỗi hoặc gói bị rớt (dropped packets).
- System Metrics: Tỷ lệ tải trung bình (1 phút, 5 phút, 15 phút). Thời gian uptime của hệ thống.

3.5.4. Cách triển khai

Tải phiên bản mới nhất từ trang phát hành chính thức của Prometheus.

Giải nén và chạy Node Exporter từ dòng lệnh:

```
./node_exporter
```

Theo mặc định, Node Exporter sẽ lắng nghe trên cổng 9100.

Thêm Node Exporter vào file cấu hình prometheus.yml:

```
scrape_configs:  
  - job_name: 'node_exporter'  
    static_configs:  
      - targets: ['<IP của máy chủ>:9100']
```

Truy cập <http://<IP>:9100/metrics> từ trình duyệt để xem danh sách các metric được Node Exporter phơi bày.

4. Grafana



Hình 4 – Logo của Grafana

4.1. Tổng quan

Grafana là một nền tảng mã nguồn mở mạnh mẽ dùng để giám sát và trực quan hóa dữ liệu. Grafana được tạo ra để tích hợp với Prometheus nhưng hiện nay đã hỗ trợ rất nhiều nguồn dữ liệu khác như Graphite, Elasticsearch, InfluxDB, MySQL, và AWS CloudWatch.

4.2. Đặc điểm nổi bật

1. Hỗ trợ đa nguồn dữ liệu

- Grafana có khả năng kết nối với nhiều nguồn dữ liệu khác nhau, từ cơ sở dữ liệu SQL đến các công cụ time-series như Prometheus hoặc InfluxDB.
- Người dùng có thể tích hợp đồng thời nhiều datasource trên cùng một dashboard.

2. Tùy chỉnh dashboard

- Grafana cho phép tạo các dashboard linh hoạt, cung cấp nhiều loại biểu đồ (graph, gauge, table...) để hiển thị dữ liệu. Hỗ trợ chế độ light và dark, phù hợp với các nhu cầu sử dụng khác nhau.

3. Tích hợp cảnh báo (Alerting)

- Hỗ trợ tạo và cấu hình cảnh báo trực tiếp trên các panel.
- Cảnh báo có thể gửi qua email, Slack, PagerDuty hoặc các công cụ khác khi phát hiện bất thường.

4. Hệ thống plugin mở rộng

- Grafana hỗ trợ cài đặt plugin để mở rộng chức năng, từ visualization đến tích hợp với các hệ thống khác.
- Người dùng có thể tải các plugin từ Grafana Marketplace.

5. Khả năng cộng tác

- Dashboard có thể được chia sẻ dưới dạng link hoặc snapshot, giúp đội nhóm dễ dàng cộng tác và phân tích dữ liệu.

6. Bảo mật và xác thực

- Hỗ trợ các cơ chế xác thực mạnh mẽ như OAuth, LDAP, và SSO.
- Có thể quản lý người dùng và phân quyền truy cập chi tiết.

4.3. Use cases phổ biến

Giám sát hạ tầng và ứng dụng: Kết hợp với Prometheus để hiển thị trạng thái của server, container, và các dịch vụ.

Giám sát kinh doanh: Tích hợp với cơ sở dữ liệu để theo dõi KPI và các chỉ số hiệu suất kinh doanh.

Quản lý dữ liệu IoT: Theo dõi dữ liệu từ các thiết bị IoT trong thời gian thực.

Phân tích log: Sử dụng với Elasticsearch hoặc Loki để trực quan hóa log và phát hiện vấn đề nhanh chóng.

B. TRIỂN KHAI HỆ THỐNG

1. Docker Compose và Cấu hình Prometheus

1.1 Docker Compose

1.1.1 services.prometheus

```
2 services:
3   prometheus:
4     image: prom/prometheus:latest
5     container_name: prometheus
6     ports:
7       - "9090:9090"
8     volumes:
9       - ./prometheus.yml:/etc/prometheus/prometheus.yml
10    command:
11      - '--config.file=/etc/prometheus/prometheus.yml'
```

Hình 5 – Định nghĩa services Prometheus

Định nghĩa dịch vụ Prometheus trong file docker-compose.yml.

1. **image:** Sử dụng image chính thức prom/prometheus:latest từ Docker Hub.
 - o Đảm bảo luôn chạy phiên bản mới nhất.
2. **container_name:** Đặt tên container là **prometheus** để dễ quản lý.
3. **ports:** Mở cổng **9090** trên máy chủ host và ánh xạ nó đến cổng **9090** trong container.
 - o Truy cập giao diện web Prometheus tại <http://localhost:9090>.
4. **volumes:** Gắn file cấu hình prometheus.yml từ máy host vào container tại đường dẫn `/etc/prometheus/prometheus.yml`.
 - o Giúp Prometheus đọc được cấu hình thu thập số liệu.
5. **command:** Chạy Prometheus với file cấu hình được chỉ định (`--config.file=/etc/prometheus/prometheus.yml`).

Đặt tham số khởi động, chỉ định Prometheus sử dụng file cấu hình được mount.

1.1.2. services.node-exporter

```
13 services:
14   node-exporter:
15     image: prom/node-exporter:latest| You, 5 hours ago
16     container_name: node-exporter
17     ports:
18       - "9100:9100"
19     command:
20       - '--path.rootfs=/host'
21     volumes:
22       - /:/host:ro
```

Hình 6 – Định nghĩa service Node Exporter

Định nghĩa dịch vụ Node Exporter trong file docker-compose.yml.

- **image:**
 - Sử dụng image chính thức prom/node-exporter:latest từ Docker Hub.
 - Node Exporter là công cụ thu thập số liệu về tài nguyên và trạng thái của hệ thống.
- **container_name:**
 - Đặt tên container là **node-exporter** để dễ nhận biết.
- **ports:**
 - Mở cổng **9100** trên máy chủ host và ánh xạ nó đến cổng **9100** trong container.
 - Node Exporter sẽ cung cấp dữ liệu tại <http://<host-ip>:9100/metrics>.
- **command:**
 - Sử dụng tham số `--path.rootfs=/host` để chỉ định đường dẫn root (/) trong host mà Node Exporter sẽ thu thập dữ liệu.
- **volumes:**
 - Gắn volume từ thư mục root (/) của máy chủ host vào container tại /host với quyền chỉ đọc (ro).
 - Điều này cho phép Node Exporter truy cập vào thông tin hệ thống host để thu thập dữ liệu.

1.1.3. services.grafana

```
23  grafana:  
24    image: grafana/grafana:latest  
25    container_name: grafana  
26    ports:  
27      - "3000:3000"  
28    environment:  
29      - GF_SECURITY_ADMIN_USER=admin  
30      - GF_SECURITY_ADMIN_PASSWORD=admin  
31    volumes:  
32      - grafana_data:/var/lib/grafana
```

Hình 7 – Định nghĩa service Grafana

Định nghĩa dịch vụ Grafana trong file docker-compose.yml.

- **image:**
 - Sử dụng image chính thức grafana/grafana:latest từ Docker Hub.
 - Grafana là công cụ trực quan hóa dữ liệu, sử dụng để tạo dashboard và biểu đồ từ dữ liệu Prometheus.

- **container_name:**
 - Đặt tên container là **grafana** để dễ quản lý.
- **ports:**
 - Mở cổng **3000** trên máy chủ host và ánh xạ nó đến cổng **3000** trong container.
 - Truy cập giao diện web của Grafana tại <http://localhost:3000>.
- **environment:**
 - Thiết lập biến môi trường:
 - **GF_SECURITY_ADMIN_USER=admin**: Tạo tài khoản quản trị với username là admin.
 - **GF_SECURITY_ADMIN_PASSWORD=admin**: Đặt mật khẩu cho tài khoản quản trị là admin.
 - Dùng để đăng nhập vào giao diện web của Grafana.
- **volumes:**
 - Tạo volume tên **grafana_data** để lưu trữ dữ liệu Grafana tại /var/lib/grafana bên trong container.
 - Giúp bảo toàn dữ liệu dashboard khi container bị xóa hoặc khởi động lại.

1.2. Prometheus Config

```
Lab03 > prometheus.yml > [ ] scrape_configs > {} 1 > [ ] static_configs > {} 0
      prometheus.json - Prometheus configuration file (prometheus.json) | You, 5 hours ago | 1 author (You)
      1   global:
      2     |   scrape_interval: 15s
      3
      4   scrape_configs:
      5     - job_name: 'prometheus'
      6       static_configs:
      7         |   - targets: ['localhost:9090']
      8
      9     - job_name: 'node-exporter'
     10       static_configs:
     11         |   - targets: ['192.168.1.61:9100']
```

Hình 8 – Cấu hình file `prometheus.yml` với các job Prometheus và Node Exporter

1.2.1. Global Configuration

yaml
Copy code
global:
 scrape_interval: 15s

- **scrape_interval: 15s:**
 - Thiết lập chu kỳ thu thập dữ liệu từ các target là **15 giây**.
 - Đây là cấu hình mặc định cho tất cả các job, trừ khi được override ở từng job cụ thể.

1.2.2. Scrape Configurations

- Cấu hình các job mà Prometheus sẽ thu thập dữ liệu từ.

Job 1: Prometheus

yaml

Copy code

```
- job_name: 'prometheus'  
  static_configs:  
    - targets: ['localhost:9090']  
      • job_name: 'prometheus':  
          o Định nghĩa một job tên là prometheus để giám sát chính Prometheus.  
      • targets: ['localhost:9090']:  
          o Prometheus tự giám sát hoạt động của chính nó bằng cách thu thập số liệu tại địa chỉ http://localhost:9090/metrics.
```

Job 2: Node Exporter

yaml

Copy code

```
- job_name: 'node-exporter'  
  static_configs:  
    - targets: ['192.168.1.61:9100']  
      • job_name: 'node-exporter':  
          o Định nghĩa một job tên là node-exporter để giám sát hệ thống.  
      • targets: ['192.168.1.61:9100']:  
          o Prometheus thu thập số liệu từ Node Exporter tại địa chỉ http://192.168.1.61:9100/metrics.
```

1.3. Run docker compose



Hình 9 – Khởi chạy 3 container bằng Docker Compose

1.3.1. Tóm tắt

Triển khai một hệ thống giám sát cơ bản bao gồm:

- Prometheus**: Thu thập và lưu trữ số liệu.
- Node Exporter**: Thu thập số liệu từ hệ thống máy chủ.
- Grafana**: Hiển thị số liệu qua các dashboard trực quan.

1.3.2. Các bước thực hiện

Cấu hình file docker-compose.yml:

- File được thiết kế để chạy các container cho **Prometheus**, **Node Exporter**, và **Grafana**.

- Cấu hình bao gồm:
 - Map các cổng cần thiết (9090, 9100, 3000).
 - Mount volume để lưu trữ dữ liệu lâu dài.
 - Đặt thông tin đăng nhập cho Grafana (admin/admin).

2. Prometheus

The screenshot shows the Prometheus web interface at `localhost:9090/targets`. It displays two targets: 'node-exporter' and 'prometheus'. Both targets are marked as '1 / 1 up' with a green 'UP' status indicator. The 'node-exporter' target has an endpoint of `http://192.168.1.61:9100/metrics` and labels `instance="192.168.1.61:9100"` and `job="node-exporter"`. The 'prometheus' target has an endpoint of `http://localhost:9090/metrics` and labels `instance="localhost:9090"` and `job="prometheus"`. Below each target, there is a 'Last scrape' timestamp and a '26ms' or '6ms' latency indicator.

Hình 10 – Giao diện Prometheus hiển thị trạng thái các target Node Exporter và Prometheus

2.1. Target: node-exporter

- **Endpoint:** `http://192.168.1.61:9100/metrics`
 - Đây là địa chỉ mà Prometheus sử dụng để thu thập số liệu từ dịch vụ **Node Exporter** đang chạy trên máy chủ có IP 192.168.1.61 qua cổng 9100.
- **Labels:**
 - `instance: 192.168.1.61:9100`
 - Thông tin định danh duy nhất cho target.
 - `job: node-exporter`
 - Tên công việc được định nghĩa trong file cấu hình Prometheus (`prometheus.yml`).
- **Last scrape:** 3.492s ago
 - Prometheus vừa thu thập số liệu từ target này cách đây khoảng **3.492 giây**.
- **State:** UP
 - Trạng thái hiện tại là **UP**, nghĩa là Node Exporter hoạt động bình thường và Prometheus đã thành công trong việc thu thập số liệu.

2.2. Target: prometheus

- **Endpoint:** `http://localhost:9090/metrics`
 - Đây là địa chỉ mà Prometheus tự thu thập số liệu của chính nó (self-monitoring).
- **Labels:**
 - `instance: localhost:9090`
 - Định danh cho instance của chính Prometheus.

Bài thực hành 03: Tìm hiểu cơ sở lý thuyết và triển khai thực nghiệm hệ thống giám sát “Prometheus và Grafana”

- job: prometheus
 - Tên công việc được định nghĩa trong file cấu hình.
- Last scrape: 10.987s ago
 - Prometheus vừa tự thu thập số liệu cách đây khoảng **10.987 giây**.
- State: UP
 - Trạng thái hiện tại là **UP**, nghĩa là Prometheus hoạt động ổn định.

2.3. Service Discovery Prometheus

The screenshot shows two sections of the Prometheus UI under the 'Status > Service discovery' tab. The top section is for the 'node-exporter' job, which has 1 target. The bottom section is for the 'prometheus' job, also with 1 target. Both sections show 'Discovered labels' and 'Target labels' side-by-side. In the 'Discovered labels' column, there are several labels like '_address', '_metrics_path', '_scheme', '_scrape_interval', '_scrape_timeout', and 'job'. In the 'Target labels' column, for the first target, there are labels like 'instance', 'job', and 'job_label'. The interface includes search and filter bars at the top.

Hình 11 – Giao diện Service Discovery của Prometheus hiển thị các labels đã khám phá

2.4. Runtime & build information prometheus

The screenshot shows the 'Status > Runtime & build information' tab. It contains two main sections: 'Build information' and 'Runtime information'. The 'Build information' section lists details such as Version (3.1.0), Revision (7086161a93b262aa0949dbf2aba15a5a7b13e0a3), Branch (HEAD), BuildUser (root@74c225e2044f), BuildDate (20250102-13:52:43), and GoVersion (go1.23.4). The 'Runtime information' section lists various metrics like Start Time (2025-01-04T05:29:03Z), Working Directory (/prometheus), Configuration Reload (Successful), Last Successful Configuration Reload (2025-01-04T05:29:03Z), WAL Corruptions (0), Goroutines (39), GOMAXPROCS (12), GOMEMLIMIT (3634425446), GOGC (75), GODEBUG (Storage Retention 15d).

Hình 12 – Thông tin Runtime và Build của Prometheus

2.5. Prometheus Config

```
global:
  scrape_interval: 15s
  scrape_timeout: 10s
  scrape_protocols:
    - OpenMetricsText1.0.0
    - OpenMetricsText0.0.1
    - PrometheusText1.0.0
    - PrometheusText0.0.4
  evaluation_interval: 1m
runtime:
  gogc: 75
scrape_configs:
  - job_name: prometheus
    honor_timestamps: true
    track_timestamps_staleness: false
    scrape_interval: 15s
    scrape_timeout: 10s
    scrape_protocols:
      - OpenMetricsText1.0.0
      - OpenMetricsText0.0.1
      - PrometheusText1.0.0
      - PrometheusText0.0.4
    metrics_path: /metrics
    scheme: http
    enable_compression: true
    follow_redirects: true
    enable_http2: true
    static_configs:
      - targets:
          - localhost:9090
  - job_name: node-exporter
    honor_timestamps: true
    track_timestamps_staleness: false
    scrape_interval: 15s
    scrape_timeout: 10s
    scrape_protocols:
      - OpenMetricsText1.0.0
      - OpenMetricsText0.0.1
      - PrometheusText1.0.0
      - PrometheusText0.0.4
    metrics_path: /metrics
    scheme: http
    enable_compression: true
    follow_redirects: true
    enable_http2: true
    static_configs:
      - targets:
```

Hình 13 – Nội dung file cấu hình Prometheus.yml chi tiết

2.6. Các thông tin mà Node exporter thu được

```
# HELP go_gc_duration_seconds A summary of the pause duration of garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 2.5201e-05
go_gc_duration_seconds{quantile="0.25"} 4.9502e-05
go_gc_duration_seconds{quantile="0.5"} 7.1503e-05
go_gc_duration_seconds{quantile="0.75"} 0.000113705
go_gc_duration_seconds{quantile="1"} 0.000477826
go_gc_duration_seconds_sum 0.018325458
go_gc_duration_seconds_count 196
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 8
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info{version="go1.22.5"} 1
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 3.711128e+06
# HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if freed.
# TYPE go_memstats_alloc_bytes_total counter
go_memstats_alloc_bytes_total 3.81076728e+08
# HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling bucket hash table.
# TYPE go_memstats_buck_hash_sys_bytes gauge
go_memstats_buck_hash_sys_bytes 1.534596e+06
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 5.092126e+06
# HELP go_memstats_gc_sys_bytes Number of bytes used for garbage collection system metadata.
# TYPE go_memstats_gc_sys_bytes gauge
go_memstats_gc_sys_bytes 3.270264e+06
# HELP go_memstats_heap_alloc_bytes Number of heap bytes allocated and still in use.
# TYPE go_memstats_heap_alloc_bytes gauge
go_memstats_heap_alloc_bytes 3.711128e+06
# HELP go_memstats_heap_idle_bytes Number of heap bytes waiting to be used.
# TYPE go_memstats_heap_idle_bytes gauge
go_memstats_heap_idle_bytes 2.49856e+06
# HELP go_memstats_heap_inuse_bytes Number of heap bytes that are in use.
# TYPE go_memstats_heap_inuse_bytes gauge
go_memstats_heap_inuse_bytes 5.218304e+06
# HELP go_memstats_heap_objects Number of allocated objects.
# TYPE go_memstats_heap_objects gauge
go_memstats_heap_objects 41876
# HELP go_memstats_heap_released_bytes Number of heap bytes released to OS.
# TYPE go_memstats_heap_released_bytes gauge
go_memstats_heap_released_bytes 2.285568e+06
# HELP go_memstats_heap_sys_bytes Number of heap bytes obtained from system.
# TYPE go_memstats_heap_sys_bytes gauge
go_memstats_heap_sys_bytes 7.716864e+06
# HELP go_memstats_last_gc_time_seconds Number of seconds since 1970 of last garbage collection.
# TYPE go_memstats_last_gc_time_seconds gauge
go_memstats_last_gc_time_seconds 1.7359712811864803e+09
# HELP go_memstats_lookups_total Total number of pointer lookups.
```

Hình 14 – Danh sách metric được phô bày từ Node Exporter tại endpoint /metrics

3. Các câu query Prometheus thông dụng cho Node Exporter:

3.1. Tổng CPU Usage

Hiển thị tỷ lệ sử dụng CPU của tất cả các core.

100 - (avg by (instance)(rate(node_cpu_seconds_total{mode="idle"}[1m])) * 100)

Series	Value
[instance="192.168.1.61:9100"]	0.4278253139239183

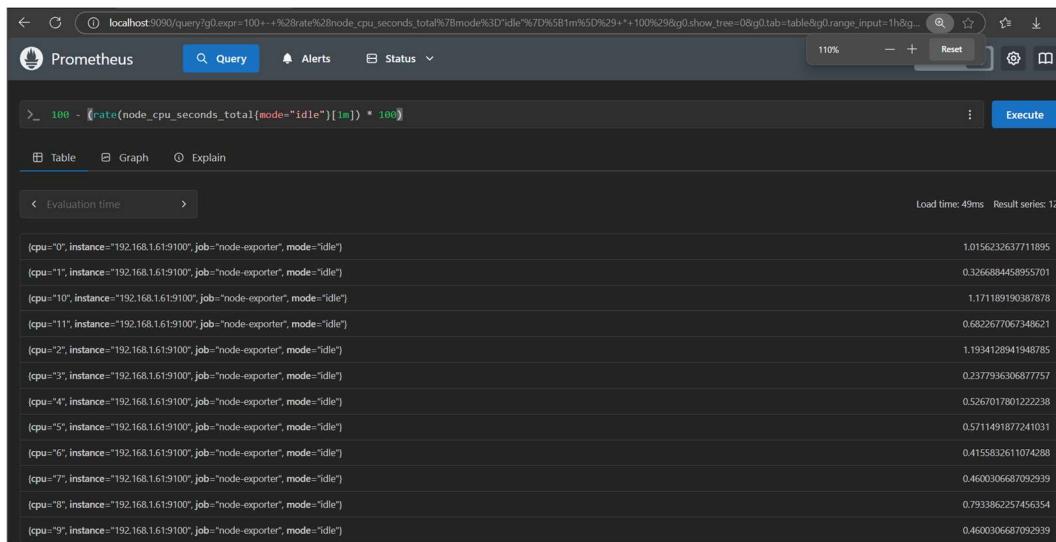
Hình 15 – Truy vấn CPU Usage bằng PromQL và hiển thị kết quả dạng bảng

3.2. CPU Usage per Core

Hiển thị tỷ lệ sử dụng CPU của từng core.

100 - (rate(node_cpu_seconds_total{mode="idle"}[1m])) * 100)

Bài thực hành 03: Tìm hiểu cơ sở lý thuyết và triển khai thực nghiệm hệ thống giám sát “Prometheus và Grafana”

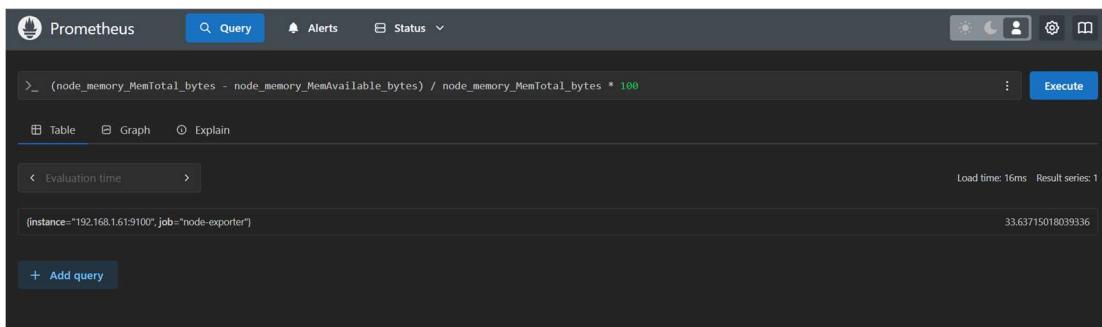


Hình 16 – Danh sách giá trị CPU Usage từ nhiều instance sử dụng PromQL

3.3. Tổng Memory Usage

Tổng lượng bộ nhớ đang sử dụng (bộ nhớ thực tế).

$$\frac{(\text{node_memory_MemTotal_bytes} - \text{node_memory_MemAvailable_bytes})}{\text{node_memory_MemTotal_bytes}} * 100$$

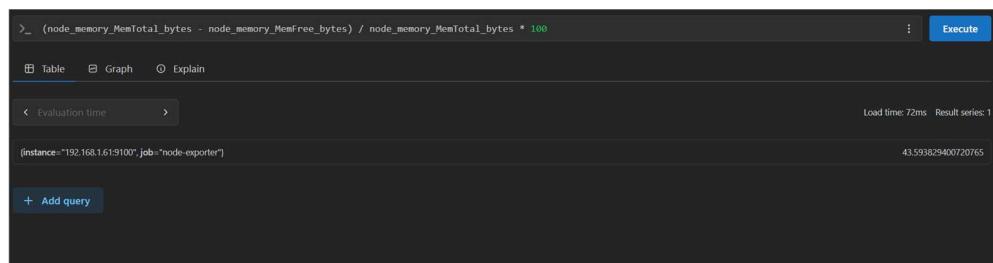


Hình 17 – Truy vấn Memory Usage (tổng dung lượng sử dụng) bằng PromQL

3.4. Memory Usage per Node

Hiển thị tỷ lệ sử dụng bộ nhớ của từng node.

$$\frac{(\text{node_memory_MemTotal_bytes} - \text{node_memory_MemFree_bytes})}{\text{node_memory_MemTotal_bytes}} * 100$$



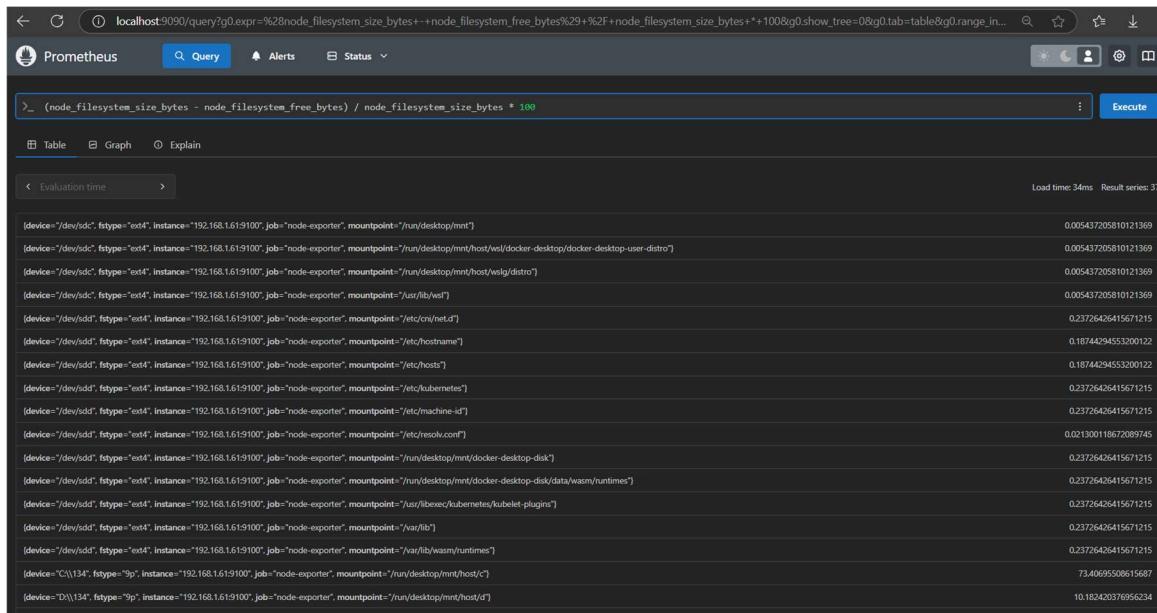
Hình 18 – Truy vấn Memory Usage (dung lượng đã sử dụng) bằng PromQL

3.5. Disk Space Usage

Sử dụng đĩa (dung lượng đã sử dụng) trên một node.

Bài thực hành 03: Tìm hiểu cơ sở lý thuyết và triển khai thực nghiệm hệ thống giám sát "Prometheus và Grafana"

$(\text{node_filesystem_size_bytes} - \text{node_filesystem_free_bytes}) / \text{node_filesystem_size_bytes} * 100$



Hình 19 – Truy vấn Filesystem Usage từ Node Exporter bằng PromQL

3.6. Disk I/O Rate (Read + Write)

Tỷ lệ đọc và ghi đĩa trong 1 phút (bytes).

rate(node_disk_io_time_seconds_total[1m])

3.7. Disk Write Speed

Tốc độ ghi trên đĩa (bytes per second).

rate(node_disk_written_bytes_total[1m])



Hình 20 – Truy vấn Disk I/O Time bằng PromQL và hiển thị kết quả dạng bảng

3.8. Filesystem Free Space

Dung lượng đĩa còn trống.

node_filesystem_free_bytes

Prometheus		Query	Alerts	Status	
					Load time: 41ms Result series: 3
< Evaluation time	>				101042395136
node_filesystem_free_bytes{device="/dev/sd1", ftype="ext4", instance="192.168.1.61:9100"} job="node-exporter", mountpoint="/run/desktop/mnt"					101042395136
node_filesystem_free_bytes{device="/dev/sd1", ftype="ext4", instance="192.168.1.61:9100"} job="node-exporter", mountpoint="/run/desktop/mnt/host/wsl/docker-desktop-user-distro"					101042395136
node_filesystem_free_bytes{device="/dev/sd1", ftype="ext4", instance="192.168.1.61:9100"} job="node-exporter", mountpoint="/run/desktop/mnt/host/wsl/distro"					101042395136
node_filesystem_free_bytes{device="/dev/sd1", ftype="ext4", instance="192.168.1.61:9100"} job="node-exporter", mountpoint="/usr/lib/wsl"					101042395136
node_filesystem_free_bytes{device="/dev/sd1", ftype="ext4", instance="192.168.1.61:9100"} job="node-exporter", mountpoint="/etc/cn.net.d"					107635933952
node_filesystem_free_bytes{device="/dev/sd1", ftype="ext4", instance="192.168.1.61:9100"} job="node-exporter", mountpoint="/etc/hostname"					2015256576
node_filesystem_free_bytes{device="/dev/sd1", ftype="ext4", instance="192.168.1.61:9100"} job="node-exporter", mountpoint="/etc/hosts"					2015256576
node_filesystem_free_bytes{device="/dev/sd1", ftype="ext4", instance="192.168.1.61:9100"} job="node-exporter", mountpoint="/etc/kubernetes"					107635933952
node_filesystem_free_bytes{device="/dev/sd1", ftype="ext4", instance="192.168.1.61:9100"} job="node-exporter", mountpoint="/etc/machine-id"					107635933952
node_filesystem_free_bytes{device="/dev/sd1", ftype="ext4", instance="192.168.1.61:9100"} job="node-exporter", mountpoint="/etc/resolv.conf"					403742720
node_filesystem_free_bytes{device="/dev/sd1", ftype="ext4", instance="192.168.1.61:9100"} job="node-exporter", mountpoint="/run/desktop/mnt/docker-desktop-disk"					107635933952
node_filesystem_free_bytes{device="/dev/sd1", ftype="ext4", instance="192.168.1.61:9100"} job="node-exporter", mountpoint="/run/desktop/mnt/docker-desktop-disk/data/wasm/runtime"					107635933952
node_filesystem_free_bytes{device="/dev/sd1", ftype="ext4", instance="192.168.1.61:9100"} job="node-exporter", mountpoint="/usr/libexec/kubernetes/kubeblet-plugins"					107635933952
node_filesystem_free_bytes{device="/dev/sd1", ftype="ext4", instance="192.168.1.61:9100"} job="node-exporter", mountpoint="/var/lib"					107635933952
node_filesystem_free_bytes{device="/dev/sd1", ftype="ext4", instance="192.168.1.61:9100"} job="node-exporter", mountpoint="/var/lib/wasm/runtime"					107635933952
node_filesystem_free_bytes{device="\\\"134\\\"", ftype="9p", instance="192.168.1.61:9100"} job="node-exporter", mountpoint="/run/desktop/mnt/host/c"					5756788608
node_filesystem_free_bytes{device="\\\"134\\\"", ftype="9p", instance="192.168.1.61:9100"} job="node-exporter", mountpoint="/run/desktop/mnt/host/d"					24110219564
node_filesystem_free_bytes{device="\\\"134\\\"", ftype="9p", instance="192.168.1.61:9100"} job="node-exporter", mountpoint="/run/desktop/mnt/host/wsl/docker-desktop-bind-mounts/Ubuntu-20.04/eac4532a655bc7f54ed9d2987058498093827705e0d10b2656fe3fae1\"}					24110219564
node_filesystem_free_bytes{device="\\\"134\\\"", ftype="9p", instance="192.168.1.61:9100"} job="node-exporter", mountpoint="/run/desktop/mnt/host/e"					430450516
node_filesystem_free_bytes{device="\\\"134\\\"", ftype="9p", instance="192.168.1.61:9100"} job="node-exporter", mountpoint="/run/desktop/mnt/host/p"					7556791508
node_filesystem_free_bytes{device="\\\"drivers\\\"", ftype="9p", instance="192.168.1.61:9100"} job="node-exporter", mountpoint="/usr/lib/wsl/drivers"					5758788608

Hình 21 – Danh sách giá trị Filesystem Free Bytes từ Node Exporter

3.9. Network Receive Rate

Tốc độ nhận dữ liệu trên giao diện mạng (bytes per second).

rate(node_network_receive_bytes_total[1m])

The screenshot shows the Prometheus web interface. The top navigation bar includes 'Prometheus' (with a logo), 'Query' (highlighted in blue), 'Alerts', 'Status', and a user icon. Below the navigation is a search bar containing the query: `>_ rate(node_network_receive_bytes_total[1m])`. To the right of the search bar are 'Execute' and 'Help' buttons. Below the search bar are tabs for 'Table' (selected), 'Graph', and 'Explain'. The main content area displays the results of the query. It includes an 'Evaluation time' dropdown set to '1m', a 'Load time: 34ms' message, and 'Result series:' information. The results show two series: one for 'eth0' (IP 192.168.1.61) and one for 'lo' (IP 192.168.1.61). Both series are labeled 'job="node-exporter"'. At the bottom left is a '+ Add query' button.

Hình 22 – Truy vấn Network Receive Bytes bằng PromQL

3.10. Load Average (1 minute)

Tải trung bình của hệ thống trong 1 phút.

node load1

The screenshot shows the Prometheus web interface. At the top, there are tabs for 'Query', 'Alerts', and 'Status'. The main area has a search bar with the query '`>_ node_load1`' and an 'Execute' button. Below the search bar are buttons for 'Table', 'Graph', and 'Explain'. A time range selector shows '< Evaluation time >'. On the right, it says 'Load time: 25ms Result series: 0.01'. At the bottom left is a '+ Add query' button.

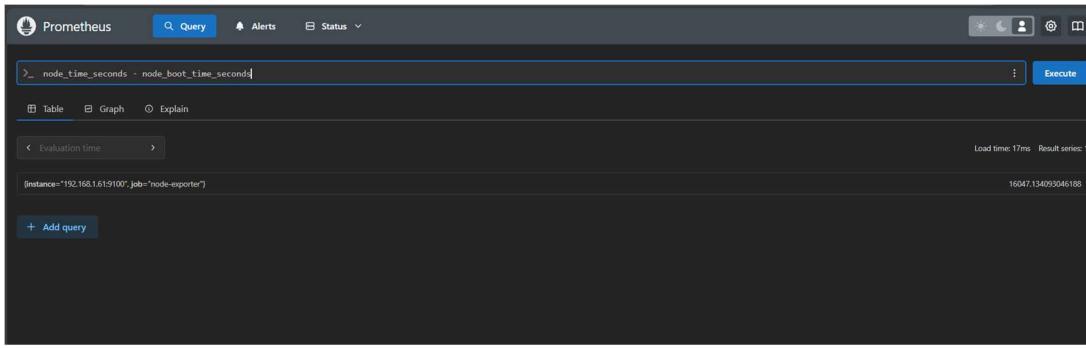
Hình 23 – Truy vấn Load Average 1 phút bằng PromQL

3.11. Uptime

Thời gian hoạt động của hệ thống kể từ khi khởi động.

node time seconds - node boot time seconds

Bài thực hành 03: Tìm hiểu cơ sở lý thuyết và triển khai thực nghiệm hệ thống giám sát “Prometheus và Grafana”



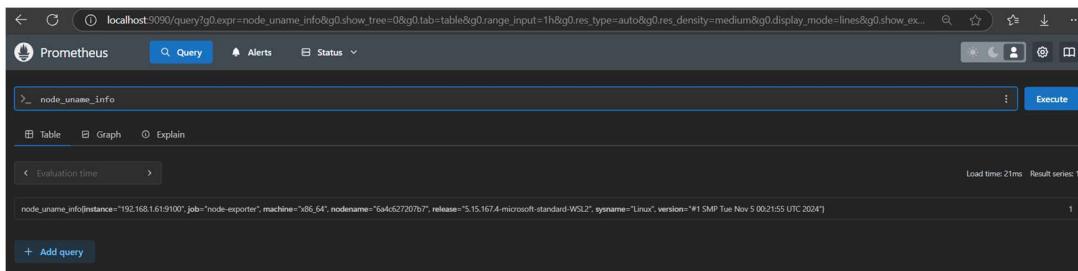
Hình 24 – Truy vấn Uptime của hệ thống bằng PromQL

3.12. Thông tin về hệ điều hành (OS)

Lấy thông tin về hệ điều hành, kernel và phiên bản kernel: node_uname_info

Câu query này sẽ trả về các thông tin như:

- sysname**: Tên hệ điều hành (Linux, Darwin, v.v.)
- nodename**: Tên máy chủ (hostname)
- release**: Phiên bản hệ điều hành
- version**: Phiên bản kernel
- machine**: Kiến trúc của hệ thống (x86_64, arm64, v.v.)

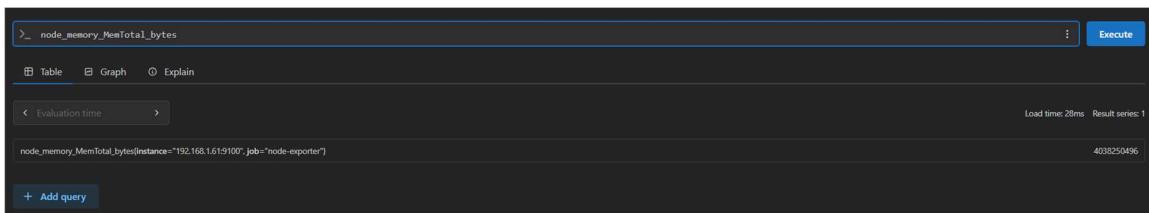


Hình 25 – Truy vấn thông tin hệ điều hành và kernel bằng PromQL

3.13. Thông tin về bộ nhớ (Memory)

3.13.1. Tổng dung lượng bộ nhớ

Lấy tổng bộ nhớ, bộ nhớ còn lại, bộ nhớ đã sử dụng, và bộ nhớ swap: node_memory_MemTotal_bytes

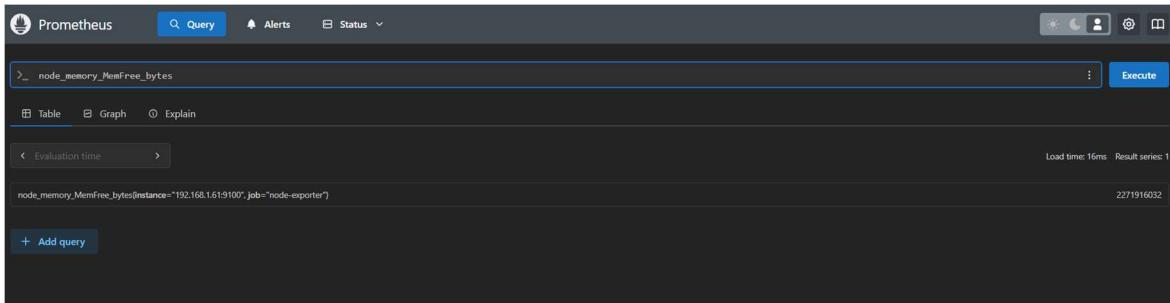


Hình 26 – Truy vấn tổng dung lượng bộ nhớ khả dụng bằng PromQL

3.13.2. Bộ nhớ còn trống

node_memory_MemFree_bytes

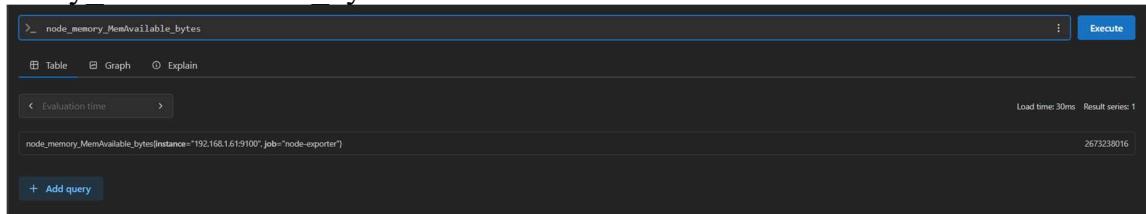
Bài thực hành 03: Tìm hiểu cơ sở lý thuyết và triển khai thực nghiệm hệ thống giám sát “Prometheus và Grafana”



Hình 27 – Truy vấn dung lượng bộ nhớ trống (Free Memory) bằng PromQL

3.13.3. Bộ nhớ khả dụng

`node_memory_MemAvailable_bytes`



Hình 28 – Truy vấn bộ nhớ sẵn sàng (Available Memory) bằng PromQL

3.14. Thông tin về đĩa (Filesystem)

Lấy thông tin về dung lượng đĩa tổng cộng, dung lượng còn trống và dung lượng khả dụng:

3.14.1. Dung lượng đĩa tổng cộng

`node_filesystem_size_bytes`

>_ node_filesystem_size_bytes		Execute
Table	Graph	Explain
Evaluation time		
Load time: 13ms	Result series: 37	
node_filesystem_size_bytes[device='/dev/sdc', fstype='ext4', instance='192.168.1.61:9100', job='node-exporter', mountpoint='/run/desktop/mnt']	1081101176832	
node_filesystem_size_bytes[device='/dev/sdc', fstype='ext4', instance='192.168.1.61:9100', job='node-exporter', mountpoint='/run/desktop/mnt/host/wsl/docker-desktop/docker-desktop-user-distro']	1081101176832	
node_filesystem_size_bytes[device='/dev/sdc', fstype='ext4', instance='192.168.1.61:9100', job='node-exporter', mountpoint='/run/desktop/mnt/host/wsl/distro']	1081101176832	
node_filesystem_size_bytes[device='/dev/sdc', fstype='ext4', instance='192.168.1.61:9100', job='node-exporter', mountpoint='/usr/lib/wsl']	1081101176832	
node_filesystem_size_bytes[device='/dev/sdd', fstype='ext4', instance='192.168.1.61:9100', job='node-exporter', mountpoint='/etc/cni/net.d']	1081101176832	
node_filesystem_size_bytes[device='/dev/sdd', fstype='ext4', instance='192.168.1.61:9100', job='node-exporter', mountpoint='/etc/hostname']	2019123200	
node_filesystem_size_bytes[device='/dev/sdd', fstype='ext4', instance='192.168.1.61:9100', job='node-exporter', mountpoint='/etc/hosts']	2019123200	
node_filesystem_size_bytes[device='/dev/sdd', fstype='ext4', instance='192.168.1.61:9100', job='node-exporter', mountpoint='/etc/kubernetes']	1081101176832	
node_filesystem_size_bytes[device='/dev/sdd', fstype='ext4', instance='192.168.1.61:9100', job='node-exporter', mountpoint='/etc/machine-id']	1081101176832	
node_filesystem_size_bytes[device='/dev/sdd', fstype='ext4', instance='192.168.1.61:9100', job='node-exporter', mountpoint='/etc/resolv.conf']	403828736	
node_filesystem_size_bytes[device='/dev/sdd', fstype='ext4', instance='192.168.1.61:9100', job='node-exporter', mountpoint='/run/desktop/mnt/docker-desktop-disk']	1081101176832	
node_filesystem_size_bytes[device='/dev/sdd', fstype='ext4', instance='192.168.1.61:9100', job='node-exporter', mountpoint='/run/desktop/mnt/docker-desktop-disk/data/wasm/runtimes']	1081101176832	
node_filesystem_size_bytes[device='/dev/sdd', fstype='ext4', instance='192.168.1.61:9100', job='node-exporter', mountpoint='/usr/libexec/kubernetes/kublet-plugins']	1081101176832	
node_filesystem_size_bytes[device='C:\\\134', fstype='9p', instance='192.168.1.61:9100', job='node-exporter', mountpoint='/run/desktop/mnt/host/c']	21653535872	
node_filesystem_size_bytes[device='D:\\\\134', fstype='9p', instance='192.168.1.61:9100', job='node-exporter', mountpoint='/run/desktop/mnt/host/d']	26843541504	

Hình 29 – Danh sách tổng dung lượng Filesystem từ Node Exporter

3.14.2. Dung lượng đĩa còn trống

`node_filesystem_free_bytes`

Bài thực hành 03: Tìm hiểu cơ sở lý thuyết và triển khai thực nghiệm hệ thống giám sát “Prometheus và Grafana”

>_ node_filesystem_free_bytes		Execute
Table	Graph	Explain
< Evaluation time >		Load time: 17ms Result series: 37
node_filesystem_free_bytes{device="/dev/sdc", ftype="ext4", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/run/desktop/mnt"}		1081042366464
node_filesystem_free_bytes{device="/dev/sdc", ftype="ext4", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/run/desktop/mnt/host/wsl/docker-desktop-user-distro"}		1081042366464
node_filesystem_free_bytes{device="/dev/sdc", ftype="ext4", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/run/desktop/mnt/host/wslg/distro"}		1081042366464
node_filesystem_free_bytes{device="/dev/sdc", ftype="ext4", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/usr/lib/wsl"}		1081042366464
node_filesystem_free_bytes{device="/dev/sdd", ftype="ext4", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/etc/cn/net.d"}		1078530326528
node_filesystem_free_bytes{device="/dev/sdd", ftype="ext4", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/etc/hostname"}		2014228480
node_filesystem_free_bytes{device="/dev/sdd", ftype="ext4", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/etc/hosts"}		2014228480
node_filesystem_free_bytes{device="/dev/sdd", ftype="ext4", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/etc/kubernetes"}		1078530326528
node_filesystem_free_bytes{device="/dev/sdd", ftype="ext4", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/etc/machine-id"}		1078530326528
node_filesystem_free_bytes{device="/dev/sdd", ftype="ext4", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/etc/resolv.conf"}		403742720
node_filesystem_free_bytes{device="/dev/sdd", ftype="ext4", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/run/desktop/mnt/docker-desktop-disk"}		1078530326528
node_filesystem_free_bytes{device="/dev/sdd", ftype="ext4", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/run/desktop/mnt/docker-desktop-disk/data/wasm/unittimes"}		1078530326528
node_filesystem_free_bytes{device="/dev/sdd", ftype="ext4", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/usr/libexec/kubernetes/kubelet-plugins"}		1078530326528
node_filesystem_free_bytes{device="/dev/sdd", ftype="ext4", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/var/lib"}		1078530326528
node_filesystem_free_bytes{device="/dev/sdd", ftype="ext4", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/var/lib/wasm/unittimes"}		1078530326528
node_filesystem_free_bytes{device="C:\134", ftype="9p", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/run/desktop/mnt/host/c"}		57543561216
node_filesystem_free_bytes{device="D:\134", ftype="9p", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/run/desktop/mnt/host/d"}		24110219264

Hình 30 – Danh sách dung lượng Filesystem còn trống từ Node Exporter

3.14.3. Dung lượng đĩa khả dụng

node_filesystem_avail_bytes

>_ node_filesystem_avail_bytes		Execute
Table	Graph	Explain
< Evaluation time >		Load time: 28ms Result series: 37
node_filesystem_avail_bytes{device="/dev/sdc", ftype="ext4", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/run/desktop/mnt"}		1026050011136
node_filesystem_avail_bytes{device="/dev/sdc", ftype="ext4", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/run/desktop/mnt/host/wsl/docker-desktop-user-distro"}		1026050011136
node_filesystem_avail_bytes{device="/dev/sdc", ftype="ext4", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/run/desktop/mnt/host/wslg/distro"}		1026050011136
node_filesystem_avail_bytes{device="/dev/sdc", ftype="ext4", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/usr/lib/wsl"}		1026050011136
node_filesystem_avail_bytes{device="/dev/sdd", ftype="ext4", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/etc/cn/net.d"}		1023537946624
node_filesystem_avail_bytes{device="/dev/sdd", ftype="ext4", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/etc/hostname"}		2014220288
node_filesystem_avail_bytes{device="/dev/sdd", ftype="ext4", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/etc/hosts"}		2014220288
node_filesystem_avail_bytes{device="/dev/sdd", ftype="ext4", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/etc/kubernetes"}		1023537946624
node_filesystem_avail_bytes{device="/dev/sdd", ftype="ext4", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/etc/machine-id"}		1023537946624
node_filesystem_avail_bytes{device="/dev/sdd", ftype="ext4", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/etc/resolv.conf"}		403742720
node_filesystem_avail_bytes{device="/dev/sdd", ftype="ext4", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/run/desktop/mnt/docker-desktop-disk"}		1023537946624
node_filesystem_avail_bytes{device="/dev/sdd", ftype="ext4", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/run/desktop/mnt/docker-desktop-disk/data/wasm/unittimes"}		1023537946624
node_filesystem_avail_bytes{device="/dev/sdd", ftype="ext4", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/usr/libexec/kubernetes/kubelet-plugins"}		1023537946624
node_filesystem_avail_bytes{device="C:\134", ftype="9p", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/run/desktop/mnt/host/c"}		57537946624
node_filesystem_avail_bytes{device="D:\134", ftype="9p", instance="192.168.1.61:9100", job="node-exporter", mountpoint="/run/desktop/mnt/host/d"}		2033537946624

Hình 31 – Danh sách dung lượng Filesystem khả dụng từ Node Exporter

3.15. Thông tin về mạng (Network)

Lấy số byte nhận và truyền qua các giao diện mạng:

3.15.1. Tốc độ nhận dữ liệu qua mạng (bytes per second):

rate(node_network_receive_bytes_total[1m])

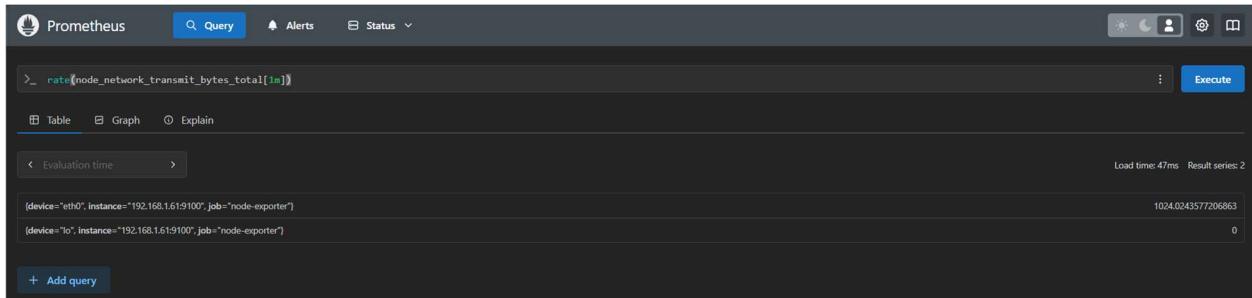
>_ rate{node_network_receive_bytes_total[1m]}		Execute
Table	Graph	Explain
< Evaluation time >		Load time: 117ms Result series: 2
(device="eth0", instance="192.168.1.61:9100", job="node-exporter")		50.53445632125158
(device="lo", instance="192.168.1.61:9100", job="node-exporter")		0

Hình 32 – Truy vấn Network Receive Bytes Rate từ Node Exporter

3.15.2. Tốc độ truyền dữ liệu qua mạng (bytes per second):

rate(node_network_transmit_bytes_total[1m])

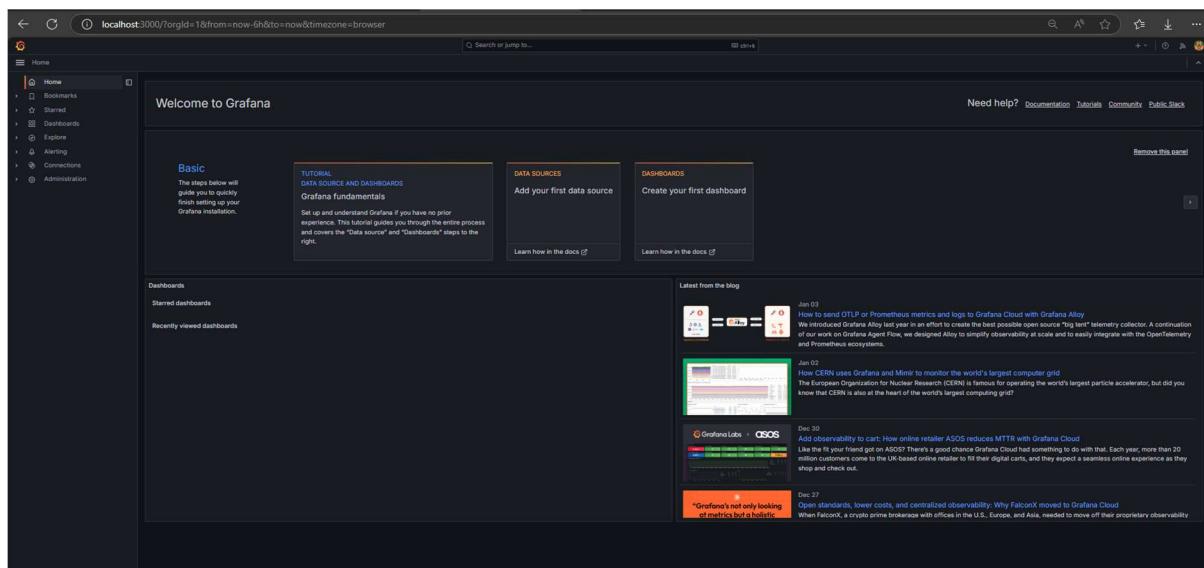
Bài thực hành 03: Tìm hiểu cơ sở lý thuyết và triển khai thực nghiệm hệ thống giám sát “Prometheus và Grafana”



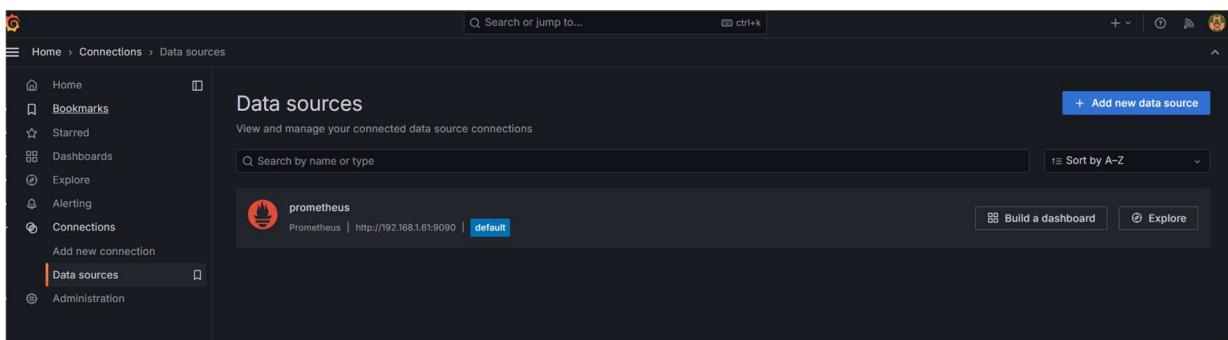
Hình 33 – Truy vấn Network Transmit Bytes Rate từ Node Exporter

4. Grafana

4.1. Tổng quát



Hình 34 – Giao diện trang chủ của Grafana



Hình 35 – Cấu hình Data sources Prometheus trong Grafana

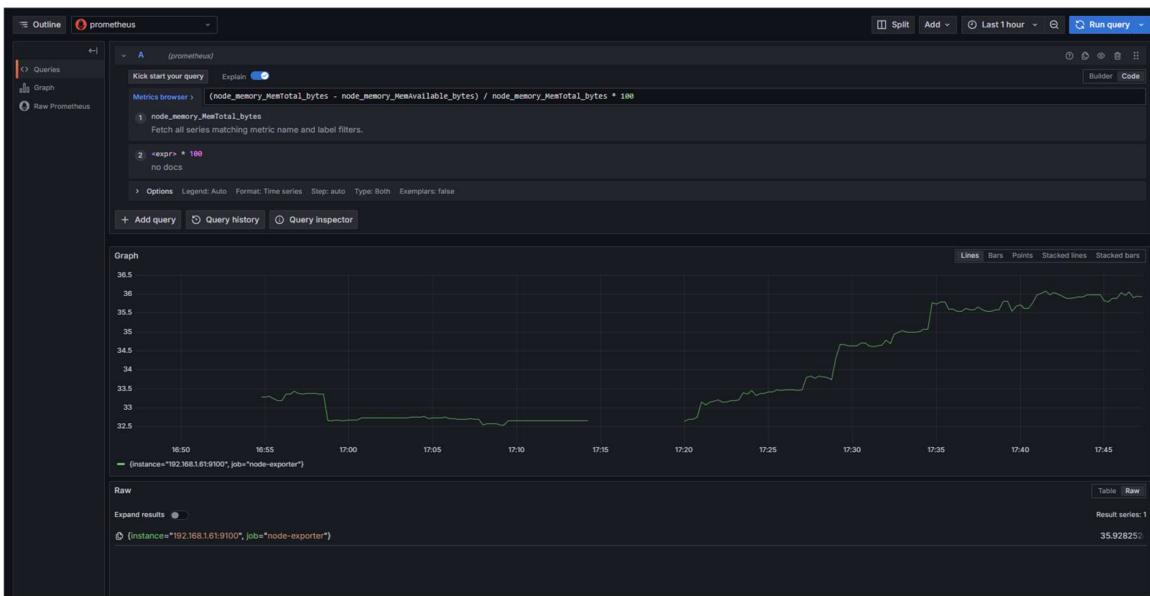
4.2. Query Prometheus trong Grafana

4.2.1. Memory Usage (Sử dụng bộ nhớ)

Để theo dõi tỷ lệ sử dụng bộ nhớ:

Bài thực hành 03: Tìm hiểu cơ sở lý thuyết và triển khai thực nghiệm hệ thống giám sát “Prometheus và Grafana”

25



Hình 36 – Hiển thị Memory Usage trong Grafana với biểu đồ chi tiết

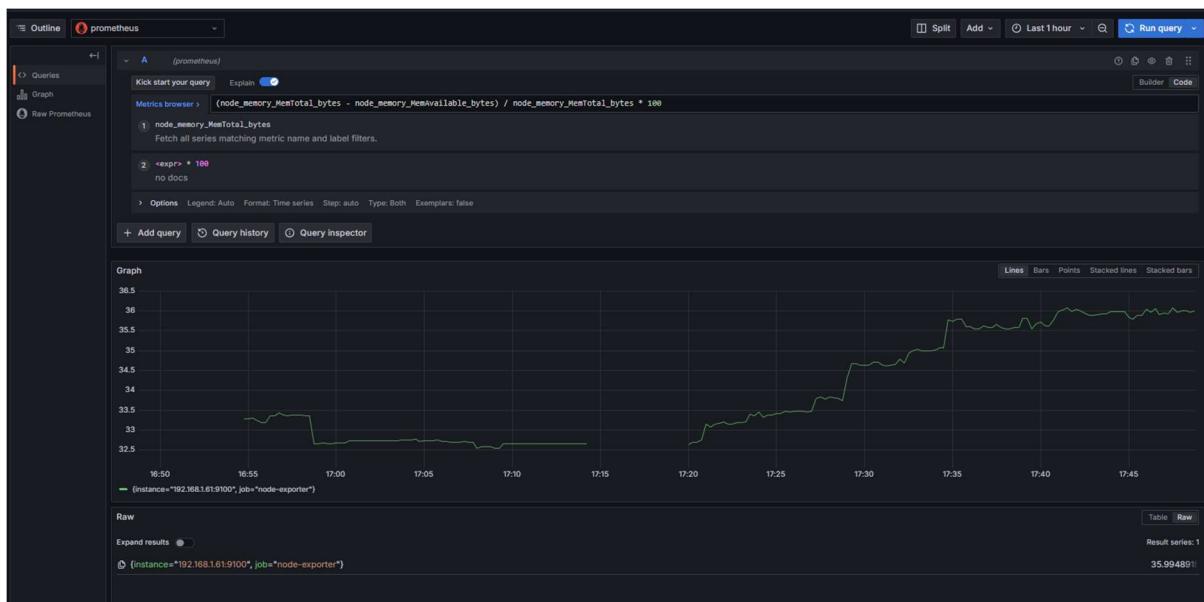
Giải thích:

- node_memory_MemTotal_bytes: Tổng bộ nhớ.
- node_memory_MemAvailable_bytes: Bộ nhớ còn khả dụng.
- Câu query này tính tỷ lệ bộ nhớ đã sử dụng.

4.2.2. Network Traffic (Lưu lượng mạng)

Để theo dõi lưu lượng mạng (bytes nhận và gửi qua các interface):

- **Tốc độ nhận dữ liệu qua mạng** (bytes per second):



Hình 37 – Hiển thị Memory Usage trong Grafana với biểu đồ chi tiết

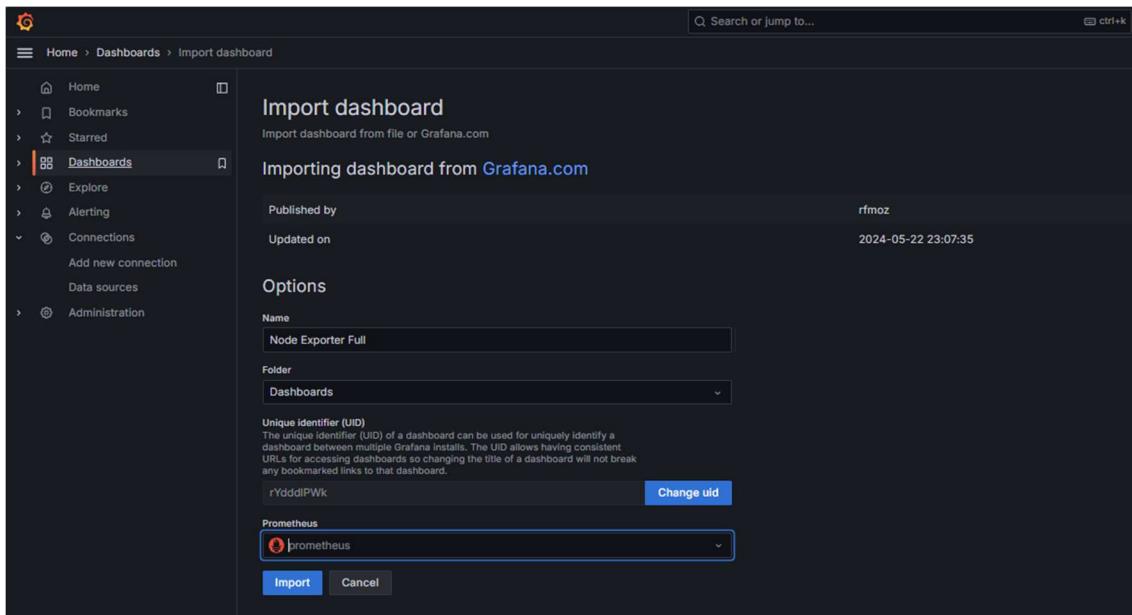
Giải thích:

- node_network_receive_bytes_total: Số byte nhận được qua các interface mạng.
- node_network_transmit_bytes_total: Số byte gửi đi qua các interface mạng.
- rate(...[1m]): Tính tốc độ thay đổi của chỉ số trong 1 phút.

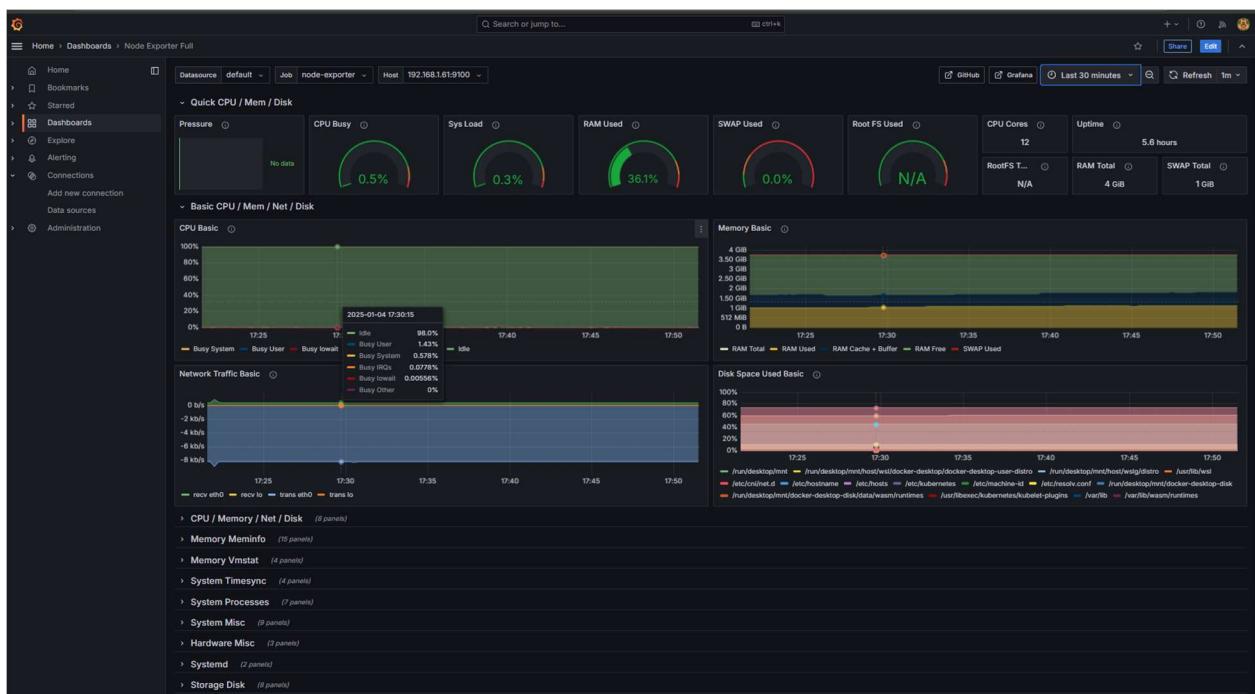
Bài thực hành 03: Tìm hiểu cơ sở lý thuyết và triển khai thực nghiệm hệ thống giám sát “Prometheus và Grafana”

26

4.3. Import Dashboard

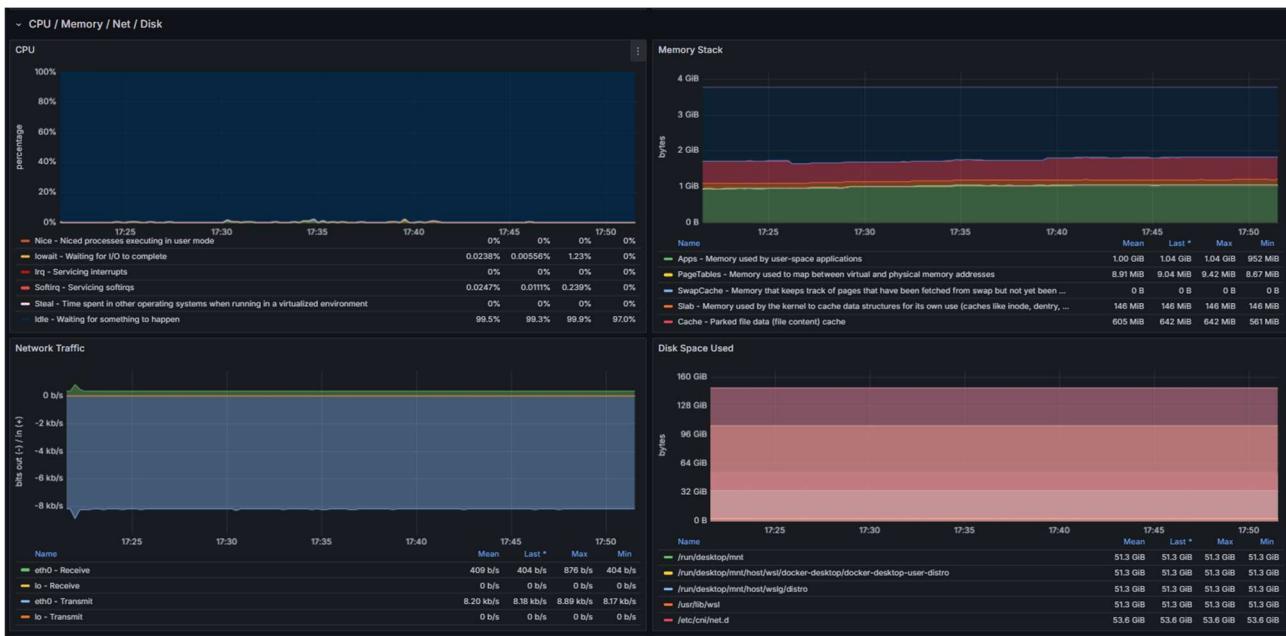


Hình 38 – Nhập dashboard Node Exporter Full từ Grafana.com

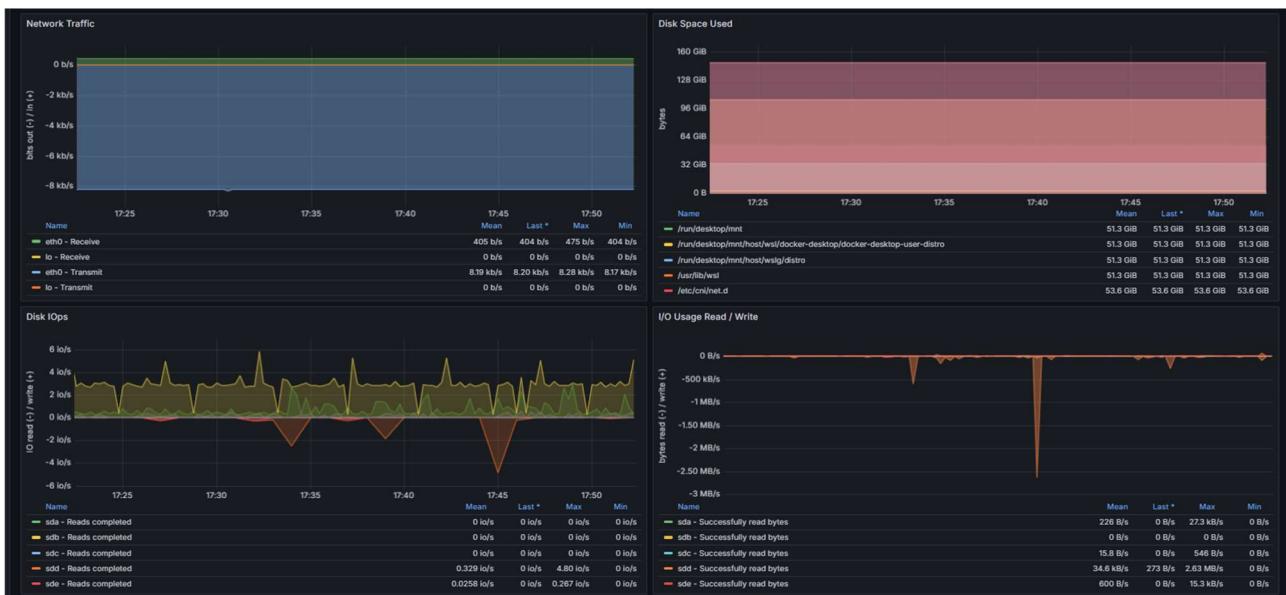


Hình 39 – Dashboard Node Exporter Full hiển thị dữ liệu CPU, RAM, Disk và Network trong Grafana

Bài thực hành 03: Tìm hiểu cơ sở lý thuyết và triển khai thực nghiệm hệ thống giám sát “Prometheus và Grafana”

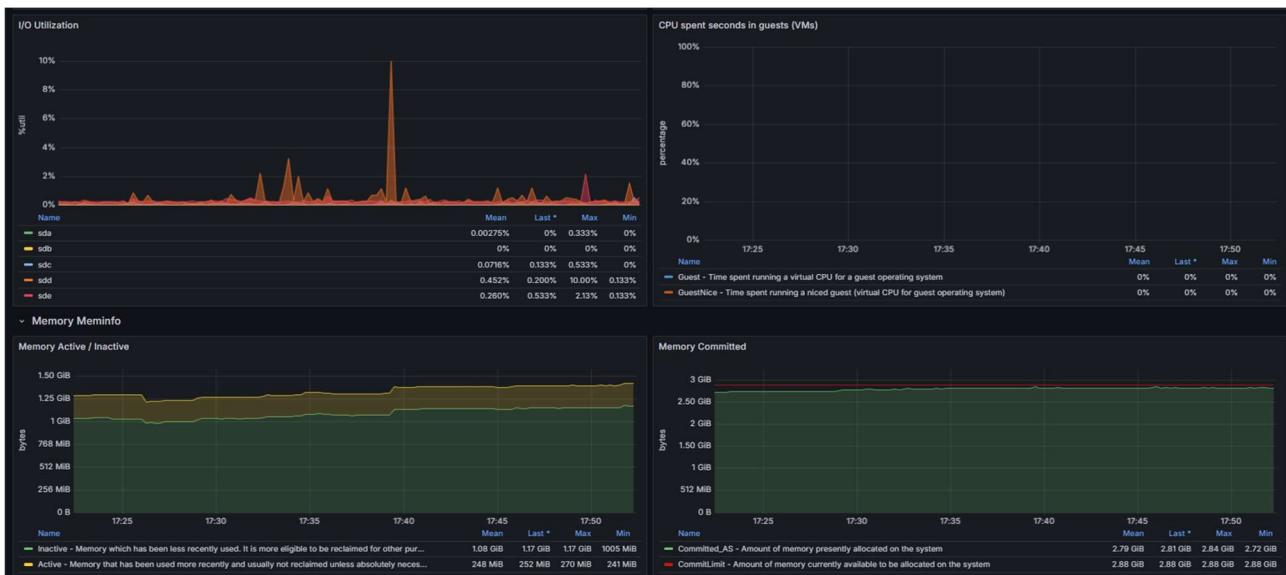


Hình 40 – Dashboard hiển thị chi tiết CPU, Memory Stack, Network Traffic và Disk Space trong Grafana



Hình 41 – Dashboard với Disk IOPs, Disk Space Used và Network Traffic trong Grafana

Bài thực hành 03: Tìm hiểu cơ sở lý thuyết và triển khai thực nghiệm hệ thống giám sát “Prometheus và Grafana”

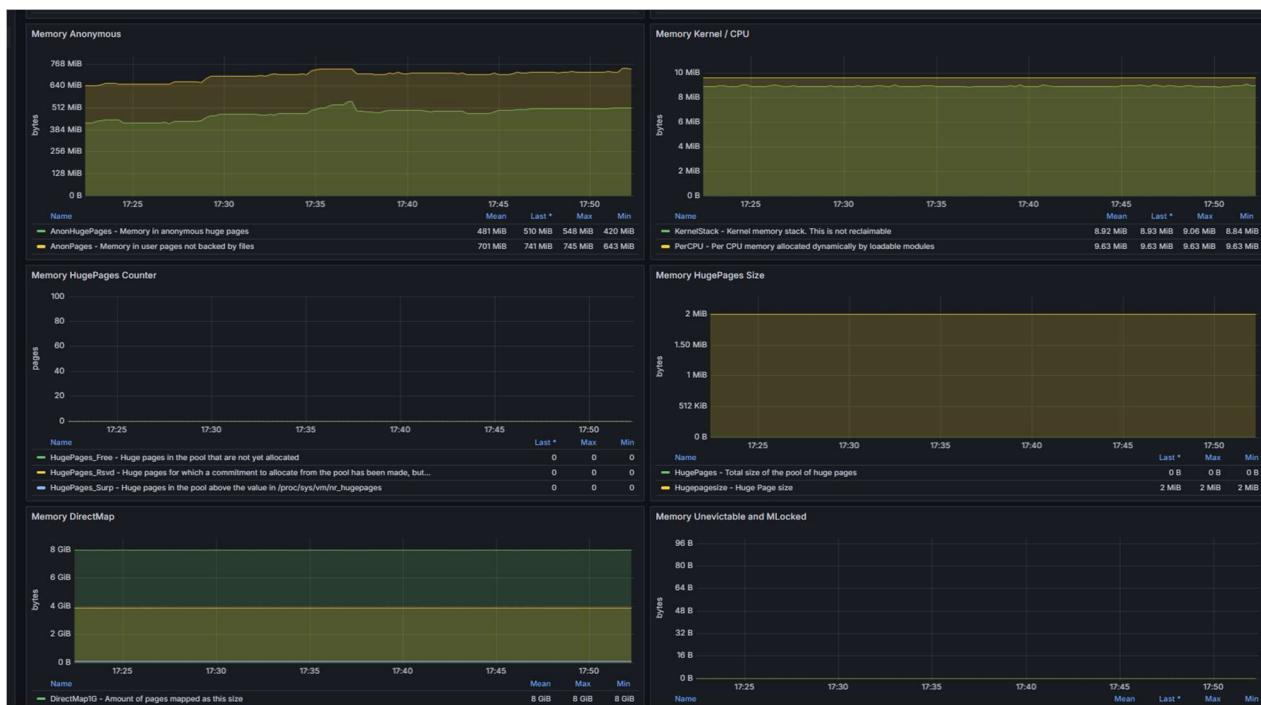


Hình 42 – Dashboard phân tích I/O Utilization, CPU in Guests, và Memory Active/Inactive trong Grafana



Hình 43 – Dashboard chi tiết Memory Active/Inactive, Memory Slab, và Memory Writeback trong Grafana

Bài thực hành 03: Tìm hiểu cơ sở lý thuyết và triển khai thực nghiệm hệ thống giám sát “Prometheus và Grafana”



Hình 44 – Dashboard Memory Anonymous, Kernel/CPU, HugePages, và DirectMap trong Grafana



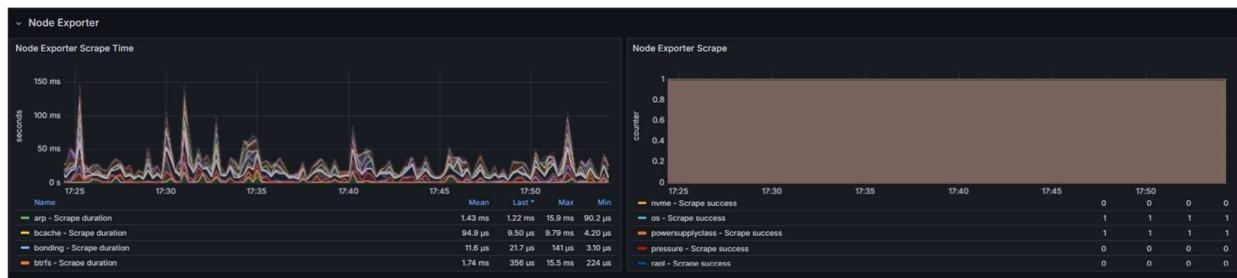
Hình 45 – Dashboard phân tích Disk IOPs, Disk Average Wait Time, và Disk Space R/W trong Grafana

Bài thực hành 03: Tìm hiểu cơ sở lý thuyết và triển khai thực nghiệm hệ thống giám sát “Prometheus và Grafana”

30



Hình 46 – Dashboard Network Traffic by Packets trong Grafana



Hình 47 – Dashboard Node Exporter Scrape Time và Scrape Success trong Grafana

TÀI LIỆU THAM KHẢO

1. Pandora FMS. (n.d.). *Why you need a monitoring system*. Retrieved December 27, 2024, from <https://pandorafms.com/blog/why-you-need-a-monitoring-system/>
2. Prometheus. (n.d.). *Overview*. Retrieved December 27, 2024, from <https://prometheus.io/docs/introduction/overview/>
3. Gimenez, M. (n.d.). *Why is Prometheus pull-based?* Retrieved December 27, 2024, from <https://dev.to/mikkergimenez/why-is-prometheus-pull-based-36k1>
4. Shewale, S. (2021, October 6). *Introduction to Grafana: A beginner’s guide to monitoring and visualization*. Medium. Retrieved December 27, 2024, from <https://medium.com/@sayalishewale12/introduction-to-grafana-a-beginners-guide-to-monitoring-and-visualization-fd6c0f79bde7>
5. How-To Geek. (n.d.). *What is Grafana and when should you use it?* Retrieved December 27, 2024, from <https://www.howtogeek.com/devops/what-is-grafana-and-when-should-you-use-it/>
6. Fosstechnix. (2020, April 5). *Prometheus Node Exporter with example*. Retrieved December 28, 2024, from <https://www.fosstechnix.com/prometheus-node-exporter-with-example/>
7. OpsRamp. (n.d.). *Prometheus Node Exporter*. Retrieved December 28, 2024, from <https://www.opsramp.com/guides/prometheus-monitoring/prometheus-node-exporter/>

HẾT./.