# **TÌM HIỂU VÀ TRIỂN KHAI PROMETHEUS TRÊN LINUX**

1. **Danh sách thành viên & Công việc**

|  |  |  |  |
| --- | --- | --- | --- |
| Họ & tên | MSSV | Công việc | Tiến độ |
| Nguyễn Thị Trang | 175A071236 | - Cài đặt demo Prometheus.  - Hướng dẫn sử dụng | Đang tìm hiểu |
| Trần Thu Thủy | 175A071198 | - Tìm hiểu tài tiệu về Prometheus  - Cài đặt demo Prometheus. | Đang tìm hiểu |

1. **Nội dung nghiên cứu** 
   1. **Prometheus là gì**

- **Prometheus** là một hệ thống giám sát nguồn mở được phát triển bởi SoundCloud (Soundcloud là dịch vụ đăng ký dựa trên web để nghe tải lên nhạc gốc).

- **Prometheus** có khả năng thu thập thông số/số liệu (**metric**) từ các mục tiêu được kết quả và có thể kích hoạt cảnh báo nếu một số điều kiện được thảo mãn yêu cầu.

**b. Một số tính năng của Prometheus**

* Mô hình dữ liệu đa chiều – time series được xác định bởi tên của số liệu (metric) và các cặp khóa – giá trị (key/value).
* Ngôn ngữ truy vấn linh hoạt.
* Hỗ trợ nhiều chế độ biểu đồ.
* Nhiều chương trình tích hợp và hỗ trợ bởi bên thứ 3.
* Hoạt động cảnh báo vấn đề linh động dễ cấu hình.
* Chỉ cần 1 máy chủ là có thể hoạt động được.
* Hỗ trợ Push các time series thông qua một gateway trung gian.
* Các máy chủ/thiết bị giám sát có thể được phát hiện thông qua service discovery hoặc cấu hình tĩnh.
  1. **Một số đặc điểm lưu ý về Prometheus**
* **Prometheus** là 100% mã nguồn mở. Bạn có thể coi mã nguồn tại Git : <https://github.com/prometheus/prometheus/>
* Phần lớn các core tính năng của Prometheus được viết bằng ngôn ngữ Go. Một số còn lại thì được viết bằng Java, Python hoặc Ruby.
* **Prometheus** không dùng để lấy dữ liệu log, thay vì vậy thì Prometheus là dịch vụ giám sát, thu thập và xử lý dữ liệu dạng metric (thông số).
* **Prometheus** sử dụng cơ chế đi lấy (pull) dữ liệu từ máy chủ remote là chính, chứ không sử dụng cơ chế đợi remote đẩy (push) dữ liệu lên ngoại trừ trường hợp sử dụng PushGateway.
* **Prometheus** sử dụng chương trình cảnh báo Alertmanager để xử lý và gửi cảnh báo đi.
* Về phần giao diện biểu đồ (đồ thị) thì Prometheus sử dụng mã nguồn Grafana để tích hợp hiển thị.
* Metric của Prometheus sử dụng chuẩn [OpenMetrics](https://github.com/RichiH/OpenMetrics).
* **Prometheus** hỗ trợ 3 hình thức cài đặt các thành phần hệ thống gồm : Docker Image, cài đặt từ source với Go và file chương trình chạy sẵn đã được biên dịch sẵn.
  1. **Chức năng của prometheus**

- Prometheus lưu trữ tất cả dữ liệu của nó trong một cơ sở dữ liệu chuỗi thời gian, nó cung cấp một mô hình dữ liệu đa chiều và một ngôn ngữ truy vấn mạnh mẽ, cho phép quản trị hệ thống không chỉ dễ dàng tinh chỉnh các định nghĩa về số liệu của chúng mà còn tạo ra các báo cáo chính xác hơn.

- Người ta cho rằng người dùng không tin cậy có quyền truy cập vào điểm cuối và nhật ký HTTP Prometheus. Họ có quyền truy cập vào tất cả các thông tin chuỗi thời gian có trong cơ sở dữ liệu, cộng với nhiều thông tin hoạt động / gỡ lỗi. Người ta cũng cho rằng chỉ những người dùng đáng tin cậy mới có khả năng thay đổi dòng lệnh, tệp cấu hình, tệp quy tắc và các khía cạnh khác của môi trường thời gian chạy của Prometheus và các thành phần khác. Mục tiêu nào sẽ loại bỏ Prometheus, tần suất và với các cài đặt khác được xác định hoàn toàn thông qua tệp cấu hình. Quản trị viên có thể quyết định sử dụng thông tin từ các hệ thống khám phá dịch vụ, kết hợp với việc tái định cư có thể cấp một số điều khiển này cho bất kỳ ai có thể sửa đổi dữ liệu trong hệ thống khám phá dịch vụ đó.

* 1. **Hoạt động của Prometheus**

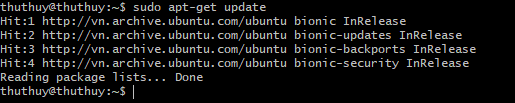
- Prometheus có khả năng thu thập thông số/số liệu (metric) từ các mục tiêu được cấu hình theo các khoảng thời gian nhất định, đánh giá các biểu thức quy tắc, hiển thị kết quả và có thể kích hoạt cảnh báo nếu một số điều kiện được thảo mãn yêu cầu.

* Thiết lập máy chủ ảo Apache2 trên UbuntuServer 18.04:

<https://www.digitalocean.com/community/tutorials/how-to-set-up-apache-virtual-hosts-on-ubuntu-16-04?fbclid=IwAR1giMUGXgWi_HrV-DfFlAbJfzsexEsHSf1SP0fu7UN9RMqLkybUwaMJ9qA>

+ Bước 1: Cài đặt Apache trên máy chủ của mình thông qua apt-get

|  |
| --- |
| * + - * sudo apt-get update       * sudo apt-get install apache2 |



+ Bước 2: Tạo cấu trúc thư mục

|  |
| --- |
| * + - * sudo mkdir -p /var/www/**trangthuy.linux**/public\_html |

+ Bước 3: Cấp quyền

|  |
| --- |
| * + - * sudo chown -R $USER:$USER /var/www/ **trangthuy.linux**/public\_html       * sudo chmod -R 755 /var/www |

+ Bước 4: Tạo trang demo cho mỗi máy chủ ảo

|  |
| --- |
| * sudo nano /var/www/ **trangthuy.linux**/public\_html/index.html |

+ Bước 5: Tạo tập tin máy chủ ảo mới

* Tạo tập tin máy chủ ảo đầu tiên:

|  |
| --- |
| * + - * sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/**trangthuy.linux**.conf |

* Mở tệp mới trong trình chỉnh sửa của bạn với quyền root:

|  |
| --- |
| * + - * sudo nano /etc/apache2/sites-available/ **trangthuy.linux**.conf |

* + - * ServerAdmin **trangthuylinux@gmail.com**
      * ServerName **trangthuy.linux**
      * ServerAlias **192.168.0.120** địa chỉ ip
      * DocumentRoot /var/www/ **trangthuy.linux/**public\_html

+ Bước 6: Kích hoạt tệp máy chủ ảo mới

|  |
| --- |
| * + - * sudo a2ensite **trangthuy.linux**.conf       * sudo a2dissite 000-default.conf       * sudo systemctl restart apache2 |

+ Bước 7: Thiết lập máy chủ cục bộ

|  |
| --- |
| * + - * sudo nano /etc/hosts       * 192.168.0.120 **trangthuy.linux** :thêm địa chỉ ip của máy. |

+ Bước 8: Kiểm tra kết quả:

[**http://trangthuy.linux**](http://trangthuy.linux)

* 1. **Hướng dẫn Sử dụng/Quản trị Prometheus**

Bước 1 - Cài đặt Prometheus Server

## Bước 2 - Cài đặt Node Exporter (Để theo dõi các số liệu của máy chủ)

## Bước 3 - Chạy Node Exporter dưới dạng dịch vụ

## Bước 4 - Khởi động máy chủ Prometheus

## Bước 5 - Cài đặt PromDash

## Bước 6 - Thiết lập môi trường Rails

## Bước 7 - Khởi động và cấu hình PromDash

**I. Hướng dẫn cài đặt Prometheus trên Ubuntu**

[1] Cài đặt gói [prometheus] bao gồm chức năng Prometheus Server và gói [prometheus-node-export] bao gồm chức năng để lấy tài nguyên chung trên Hệ thống như sử dụng CPU hoặc Bộ nhớ.

|  |
| --- |
| * sudo apt -y install prometheus prometheus-node-exporter |

[2] Cấu hình cài đặt cơ bản trên file prometheus.yml

|  |
| --- |
| * sudo vi /etc/prometheus/prometheus.yml |

Sau đó khởi động lại Prometheus

|  |
| --- |
| * sudo systemctl restart prometheus |

**global:**

**scrape\_interval: 15s # By default, scrape targets every 15 seconds.**

**evaluation\_interval: 15s # By default, scrape targets every 15 seconds.**

**# scrape\_timeout is set to the global default (10s).**

**# Attach these labels to any time series or alerts when communicating with**

**# external systems (federation, remote storage, Alertmanager).**

**external\_labels:**

**monitor: 'example'**

**# Load and evaluate rules in this file every 'evaluation\_interval' seconds.**

**rule\_files:**

**# - "first.rules"**

**# - "second.rules"**

**# A scrape configuration containing exactly one endpoint to scrape:**

**# Here it's Prometheus itself.**

**scrape\_configs:**

**# The job name is added as a label `job=<job\_name>` to any timeseries scraped from this config.**

**- job\_name: 'prometheus'**

**# Override the global default and scrape targets from this job every 5 seconds.**

**scrape\_interval: 5s**

**scrape\_timeout: 5s**

**# metrics\_path defaults to '/metrics'**

**# scheme defaults to 'http'.**

**static\_configs:**

**- targets: ['localhost:9090']**

**- job\_name: node**

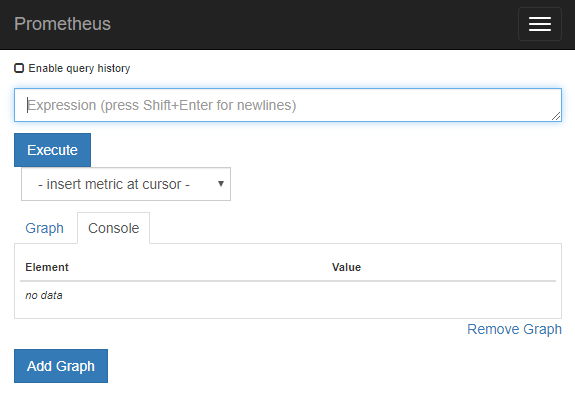
**# If prometheus-node-exporter is installed, grab stats about the local**

**# machine by default.**

**static\_configs:**

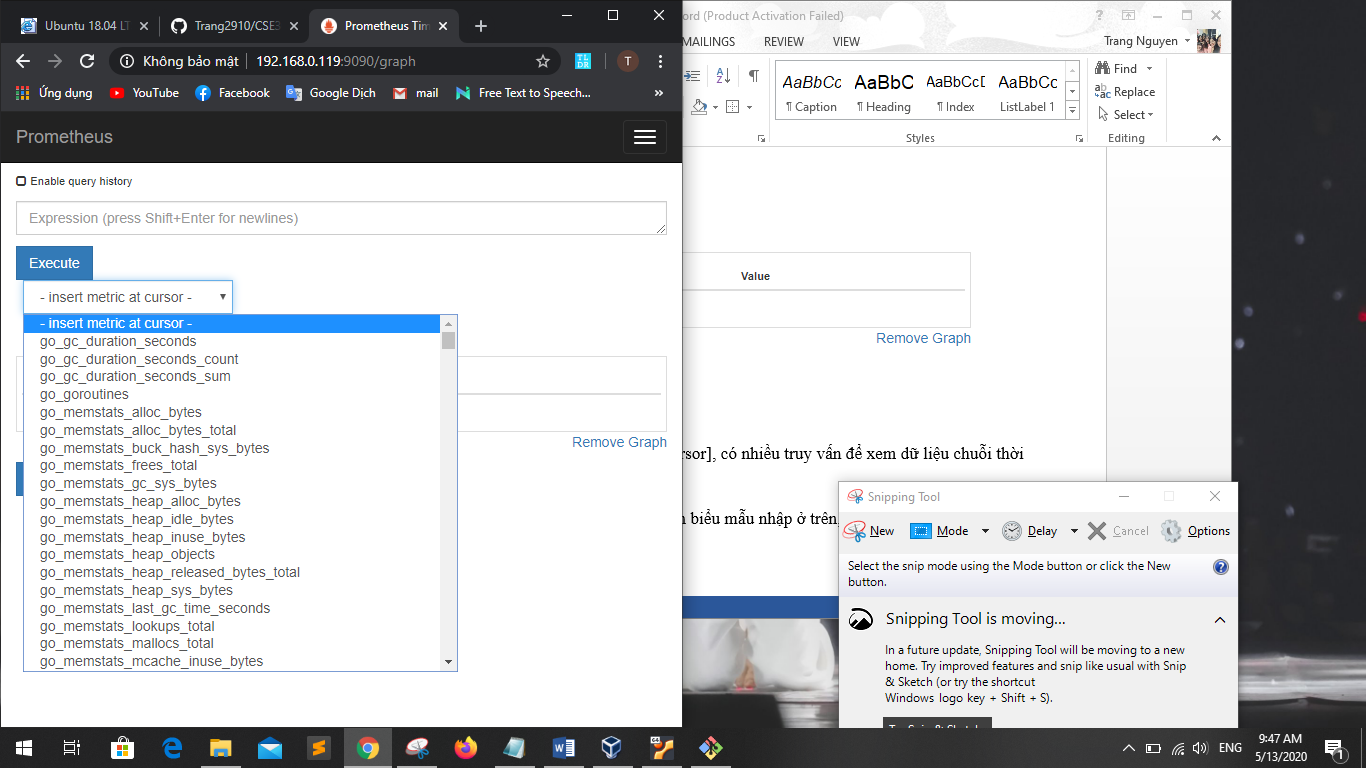
**- targets: ['localhost:9100']**

[3] Truy cập vào [<http://trangthuylinux.com:9090/>] từ máy chủ khách



[4] Để nhấp vào [insert metric at cursor], có nhiều truy vấn để xem dữ liệu chuỗi thời gian.

Cũng có thể nhập truy vấn trực tiếp trên biểu mẫu nhập ở trên, bằng Ngôn ngữ biểu thức Prometheus.

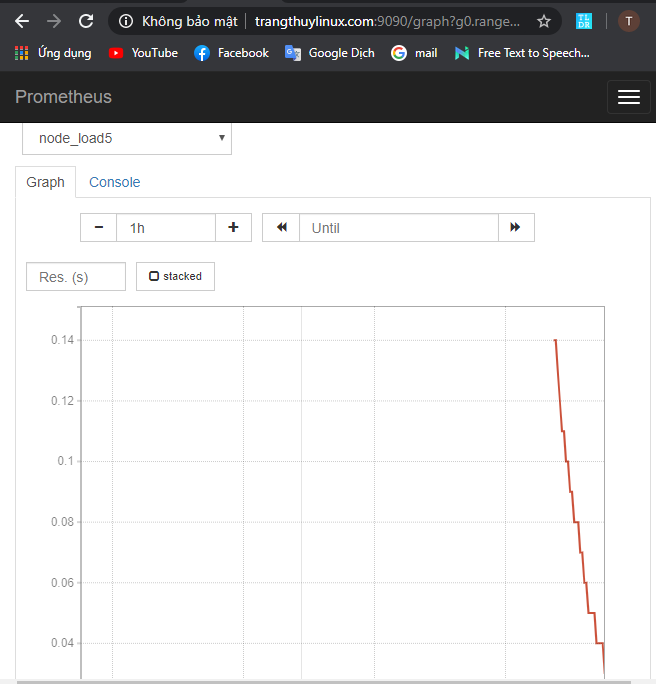


[5] Đây là trạng thái thực thi [node\_load5].

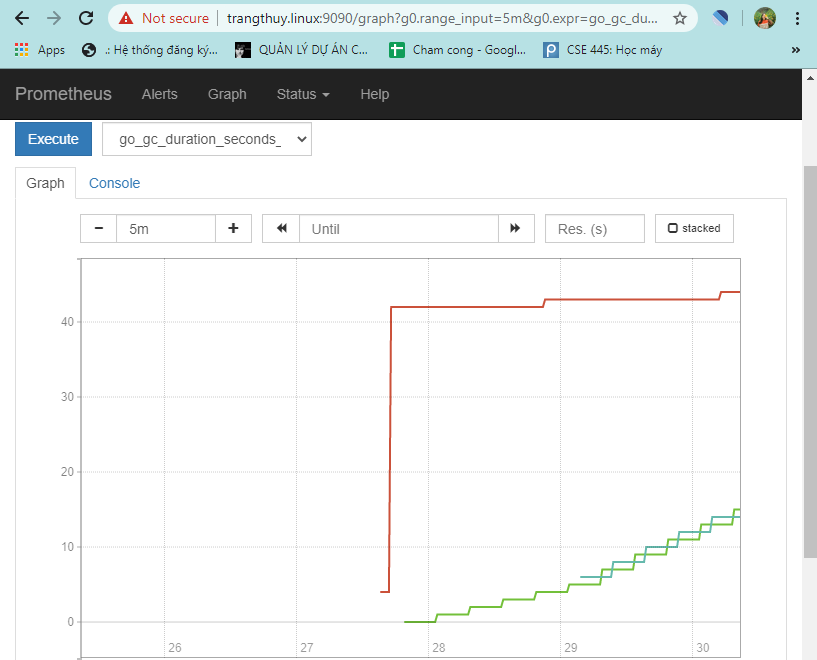
Prometheus sẽ chủ động pull (kéo) các metrics về qua HTTP mỗi 10s hay 30s do chúng ta thiết lập. Bản thân các service thì thường không thể tự export được các metrics cho Prometheus mà cần đến các Instructmentation/Exporter

Khám phá qua trang chủ, bạn sẽ thấy một Input là Expression, là chỗ để mình nhập các câu query PromQL - là một ngôn ngữ truy vấn dữ liệu được Prometheus tự tạo ra. Với ưu điểm là câu truy vấn siêu ngắn gọn và dễ hiểu.  Sau khi execute một query thì result sẽ hiển thị bên dưới ở tab Console. Chúng ta có thể nhấn sang tab Graph để xem biểu đồ mà Prometheus vẽ sẵn cho chúng ta. Cái Graph này nó thực sự quá simple với nhu cầu thực tế. Vì vậy nên mọi người hay kết hợp với Grafana để visualize kết quả.

* Chọn tên một metric node\_load5 mà prometheus đã scrape được vào Expression rồi chạy thử để lấy dữ liệu của metric này trong 1h vừa qua (m = minutes, h = hours, d = days):

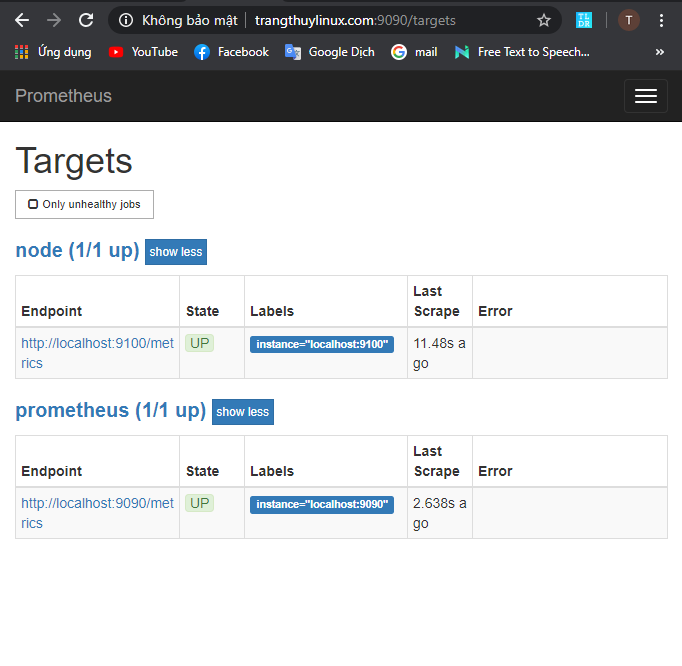


* Chọn tên một metric go\_gc\_duration\_seconds\_count mà prometheus đã scrape được vào Expression rồi chạy thử để lấy dữ liệu của metric này trong 5m vừa qua (m = minutes, h = hours, d = days):



|  |
| --- |
| \* Giải thích một số metric:  - go\_gc\_duration\_seconds: Tóm tắt về thời lượng gọi của GC  - go\_goroutines: Số lượng goroutines hiện đang tồn tại  - go\_memstats\_alloc\_bytes: Số lượng byte được phân bổ và vẫn đang sử dụng  - go\_memstats\_alloc\_bytes\_total: Tổng số byte được phân bổ, ngay cả khi được giải phóng  - go\_memstats\_buck\_hash\_sys\_bytes: Số lượng byte được sử dụng bởi bảng băm cấu hình  - go\_memstats\_gc\_sys\_bytes: Số byte được sử dụng cho siêu dữ liệu của hệ thống thu gom  - go\_memstats\_heap\_alloc\_bytes: Số lượng byte heap được phân bổ và vẫn đang được sử dụng  - go\_memstats\_heap\_idle\_bytes: Số lượng byte heap đang chờ sử dụng  - go\_memstats\_heap\_objects: Số lượng đối tượng được phân bổ  - go\_memstats\_heap\_release\_bytes\_total: Tổng số byte heap được phát hành cho HĐH  - http\_Vquest\_duration\_microseconds: Độ trễ yêu cầu HTTP tính bằng micrô giây  - http\_request\_size\_bytes: Các kích thước yêu cầu HTTP bằng byte  - http\_response\_size\_bytes: Kích thước phản hồi HTTP theo byte  - node\_disk\_bytes\_read: Tổng số byte được đọc thành công  - node\_load5: trung bình tải 5m |

[6] Để nhấp vào [Status] - [Targets], có thể xác nhận trạng thái cho từng nút.



**II. Cài đặt Node Exporter (Để theo dõi các số liệu của máy chủ)**

Prometheus Node Exporter là một chương trình exporter viết bằng ngôn ngữ Golang. Exporter là một chương trình được sử dụng với mục đích thu thập, chuyển đổi các metric không ở dạng kiểu dữ liệu chuẩn Prometheus sang chuẩn dữ liệu Prometheus. Sau đấy exporter sẽ expose web service api chứa thông tin các metrics hoặc đẩy về Prometheus.

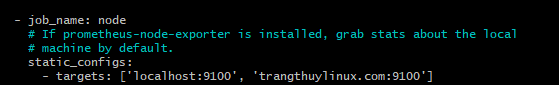
Node Exporter này sẽ đi thu thập các thông số về máy chủ Linux như : ram, load, cpu, disk, network,…. từ đó tổng hợp và xuất ra kênh truy cập các metrics hệ thống này ở port TCP 9100 để Prometheus đi lấy dữ liệu metric cho việc giám sát.

[1] Cài đặt gói [prometheus-node-export] bao gồm chức năng để lấy tài nguyên chung trên Hệ thống như CPU ​​hoặc sử dụng Bộ nhớ trên Nút mà bạn muốn thêm.

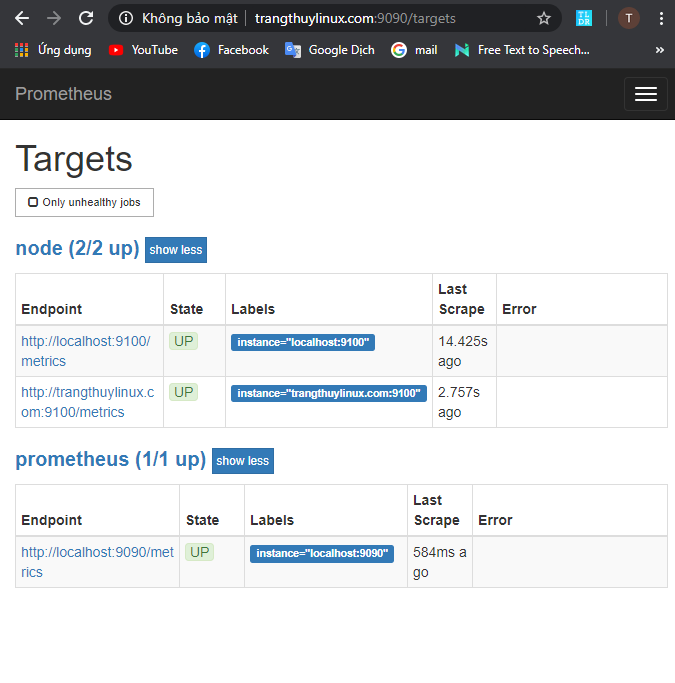
|  |
| --- |
| * sudo apt -y install prometheus-node-exporter * sudo systemctl status prometheus-node-exporter |

[2] Thêm cài đặt trên Prometheus Server Config.

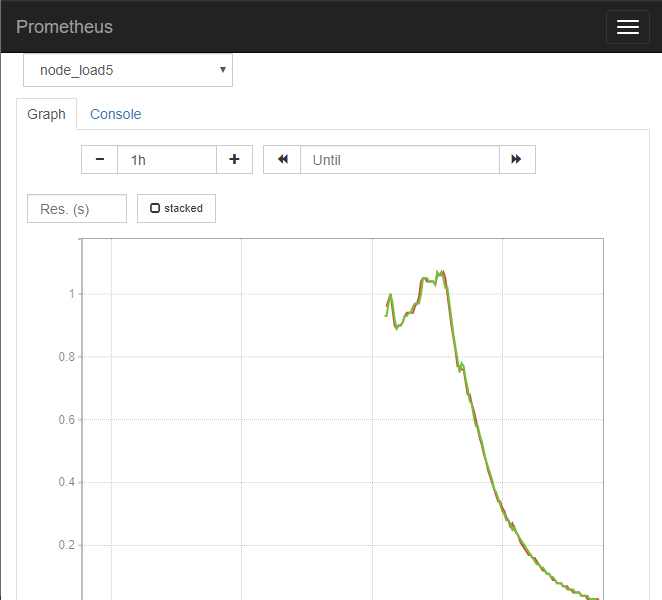
|  |
| --- |
| * sudo nano /etc/prometheus/prometheus.yml |



[3] Truy cập vào UI Web Prometheus và nhấp vào [Status] - [Targets] để xác minh các nút mới được liệt kê.



Chọn tên một metric node\_load5 mà prometheus đã scrape được vào Expression rồi chạy thử để lấy dữ liệu của metric này trong 1h vừa qua (m = minutes, h = hours, d = days):



**III. Cài đặt cảnh báo trên Prometheus**

Bất kỳ người dùng nào có quyền truy cập vào điểm cuối HTTP Alertmanager đều có quyền truy cập vào dữ liệu của nó. Họ có thể tạo và giải quyết các cảnh báo. Họ có thể tạo, sửa đổi và xóa các khoảng lặng.

Nơi thông báo được gửi đến được xác định bởi tệp cấu hình. Với một số thiết lập khuôn mẫu nhất định, có thể thông báo kết thúc tại một điểm đến được xác định cảnh báo. Ví dụ: nếu thông báo sử dụng nhãn cảnh báo làm địa chỉ email đích, bất kỳ ai có thể gửi thông báo đến Alertmanager đều có thể gửi thông báo đến bất kỳ địa chỉ email nào. Nếu đích được xác định cảnh báo là trường bí mật có thể đặt được, bất kỳ ai có quyền truy cập vào Prometheus hoặc Alertmanager đều có thể xem các bí mật. Bất kỳ trường bí mật nào có thể tạo khuôn mẫu đều được dùng để định tuyến thông báo trong trường hợp sử dụng ở trên. Chúng không nhằm mục đích tách biệt các tệp bí mật khỏi các tệp cấu hình bằng tính năng tệp mẫu. Bất kỳ bí mật nào được lưu trữ trong các tệp mẫu có thể được thực hiện bởi bất kỳ ai có thể định cấu hình máy thu trong tệp cấu hình Alertmanager. Ví dụ: trong các thiết lập lớn, mỗi nhóm có thể có một đoạn tệp cấu hình cảnh báo mà họ kiểm soát hoàn toàn, sau đó được kết hợp thành tệp cấu hình cuối cùng đầy đủ.

Hoạt động cảnh báo trong hệ thống Prometheus chia làm 2 phần :

+ Các rule cảnh báo được thiết lập ở Prometheus Server và gửi cảnh báo đó đến Alertmanager.

+ Alertmanager sẽ quản lý các cảnh báo (alert), xử lý nội dung alert nếu có tuỳ biến này và điều hướng đầu tiếp nhận thông tin cảnh báo như email, chat platform, call,…

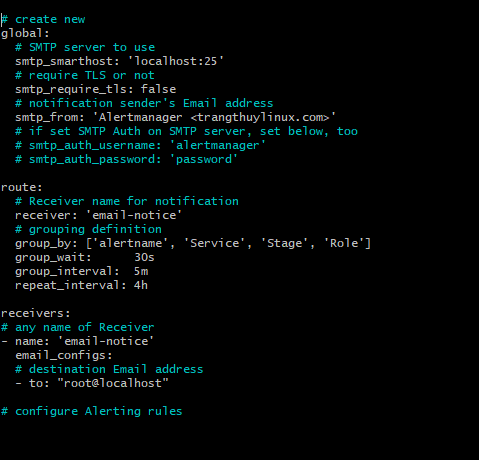
Ở đây ta sử dụng cấu hình cảnh báo với trình nhận email.

[1] Cài đặt Alertmanager trên Máy chủ Prometheus.

|  |
| --- |
| * sudo apt -y install prometheus-alertmanager |

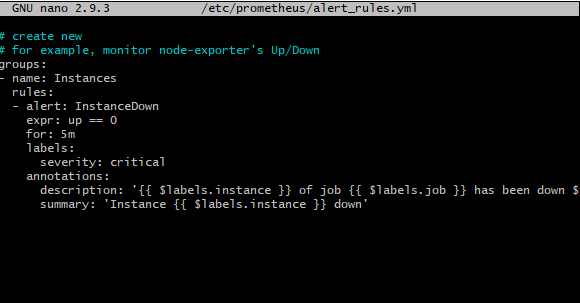
[2] Cấu hình cài đặt cảnh báo Prometheus.

|  |
| --- |
| * sudo mv /etc/prometheus/alertmanager.yml /etc/prometheus/alertmanager.yml.org * sudo nano /etc/prometheus/alertmanager.yml |

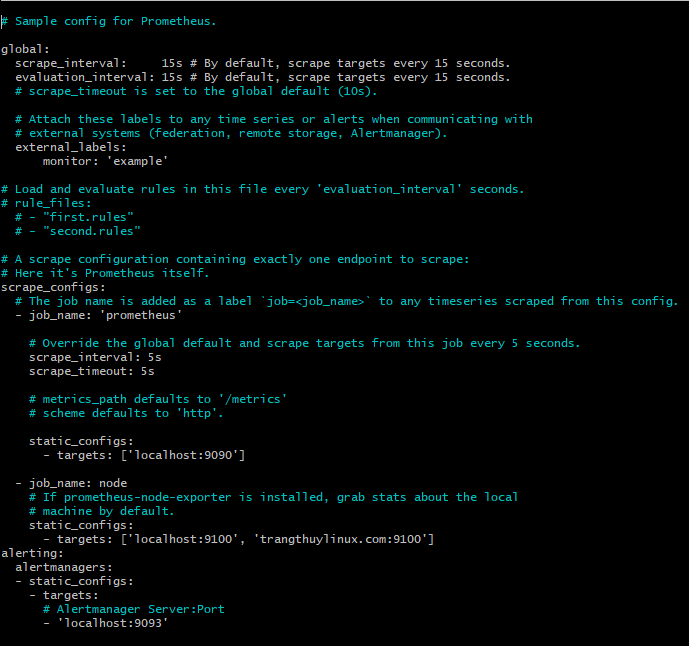


Cấu hình quy tắc cảnh báo

|  |
| --- |
| * sudo nano /etc/prometheus/alert\_rules.yml |



|  |
| --- |
| * sudo nano /etc/prometheus/prometheus.yml |



[3] Truy cập <http://trangthuylinux.com:9093/>

