# What is Monitoring

monitoring là cách ta sử dụng các công cụ để giám sát hệ thống và ứng dụng của ta. Monitoring sẽ cung cấp cho ta các giá trị về hiệu suất, trình trạng của hệ thống và ứng dụng, để ta có thể nhanh chóng phát hiện các vấn đề xảy ra với hệ thống.

Nhưng monitoring không dừng lại ở việc chỉ giám sát các giá trị về hệ thống như CPU hoặc memory, mà còn bao gồm các giá trị có thể giúp ích về mặt kinh doanh như số lượng khách hàng truy cập trang web, thời gian khách hàng ở lại trang web, số lượng khách hàng bấm vào một sản phẩm nào đó, ...

Nên đối với monitoring ta sẽ có hai đối tượng là:

* Hệ thống (the system).
* Kinh doanh (the business).

## System Monitoring

Infrastructure monitoring: CPU, Memory, disk

Application monitoring: monitor service, thời gian phản hồi của ứng dụng, tình trạng của ứng dụng (chạy hay chết), CPU và memory ứng dụng, số lượng yêu cầu (request) vào ứng dụng trong 1 giây, ...

Network monitoring: tầng mạng - các giá trị như số lượng *byte* ra và vào, tình trạng của firewall, ...

## Business

Để monitoring được tương tác của người dùng thì ta cần phải thực hiện ở tầng code front-end hoặc backend, ví dụ ở tầng front-end để theo dõi được hành vi của người dùng ta sẽ dùng Google Analytics.

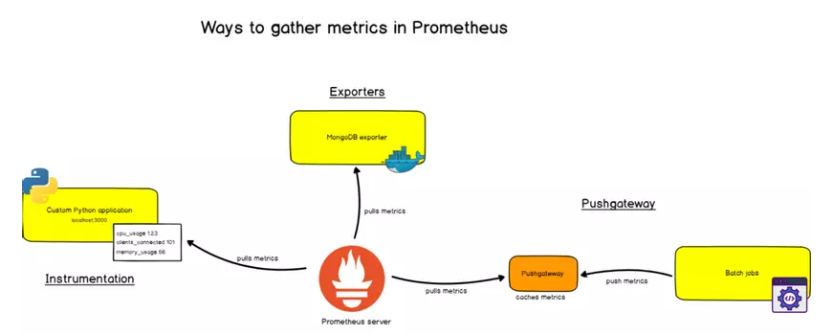
# Prometheus

Dữ liệu mà nó lưu trữ có thể là các metrics (thông số) về tình trạng server như lượng RAM, CPU

Prometheus sẽ chủ động pull (kéo) các metrics về qua HTTP. Bản thân các service thì thường không thể tự export được các metrics cho Prometheus mà cần đến các Instructmentation/Exporter

Install exporter để monitor các service

* Exporter là những app được viết cho mấy cái thông dụng như Database, Server. Chúng ta chỉ cần chạy nó và nó sẽ export các metrics thu thập được cho mình.



Một số exporter

* Node Exporter: export các metrics một con node (hiểu là một server) như CPU, RAM của node, dung lượng ổ đĩa, số lượng request tới node đấy, .etc.
* Postgres Exporter, giúp đọc dữ liệu từ các bảng trong Postgres và export ra cho Prometheus
* HAProxy Exporter

# Prometheus – grafana

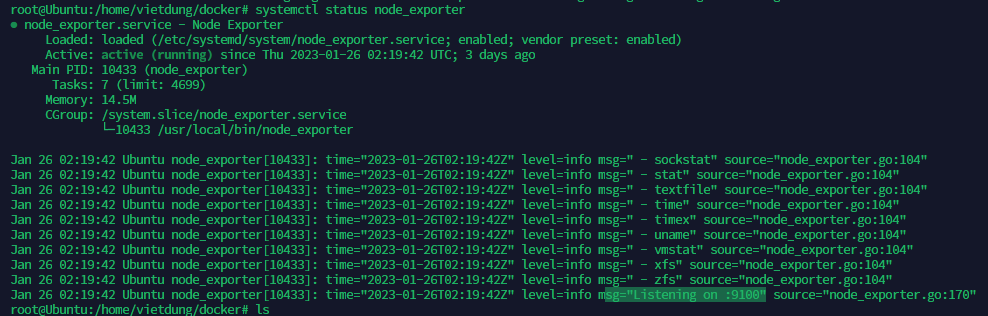
Grafana cung cấp giao diện để monitor,

Grafana kết hợp các plugin để monitor, bao gồm cả prometheus

Prometheus là công cụ để monitor metric của server

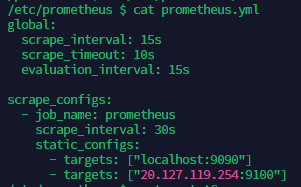
Prometheus cũng là 1 builtin exporter để monitor server port 9090

Nhưng n ko monitor được 1 số thông số của server, ta cần node\_exporter port 9100



Listen port 9100

* Ta có file config của prometheus



* Prometheus monitor cả nó và node\_expoter

Vì vậy grafana chỉ cần add prometheus vào là sẽ lấy được cả log của node\_exporter

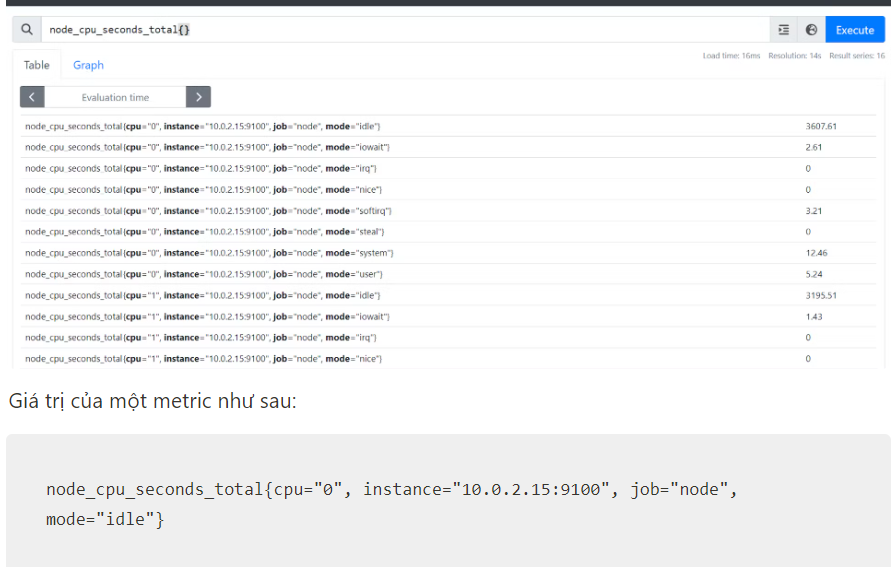
Logs đẩy về prometheus, prometheus đẩy về grafana

## Tính CPU

## CPU Utilization

CPU Utilization là một chỉ số đo lường tỉ lệ thời gian CPU sử dụng để thực hiện các tác vụ của hệ thống so với tổng thời gian hoạt động của CPU trong một khoảng thời gian nhất định. Nó được tính bằng phần trăm và thường được sử dụng để đánh giá hiệu suất hoạt động của hệ thống. Nếu tỉ lệ này quá cao, có thể dẫn đến tình trạng quá tải CPU và ảnh hưởng đến hoạt động của hệ thống.

Metric mà Node Exporter thu thập về cho CPU Utilization là node\_cpu\_seconds\_total, để kiểm tra ta truy cập địa chỉ http://localhost:9090/graph và gõ vào node\_cpu\_seconds\_total{}.



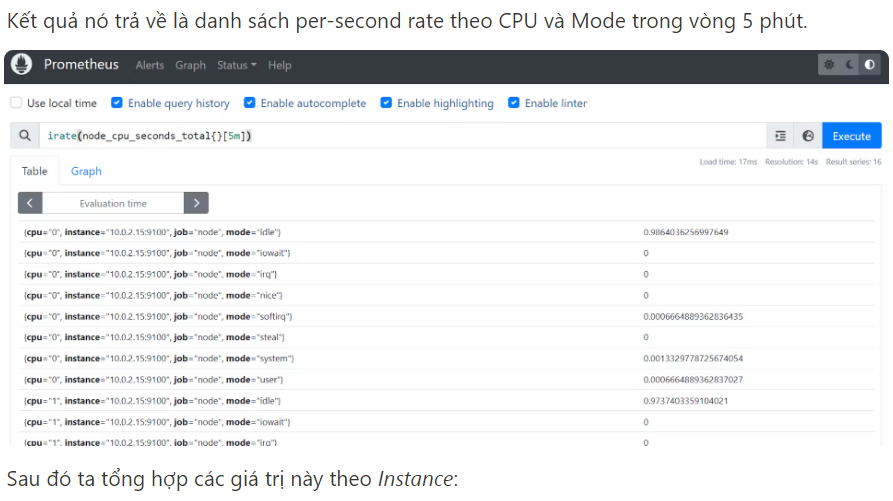
* Hai labels instance và job giúp xác định metric này được thu thập từ đâu
* Label cpu chỉ định thứ tự *core* của CPU trong máy chủ, ví dụ ta hay nghe CPU có 2 core hay 4 core
* Label mode là chế độ của CPU như: user, system, idle, …

Đây là dữ liệu thô và khi nhìn vào ta cũng không hình dung được gì nhiều về tình trạng hiện tại của máy chủ. Ta cần chuyển nó sang các giá trị hữu ích cho người giám sát, ví dụ như phần trăm CPU được sử dụng trong vòng 5 phút, hay còn được gọi là CPU Utilization. Công thức để tính được giá trị CPU Utilization như sau:

100 - avg(irate(node\_cpu\_seconds\_total{mode="idle"}[5m])) by (instance) \* 100

Tiếp theo ta tìm hiểu chi tiết về công thức trên, để tính được CPU Utilization đầu tiên ta cần tính *per-second rate* của CPU. Prometheus PromQL cung cấp hàm irate để tính giá trị trên, ví dụ ta tính tỷ lệ trong vòng 5 phút:

irate(node\_cpu\_seconds\_total{}[5m])



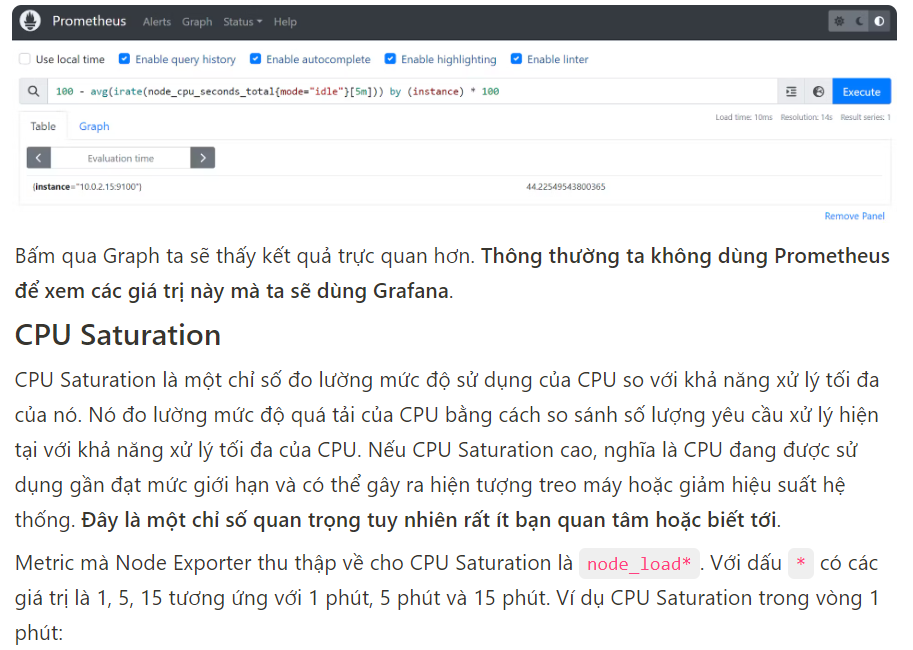


Nên để tính được tài nguyên CPU đang được sử dụng, ta sẽ lấy tổng tài nguyên của CPU (100) trừ đi cho tài nguyên đang ở trạng thái idle. Phần trăm tài nguyên đang ở trạng thái idle:

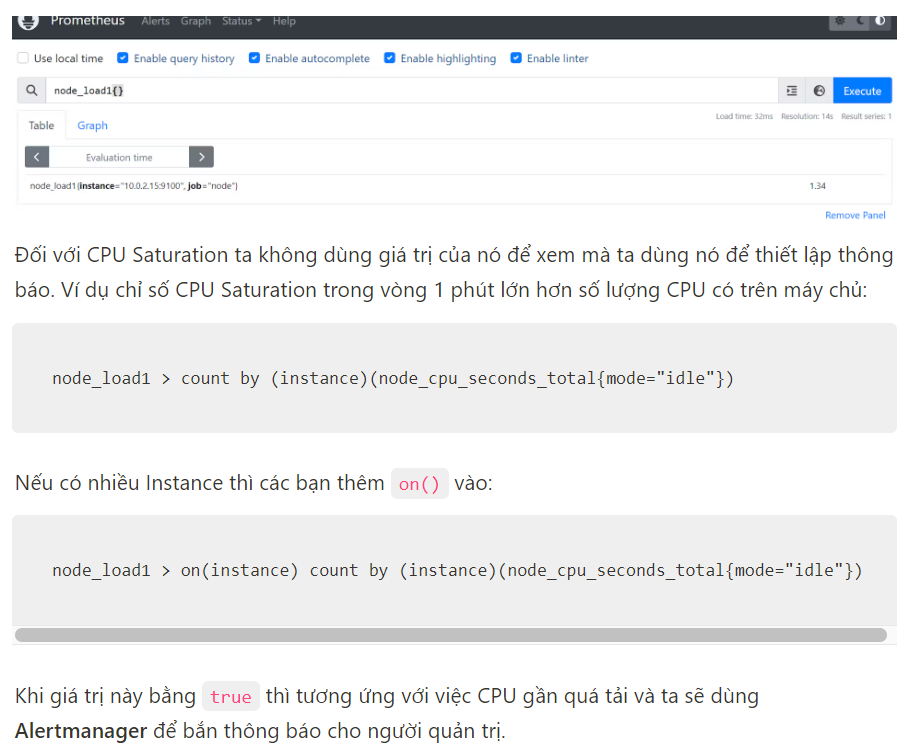
avg(irate(node\_cpu\_seconds\_total{mode="idle"}[5m])) by (instance) \* 100

Phần trăm CPU đang được sử dụng (CPU Utilization):

100 - avg(irate(node\_cpu\_seconds\_total{mode="idle"}[5m])) by (instance) \* 100



node\_load1{}



# Triển khai PMM

Install with docker

Add data source promethues với loki



Cái này chỉ là import những dashboarch có sẵn

Promtail (đẩy log) -> loki -> grafana

Prometheus pull metric từ (node exporter, mongo\_exporter) -> grafana

Prometheus sẽ scrape mấy thằng này, trừ thằng loki

## Monitor mongodb

1. Bắn log mongodb

Cài promtail

Promtail (đẩy log) -> loki -> grafana

PROMTAIL\_VERSION=$(curl -s "https://api.github.com/repos/grafana/loki/releases/latest" | grep -Po '"tag\_name": "v\K[0-9.]+')

sudo mkdir /opt/promtail

sudo wget -qO /opt/promtail/promtail.gz "https://github.com/grafana/loki/releases/download/v${PROMTAIL\_VERSION}/promtail-linux-amd64.zip"

sudo gunzip /opt/promtail/promtail.gz

sudo chmod a+x /opt/promtail/promtail

sudo ln -s /opt/promtail/promtail /usr/local/bin/promtail

sudo wget -qO /opt/promtail/promtail.yaml "https://raw.githubusercontent.com/grafana/loki/v${PROMTAIL\_VERSION}/clients/cmd/promtail/promtail-local-config.yaml"

promtail -version

sudo tee /etc/systemd/system/promtail.service<<EOF

[Unit]

Description=Promtail client for sending logs to Loki

After=network.target

[Service]

ExecStart=/opt/promtail/promtail -config.file=/opt/promtail/promtail.yaml

Restart=always

TimeoutStopSec=3

[Install]

WantedBy=multi-user.target

EOF

sudo tee /opt/promtail/promtail.yaml<<EOF

server:

  http\_listen\_port: 9080

  grpc\_listen\_port: 0

positions:

  filename: /tmp/positions.yaml

clients:

  - url: http://192.168.48.1:3100/loki/api/v1/push

scrape\_configs:

- job\_name: MongoDB\_logs

  static\_configs:

  - targets:

      - localhost

    labels:

      job:  mongodb-1

**\_\_path\_\_**: /var/log/mongodb/\*.log

EOF

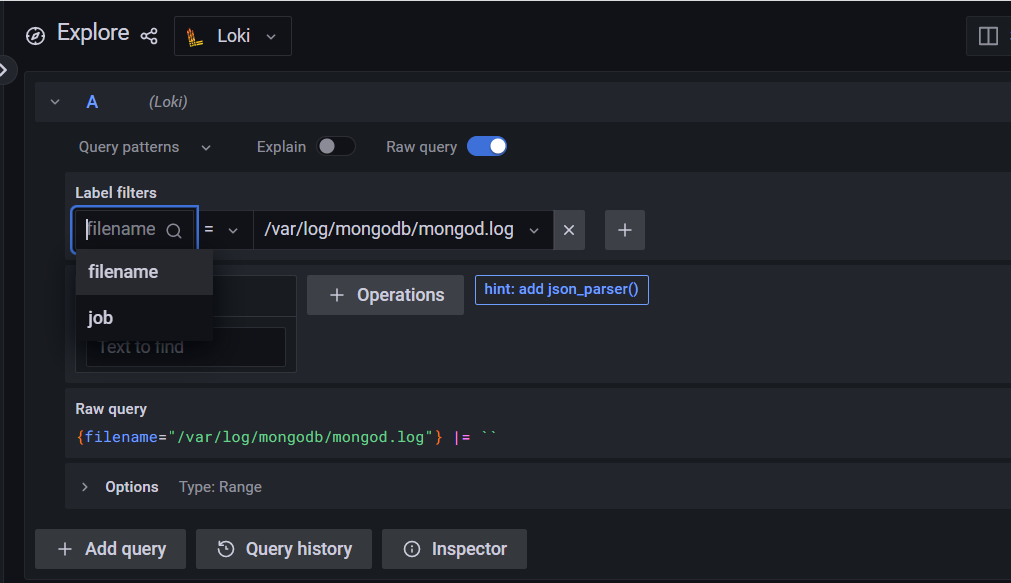
sudo systemctl enable promtail

sudo service promtail restart

sudo systemctl status promtail

đẩy log đến loki

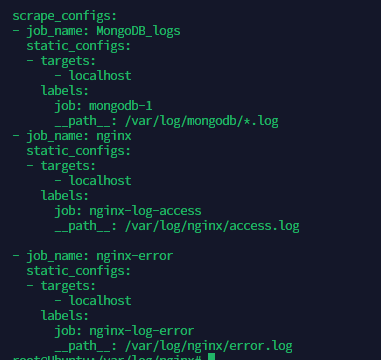
* check log promtail oke chưa
* check loki service oke chưa
* check data source được add chưa

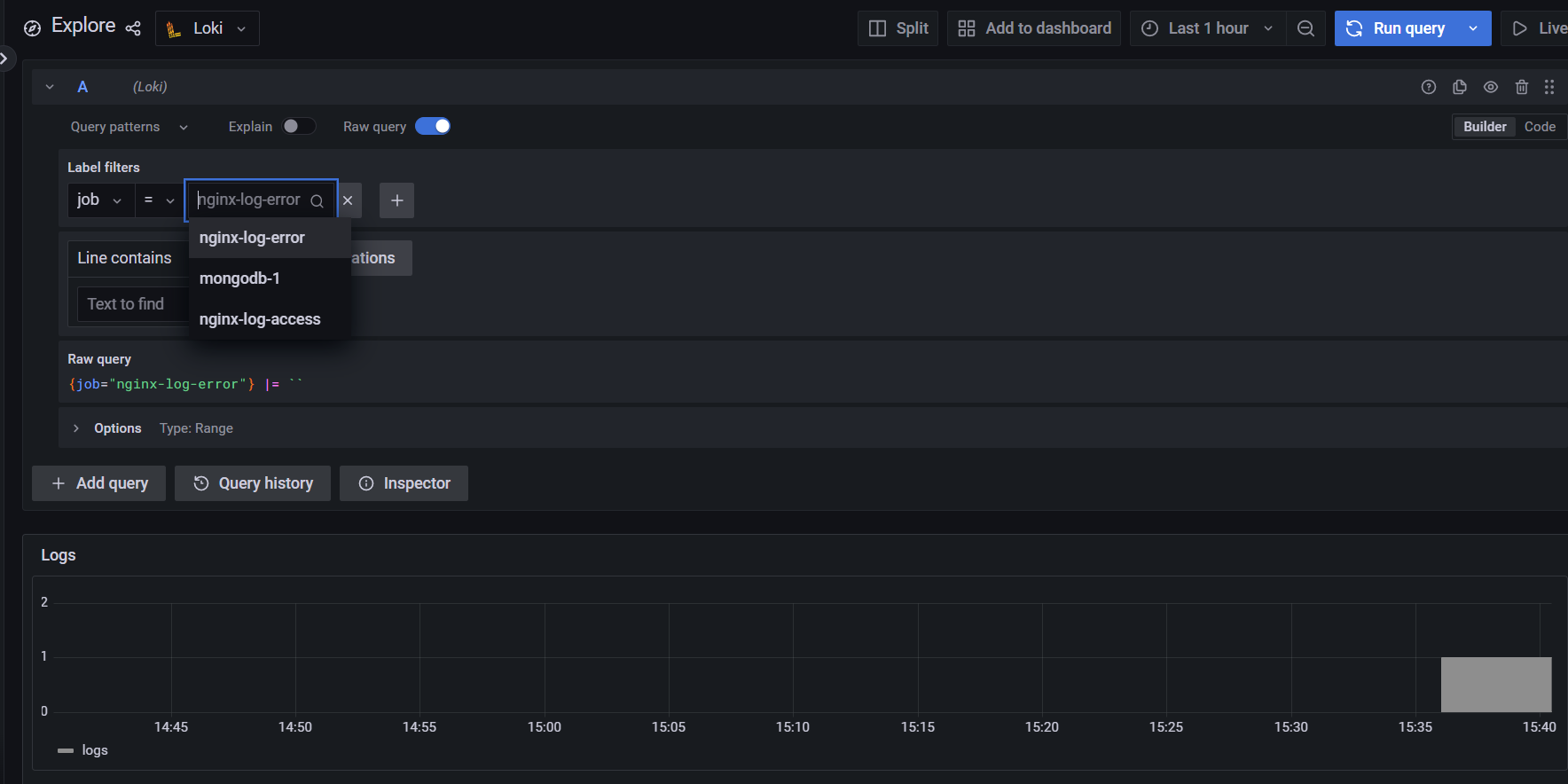


Và oke, 1 là filename

* 2 là cái job (là cái label mình vừa add ở file config promtail)

Ta cũng có thể chia chi tiết cho log error và log access trong promtail





Chia log chi tiết thế này oke hơn.

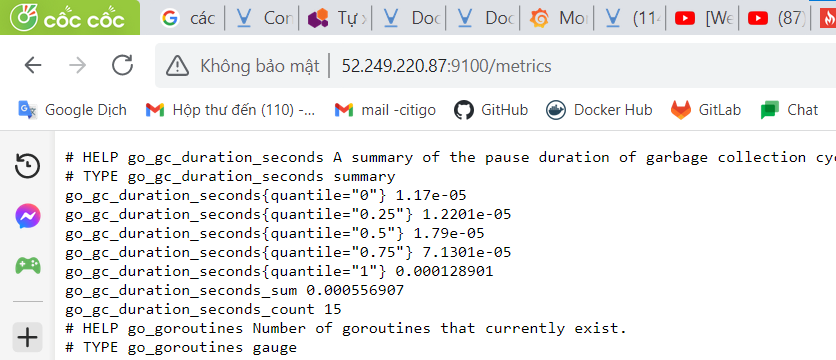
Oke xog phần log

1. Phần dashboarsh metric

* Monitor node

Install node\_exporter

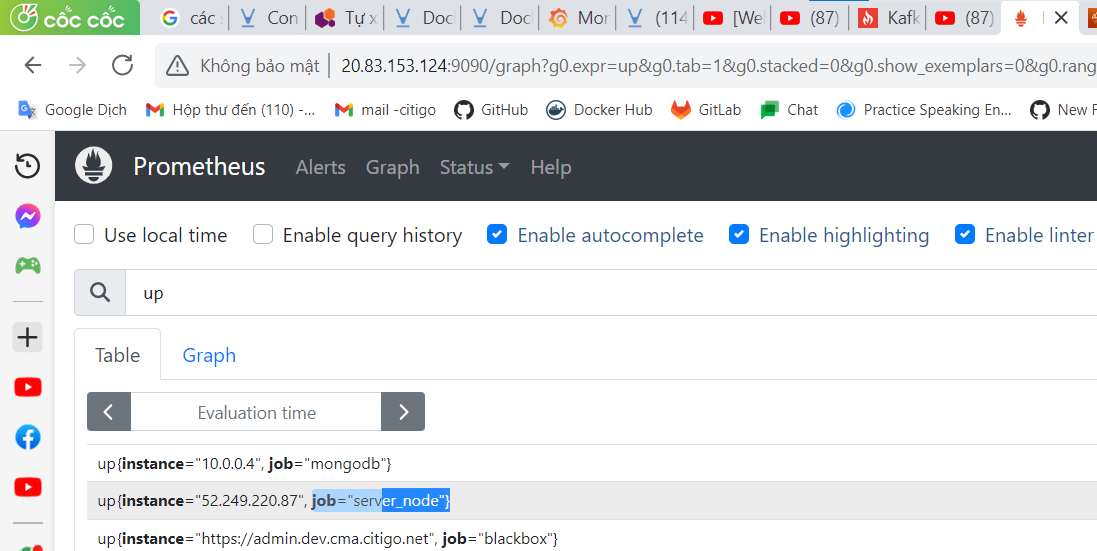
* Check node\_exporter đã hoạt động chưa



Oke node\_exporter đã pull được metric

* Check prometheus đã scrape node exporter được chưa

Vào prometheus check query



Oke n đã scrape được

* Vậy là node\_exporter xog