**Prometheus và Grafana - giải pháp tối ưu cho hệ thống Monitor**

**I. Prometheus**

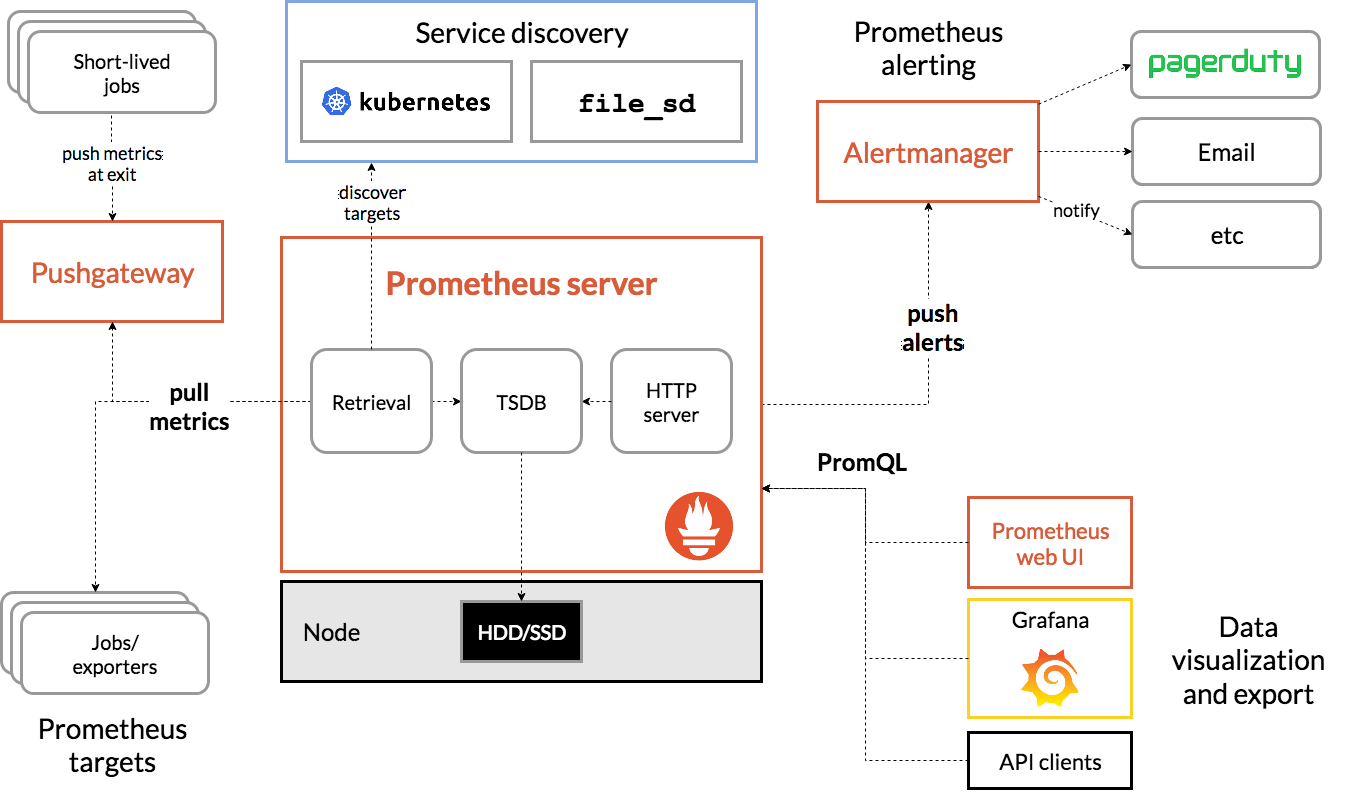
Prometheus là một open-source systems monitoring và alerting ban đầu được xây dựng tại SoundCloud. Vào năm 2012 nhiều công ty, tổ chức đã đứng ra bảo trợ cho Prometheus, và project này cực kỳ và phát và có rất nhiều người dùng. Hiện tại nó không còn là một project độc lập mà được phát triển bởi rất nhiều công ty khác nhau. Nó sử dụng mã nguồn GoLang của google. Hiện tại thì Prometheus 100% là open source và có Apache 2 License on GitHub.

**Tính năng:**

Prometheus sử dụng mã nguồn Go nên có thể dễ dàng build với các file binary, nó lấy metric của device cực kỳ nhanh và nhẹ. Một số tính năng chính mình xin trích dẫn lại từ trang chủ của Prometheus:

* A multi-dimensional data model with time series data identified by metric name and key/value pairs
* PromQL, a flexible query language to leverage this dimensionality
* No reliance on distributed storage; single server nodes are autonomous
* Time series collection happens via a pull model over HTTP
* Pushing time series is supported via an intermediary gateway
* Targets are discovered via service discovery or static configuration
* Multiple modes of graphing and dashboarding support

**Kiến trúc**



**II. Grafana**

Grafana là một nền tảng để xây dựng các analytics và monitoring. Nói nôm na cho mọi người dễ hình dung 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 ví dụ như cpu, ram, dish, network, iops, session.

Việc xây dựng Dashboard nó là một phần quan trọng trong việc monotor của hệ thống. Grafana support rất nhiều giải pháp monitor khác nhau.



**III. Cài đặt và cấu hình**

Mô hình

**A close up of a sign

Description automatically generated**

Trong mô hình này Monitor Server chạy trên nền CentOS7 (sử dụng CentOS8 với tập lệnh tương tự). Để nhanh + gọn = tất cả các node kết nối với nhau qua card VMNET8 (NAT) của VMWARE

**1. Cài Prometheus trên Monitor Server**

**Cách 01: Cài đặt Prometheus bằng Docker (tham khảo internet)**

**Cách 02: Cài đặt Prometheus bằng Package.**

* Update Linux, disable selinux, sync NTP linux:

#yum update -y

#ntpdate 1.ro.pool.ntp.org

#vim /etc/sysconfig/selinux

Change “SELINUX=enforcing” to “SELINUX=disabled”.

Save file. Sau đó reboot lại server.

* Mở port: 3000/tcp, 9090/tcp, 22/tcp

Download source prometheus tại: <https://github.com/prometheus/prometheus/releases>

#tar xvzf TEN\_FILE.tar.gz

#mv prometheus-2.20.1.linux-amd64 /usr/local/prometheus/

Tạo service prometheus trong systemd

#vim /etc/systemd/system/prometheus.service

Nội dung file là:

[Unit]

Description=Prometheus

Wants=network-online.target

After=network-online.target

[Service]

User=root

Group=root

Type=simple

ExecStart=/usr/local/prometheus/prometheus \

--config.file /usr/local/prometheus/prometheus.yml \

--storage.tsdb.path /usr/local/prometheus/ \

--web.console.templates=/usr/local/prometheus/consoles \

--web.console.libraries=/usr/local/prometheus/console\_libraries

[Install]

WantedBy=multi-user.target

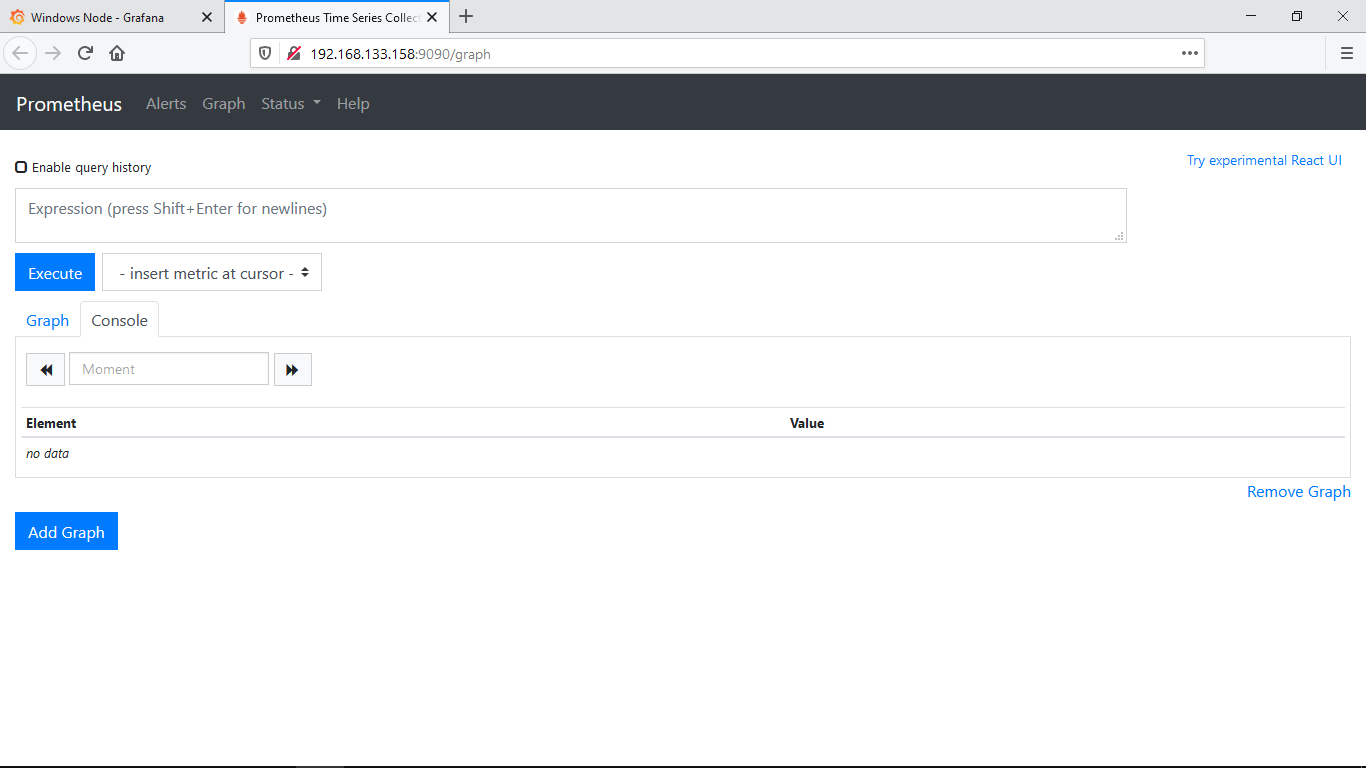
Restart và enable services.

#systemctl daemon-reload

#systemctl start prometheus

#systemctl status prometheus

Truy cập vào Prometheus: <http://IP:9090> để kiểm tra



**2. Cài Grafana trên Monitor Server**

Download tại: <https://grafana.com/grafana/download>

#wget <https://dl.grafana.com/oss/release/grafana-7.1.3-1.x86_64.rpm>  
#service grafana-server start

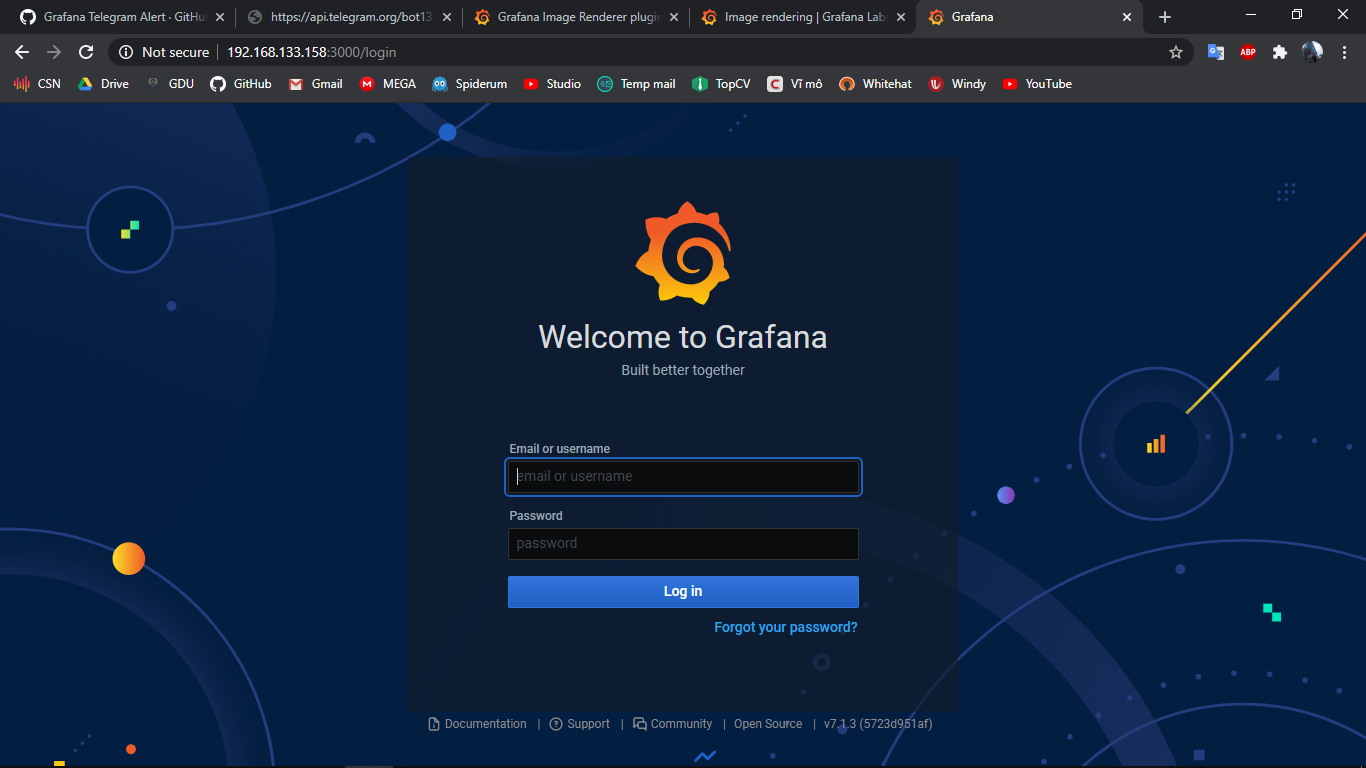
#systemctl daemon-reload

#systemctl start grafana-server

#systemctl status grafana-server

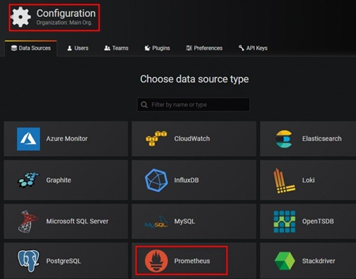
#systemctl enable grafana-server.service

Truy cập vào Grafana: http://IP:3000

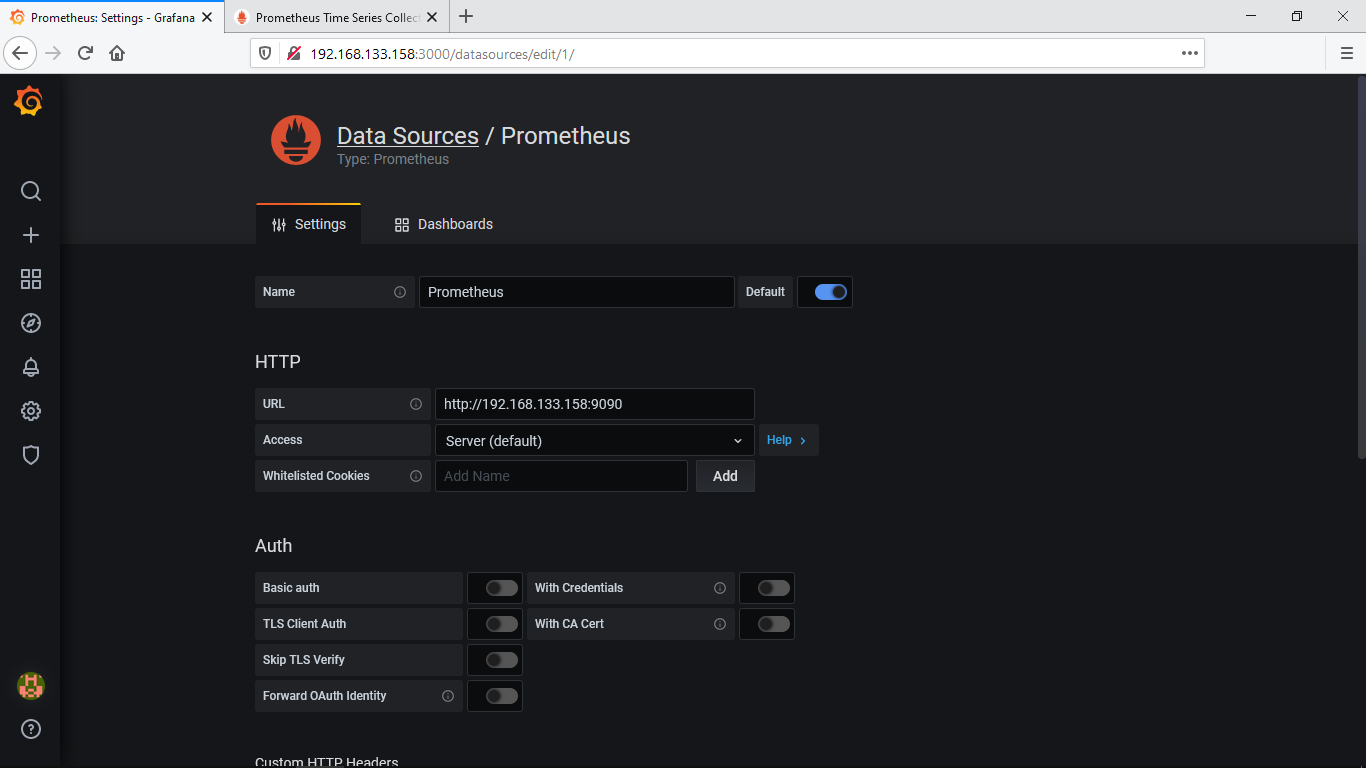


=> Passwork mặt định là admin/admin, và sau đó sẽ đổi password sau lần đầu tiên đăng nhập

Chọn Configuration, sau đó chọn data source type là Prometheus



Nhập thông tin của prometheus, sau đó save lại.



=> Như vậy chúng ta đã hoàn thành việc kết nối giữa Grafana và Prometheus

**3. Giám sát Windows Server**

Để giám sát được các server Windows, chúng ta cần phải cài đặt windows\_exporter tên cũ là wmi\_exporter (Agent collectors) để Promethues có thể thu thập được metric từ các server Windows này.

Nếu chỉ đơn thuần download các agent này cài đặt vào máy chủ Windows thì bạn chỉ có thể thu thập metric ở mức cơ bản. Khi cài agent qua CMD sẽ giúp các bạn chọn lọc thu thập những loại metric nào cần thu thập, giảm các loại metric không cần thiết, có thể thu thập nhiều loại metric hơn mặc định. Dưới dây là hướng dẫn bạn thu thập metric của những services nâng cao cho service Windows. Ví dụ như: AD, DNS, SQL, IIS, …

**Trên Sever Windows**

Download windows\_exporter

Link download Agent <https://github.com/prometheus-community/windows_exporter/releases>

Có 2 phiên bản Agent:

**windows\_exporter.exe** (click to run, dành cho các bạn nào chỉ cần các metric được enable sẵn).

**windows\_exporter.msi** (dùng để cài đặt thông qua CMD, enable các tính năng thu thập metric nâng cao).

Mở port9182/tcp trên windows server

Mở CMD với quyền Administrator.

Chạy comment msiexec với cú pháp như sau:

***msiexec /i C:\windows\_exporter-0.13.0-amd64.msi ENABLED\_COLLECTORS=”ad,cpu,cs,logon,memory,logical\_disk,os,service,system,process,tcp,net,textfile,thermalzone”***

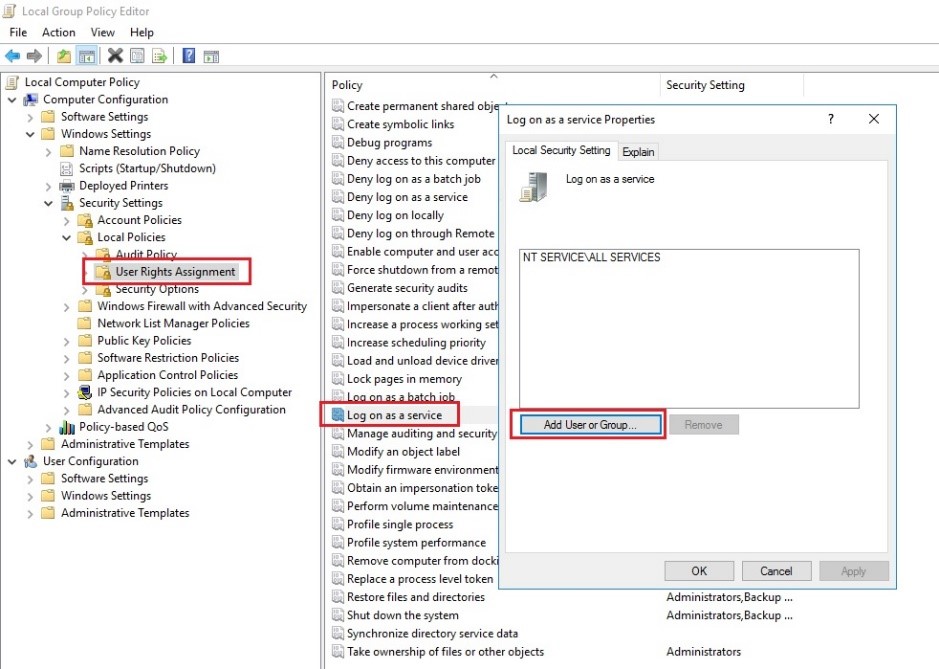
Trong đó ***C:\windows\_exporter-0.13.0-amd64.msi***là đường dẫn chứa file windows\_exporter.msi.

***ENABLED\_COLLECTORS=*”**các loại metric cần thu thập, tên của loại metric trong hình bên trên**”**

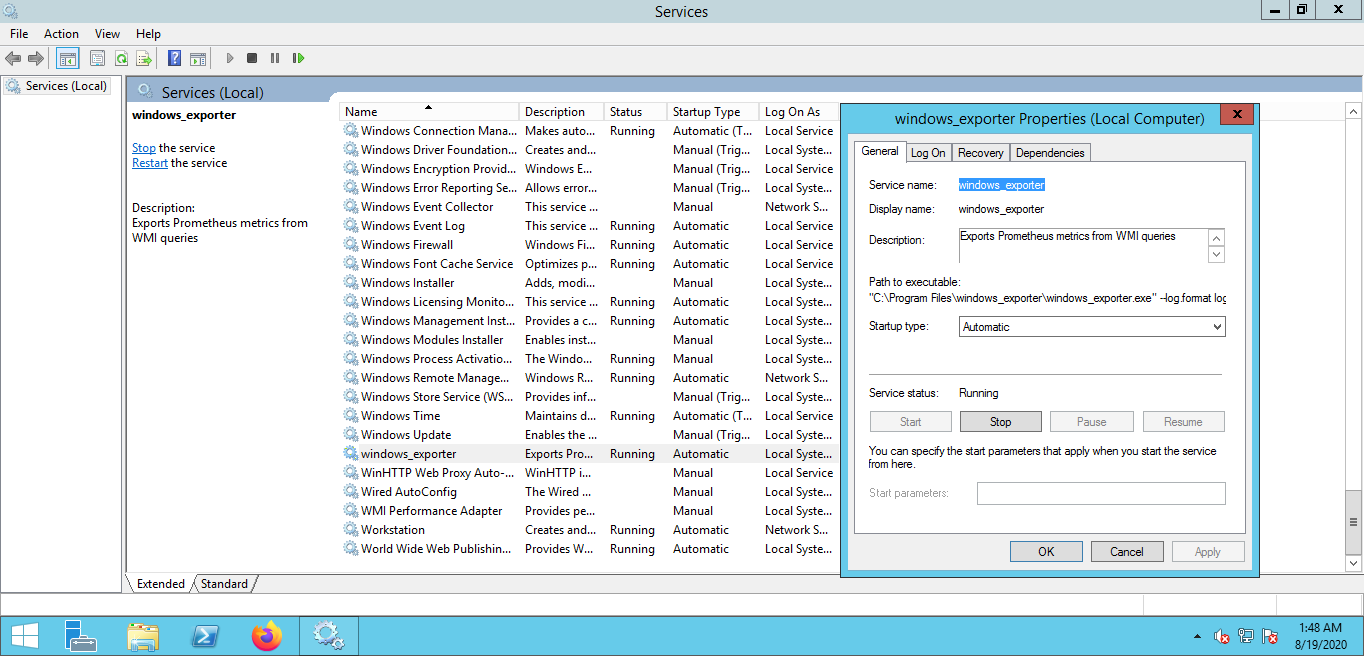
Tùy theo windows server của bạn mà lựa chọn metric cho phù hợp.

Nếu gặp lỗi Verify that you have sufficient privileges to install system services

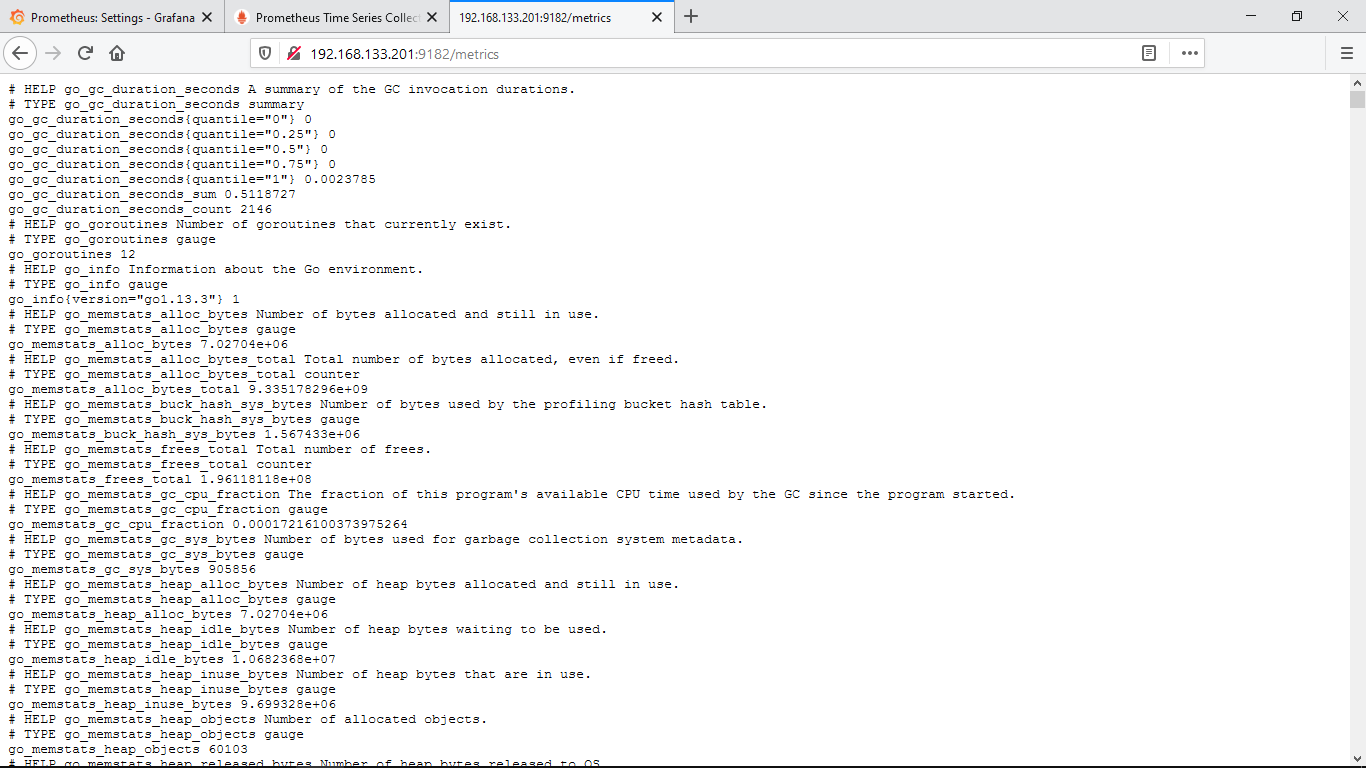
1. Run **gpedit.msc.**
2. Navigate to **Computer Configuration > Windows Settings > Security Settings > Local Policies >User Rights Assignment**.
3. In the details pane, double-click **Log on as a service**.
4. Click **Add User or Group**… and add the account to the list of accounts that have the Log on as a service right. Once you have selected the user, click **OK**.
5. Click **OK** and close the policy editor.



Kết quả sau khi cài đặt xong: bản mới service tên là windows\_exporter



Kiểm tra metric bạn truy cập như sau: [http://ipserver:9182](http://ipserver:9182/)



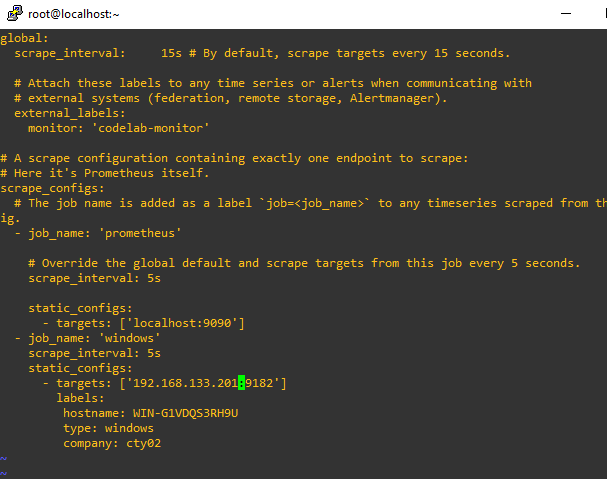
**Trên Monitor Server**

Tạo job trong prometheus để giám sát Windows Server này với nội dung sau:

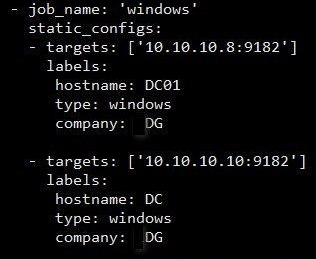
Datasheet: <https://prometheus.io/docs/prometheus/latest/getting_started/>

#nano /usr/local/prometheus/prometheus.yml

Tại đây mình thêm trường labels vào để dễ dàng filter metric.



Tương tự nếu bạn muốn add nhiều server windows trên cùng 1 job, bạn chỉ cần copy đoạn target và edit lại thông tin.



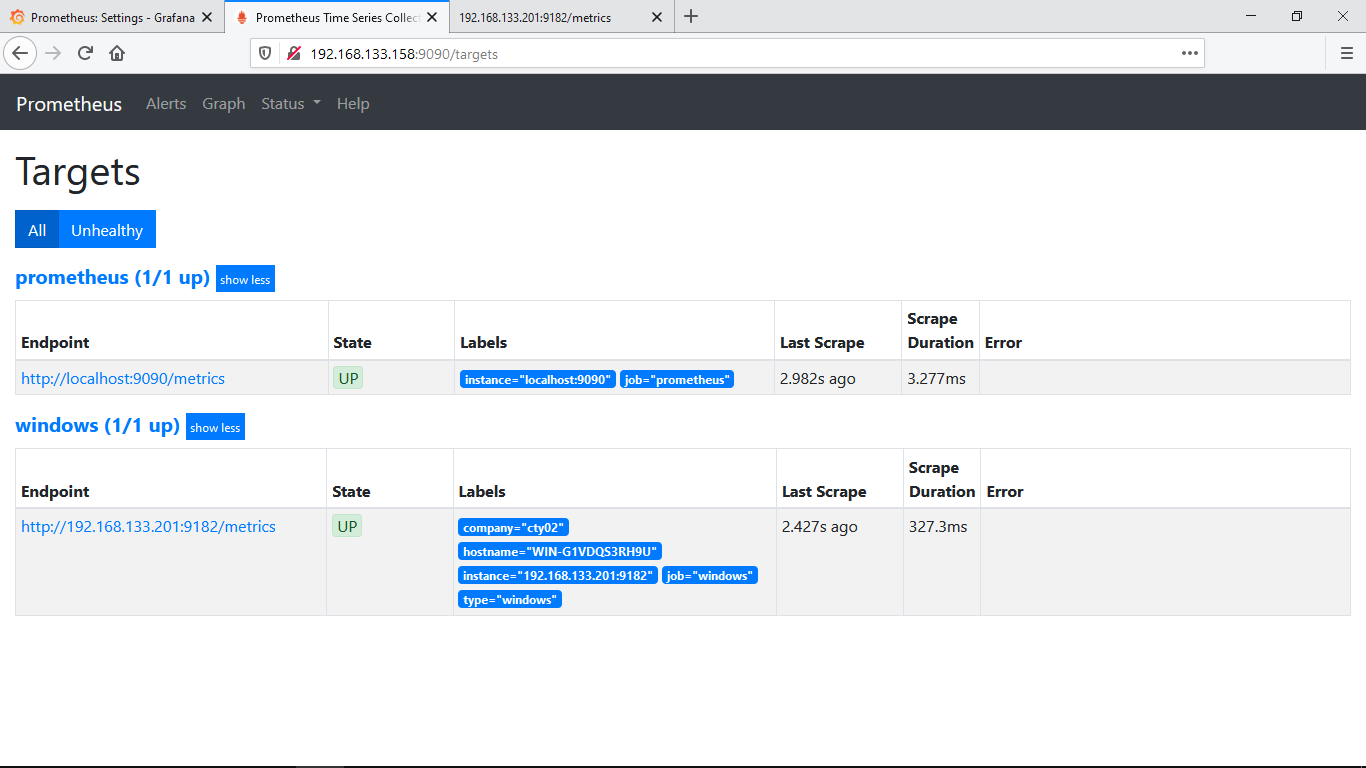
Restart serivce prometheus

#systemctl restart prometheus

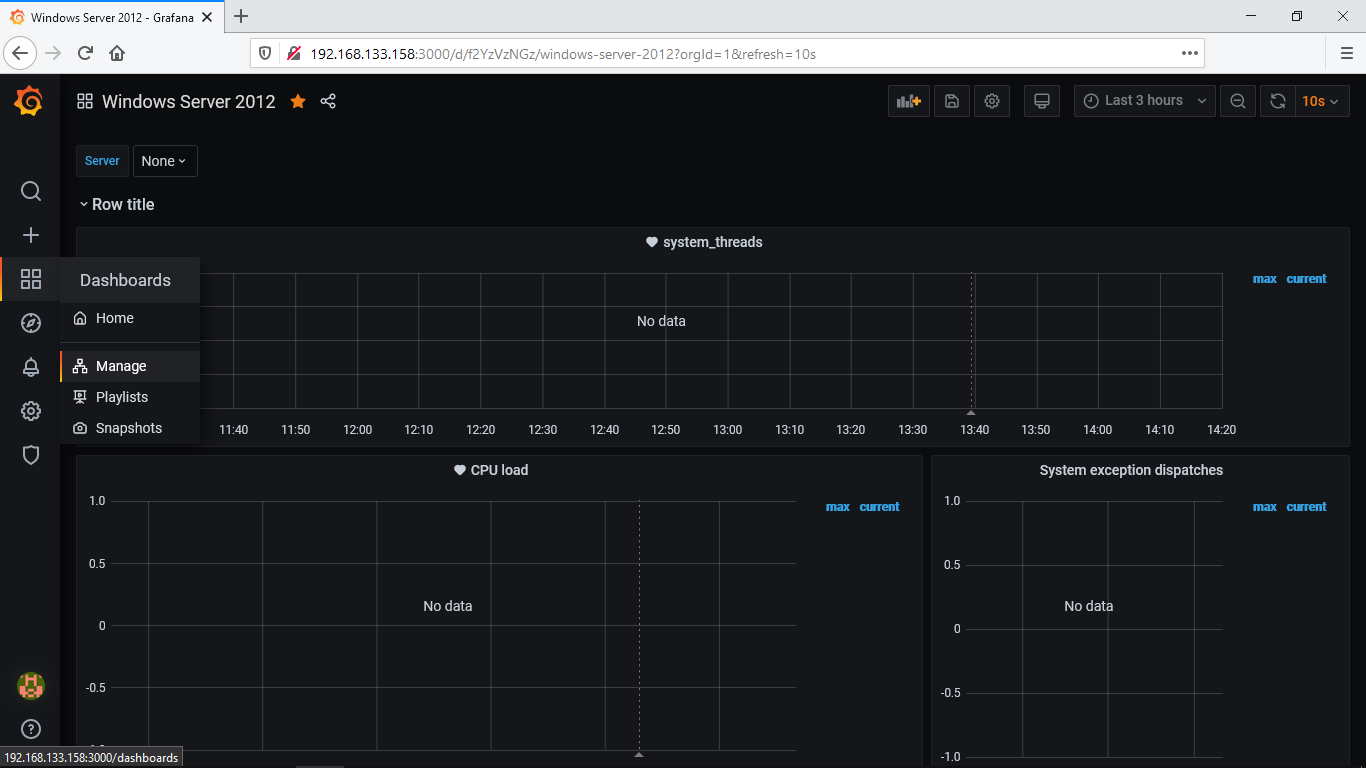
#systemctl status prometheus

Kiểm tra lại host được thêm vào prometheus thành công hay không.

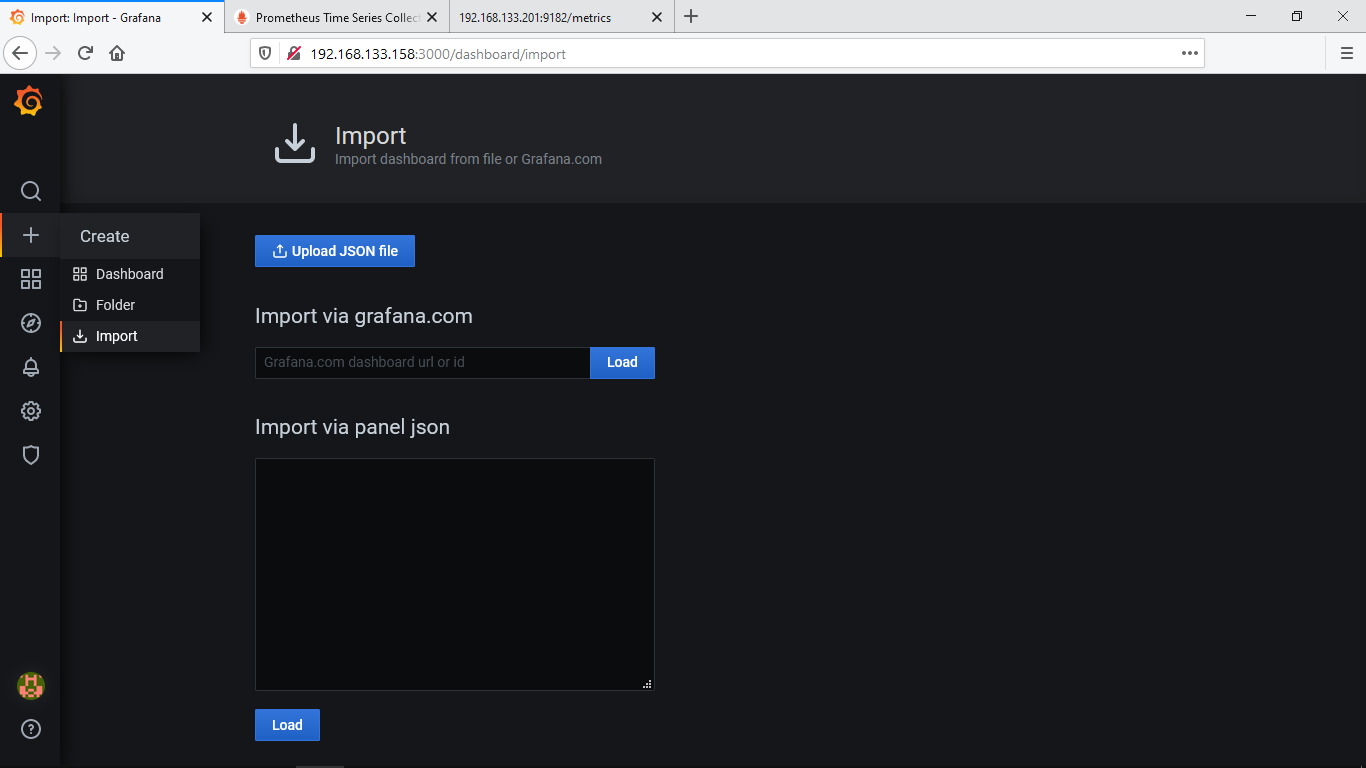
Truy cập vào Prometheus sau đó chọn Status 🡪 target để kiểm tra.



Tạo Dashboard monitor Windows Server trên Grafana. Bạn có thể tạo 1 Dashboard tổng cho tất cả các node rồi tạo các Dashboard riêng cho từng node để dễ quản lí nếu có nhiều node trong hệ thống.  
Truy cập vào**Grafana**, sau đó chọn **Manage Dashboards** như hình:



Tại đây bạn có thể chọn **New Dashboard** để tạo mới (tự add từng panel, recommend) hoặc **Import** (sử dụng những dashboard có sẳn rồi edit lại cho phù hợp). Tại đây chúng ta chọn **Import.** Có sẵn 1 file json cho node Windows mẫu trong cùng folder với file hướng dẫn này. Tiếp tục chọn vào **Upload .json file** để import dashboard

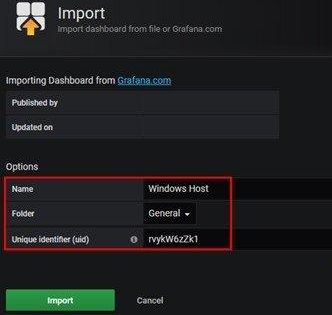


Sau khi chọn file cần import bạn cần chú ý các tham số sau:

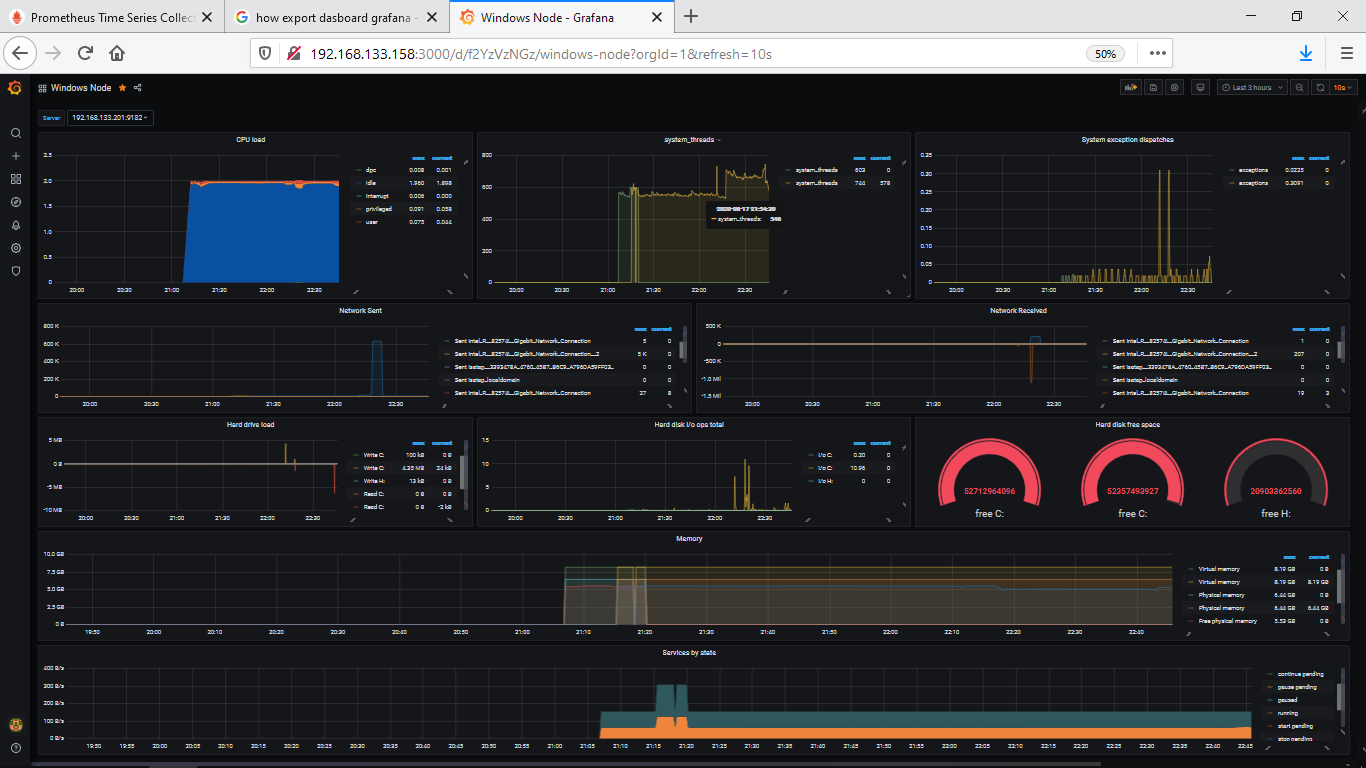
**Name :** tên của dashboard.

**Folder :** thư mục chứa dashboard.

**UID :** số định danh của dashboard, nếu số này trùng thì khi bạn import dashboard cũ sẽ bị ghi đè.



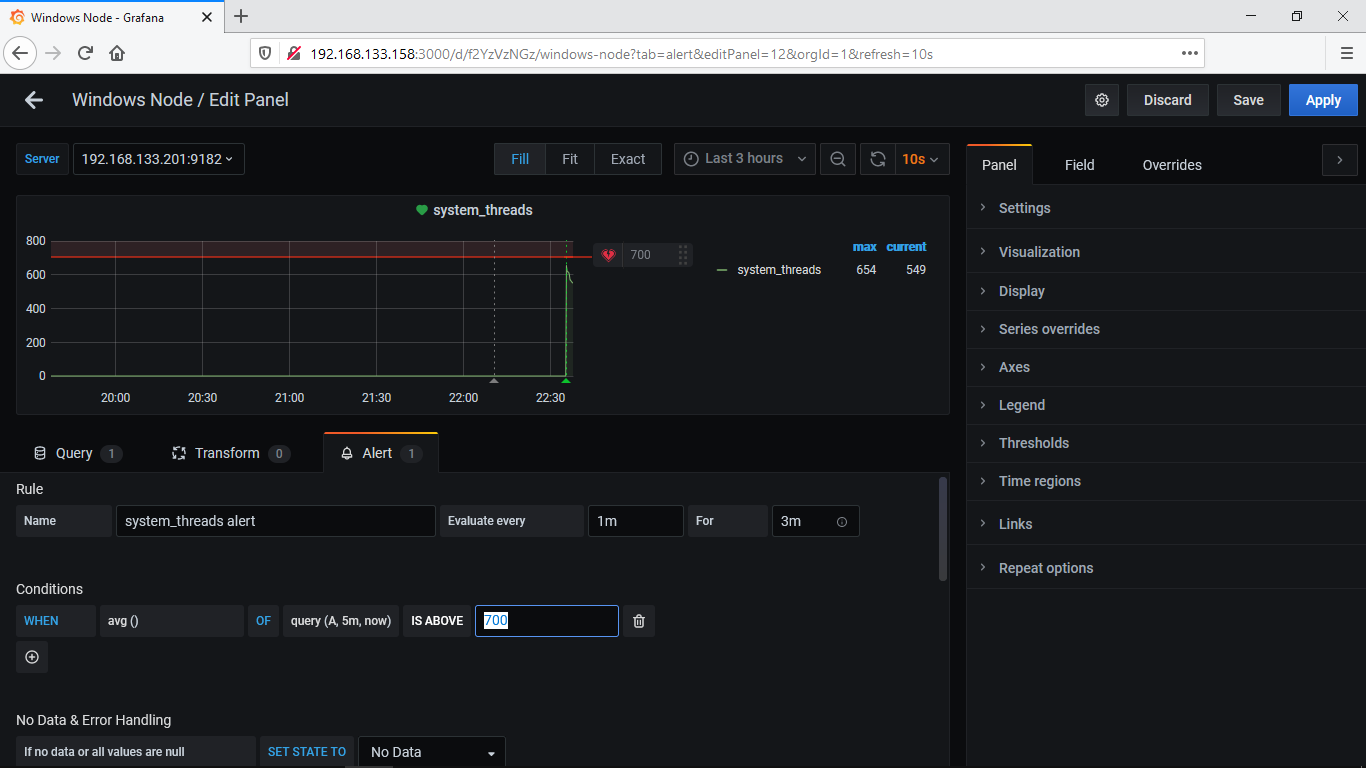
Đây là kết quả sau khi bạn import dashboard thành công.



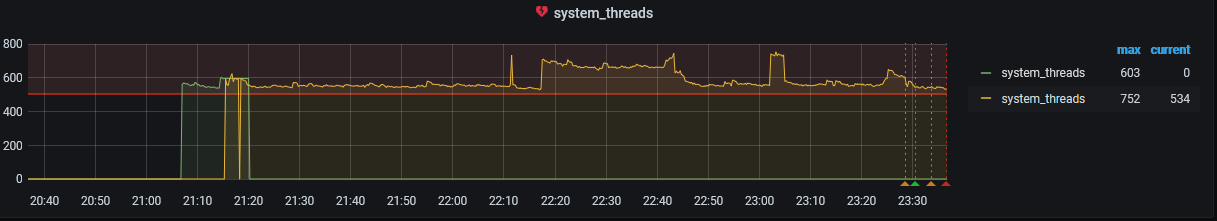
Hoặc download các dashboard mẫu tại đây: <https://grafana.com/grafana/dashboards>

**Tạo cảnh báo:**

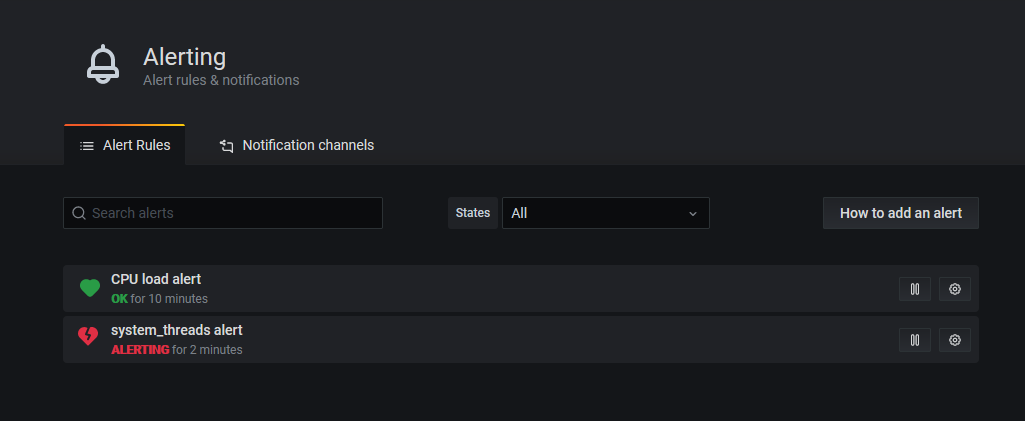
Có thể tạo 1 Panel chứa tất cả Alert hoặc tạo Alert trong từng Panel. Chọn tên, thời gian đánh giá cũng như giá trị ngưỡng của Alert này. Bạn nên tìm hiểu kĩ các thông số trong này để tạo cảnh báo như ý muốn. Save + apply. Lưu ý, vì đang làm lab nên hãy chỉnh thông số sao cho dễ tạo event giả để test tính năng.



Cảnh báo hiển thị trong Panel của Dashboard



Cảnh báo hiển thị trong Alerting



**4. Giám Sát Linux Server**

**Trên Linux Server**

Tắt selinux

Download node\_exporter trên Server cần giám sát.

<https://github.com/prometheus/node_exporter/releases/>

Giải nén source code và copy đến đường dẫn sau /usr/local/node\_exporter

#tar -xvzf TEN\_FILE.tar.gz

#mv node\_exporter-1.0.1.linux-amd64/node\_exporter /usr/local/node\_exporter

Tạo service trong systemd cho node\_exporter

***#vim /etc/systemd/system/node\_exporter.service***

Nội dung trong file như sau:

*[Unit]*

*Description=Prometheus Node Exporter*

*Wants=network-online.target*

*After=network-online.target*

*[Service]*

*User=root*

*Group=root*

*Type=simple*

*ExecStart=/usr/local/node\_exporter/node\_exporter*

*[Install]*

*WantedBy=multi-user.target*

Restart và enable service

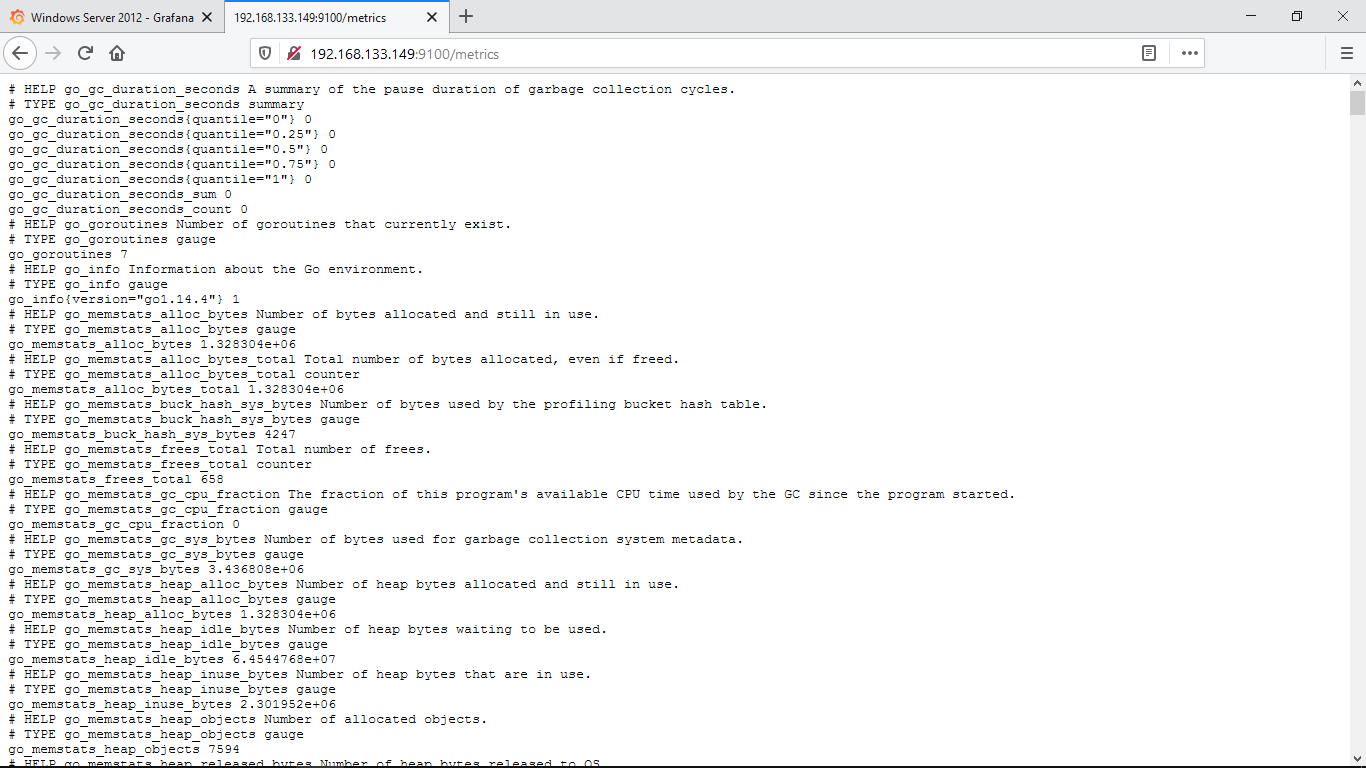
#systemctl daemon-reload

#systemctl start node\_exporter

#systemctl enable node\_exporter

Mở port 9100/tcp

Truy cập vào node\_exporter để kiểm tra http://IP:9100



**Trên Monitor Server**

Tạo job trong prometheus để giám sát server Linux này với nội dung job sau:

*- job\_name: ‘linux’*

*scrape\_interval: 5s*

*static\_configs:*

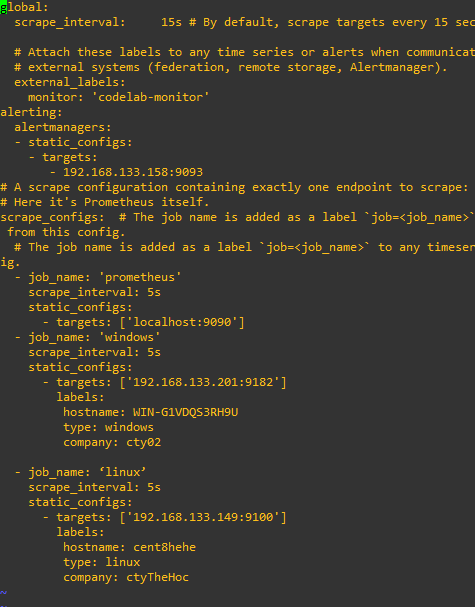
*- targets: ['192.168.133.149:9100']*

*labels:*

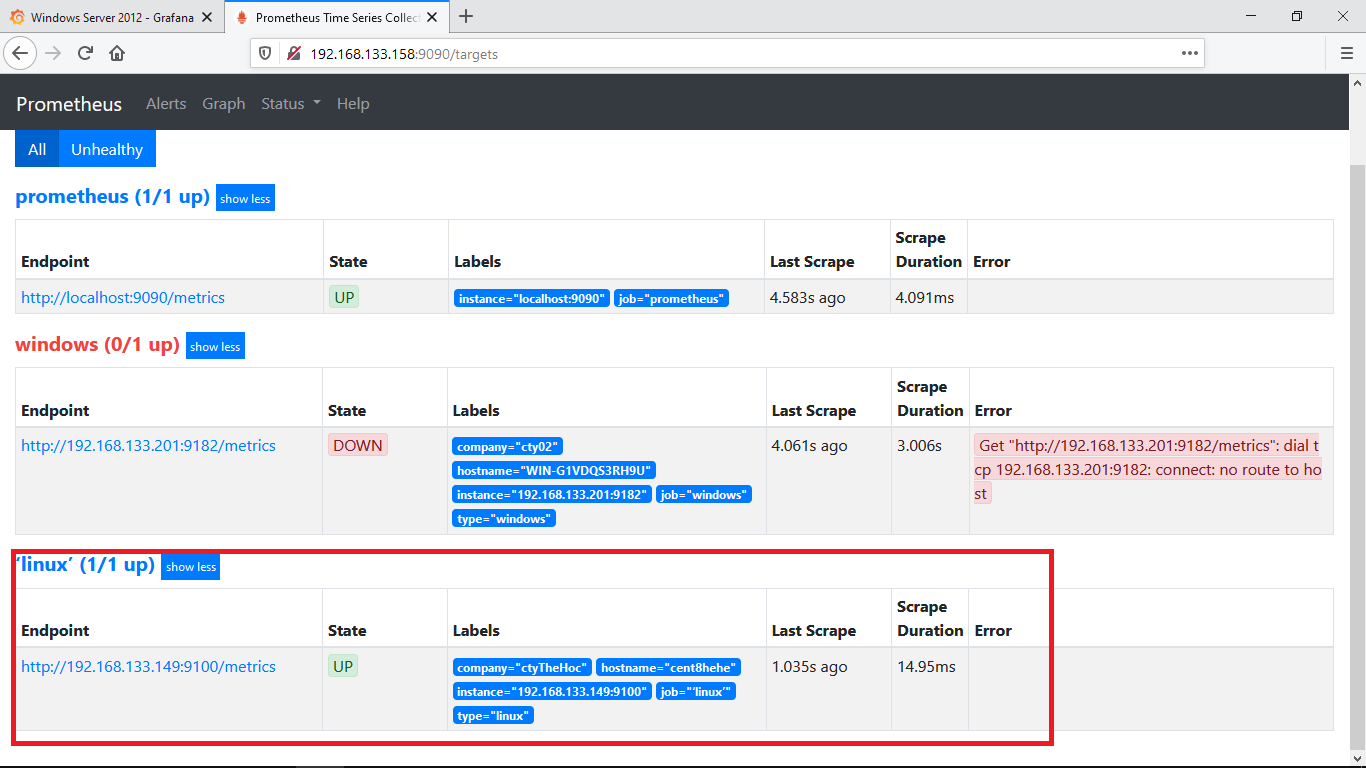
*hostname: cent8hehe*

*type: linux*

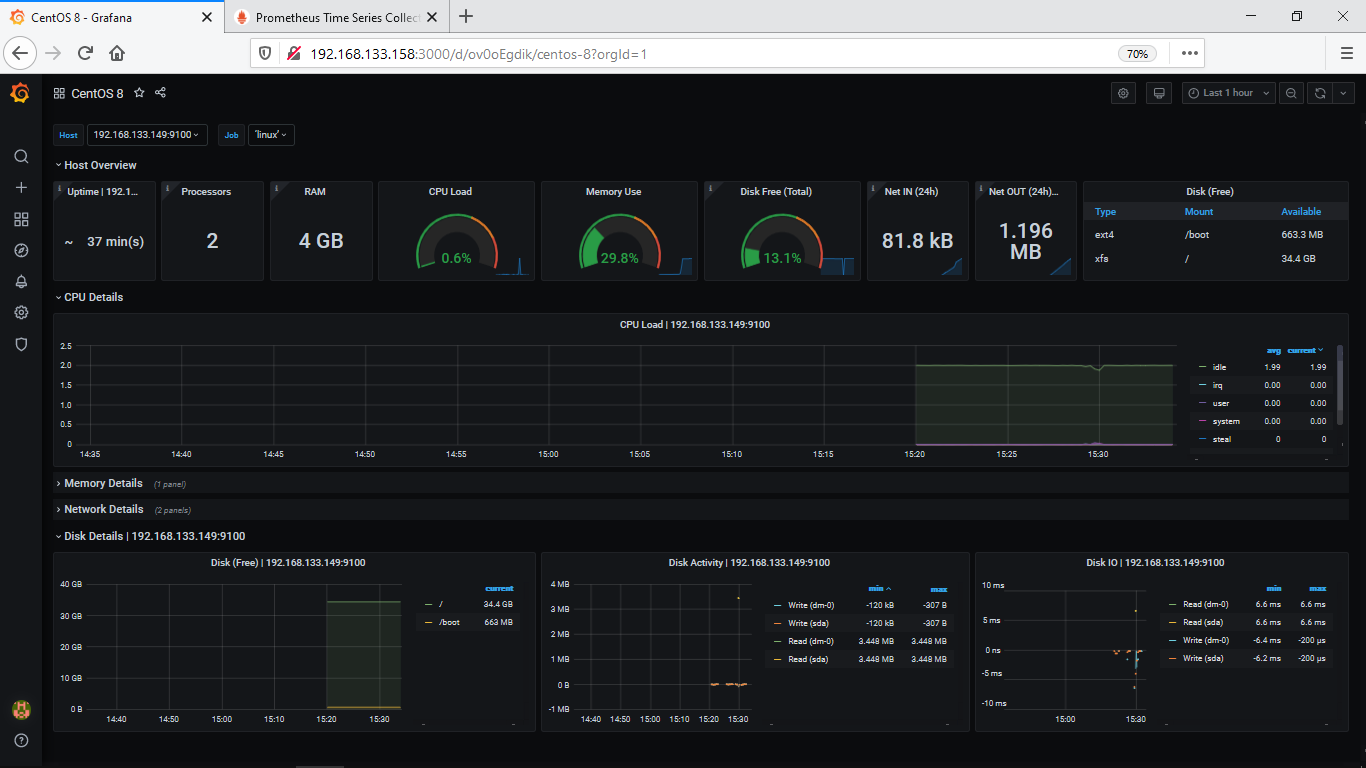
*company: ctyTheHoc*



Restart lại prometheus service và kiểm tra kết quả. http://IPMonitor:9090

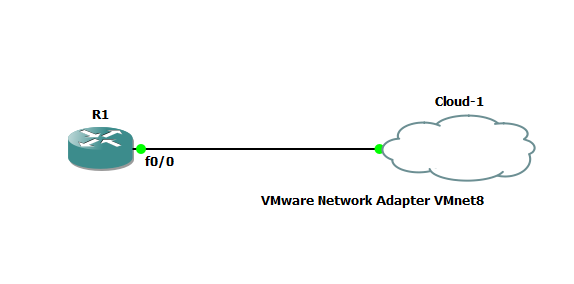


Đăng nhập vào Grafana để tạo dashboard và giám sát Linux Server



**5. Giám sát thiết bị Cisco**

- Đảm bảo kết nối thông suốt giữa router và Monitor Server

****

Để prometheus có thể thu thập được metric của Cisco bạn phải enable snmp trên những device Cisco.  Để enable snmp:

*R1(config)#snmp-server community snmpmonitor RO*

Để giám sát các thiết bị mạng, các hệ thống khác sẽ dựa vào snmp để thu thập metric. Prometheus cũng không ngoại lệ, nó sử dụng snmp\_exporter để thu thập metric trên các thiết bị mạng. Do vậy suy ra những gì có snmp thì promethues sẽ thu thập metric được.

Để snmp\_exporter hoạt động được, chúng ta cần phải có file **snmp.ym**l. Với snmp\_exporter nó cung cấp cho chúng ta 1 tool để generate ra file snmp.yml này.  Nội dung trong file snmp.yml này chính là các tham số walk và oid của mibs.

Trên Monitor Server chúng ta cần phải install GoLang để sử dụng **SNMP Exporter Config Generator.**

Download source GoLang tại đây <https://golang.org/dl/>

Tiếp tục giải nén file vừa download và chuyển nó đến đường dẫn **/usr/local/**

#tar -xvzf TENFILE.tar.gz

#mv go /usr/local/

Thiết lập các biến môi trường GOROOT và GOPATH

#export GOROOT=/usr/local/go

#export GOPATH=$HOME/go

#export PATH=$GOPATH/bin:$GOROOT/bin:$PATH

Kiểm tra version GoLang và verify lại cấu hình

*#go version*

*#go env*

Enable GoLang tự khởi chạy sau khi restart, thêm dòng lệnh thiết lập các biến môi trường vào .bashrc

#vi .bashrc

Bỏ đoạn export vào cuối file và save lại.

***export GOROOT=/usr/local/go***

***export GOPATH=$HOME/go***

***export PATH=$GOPATH/bin:$GOROOT/bin:$PATH***

Sử dụng **SNMP Exporter Config Generator** biên dịch và download mibs.

#yum install git zip unzip gcc gcc-g++ make net-snmp net-snmp-utils net-snmp-libs net-snmp-devel # RHEL-based distros

Sử dụng **go get** để download **generator**về

#go get github.com/prometheus/snmp\_exporter/generator

Thực hiện biên dịch và download mibs từ tool generator.

#cd ***/****root/go/src/github.com/prometheus/snmp\_exporter/generator/*

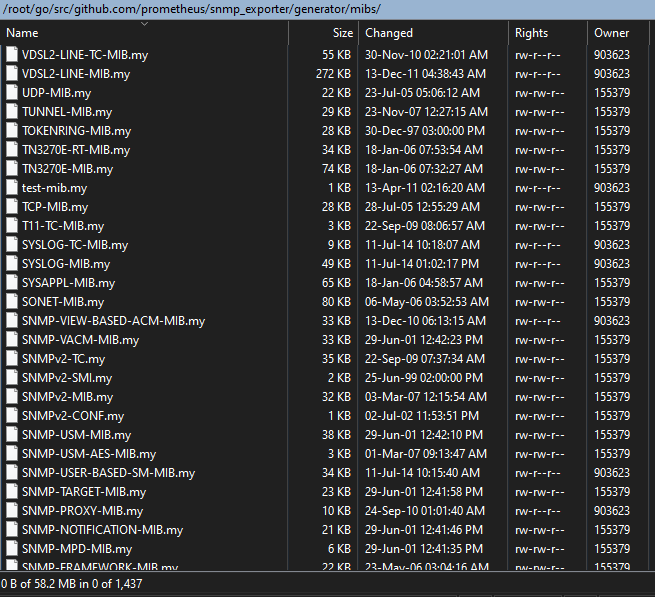
#go build

#make mibs

Bạn cũng có thể download từ hãng về và bỏ vào thư mục chứa các mibs ***/root/go/src/github.com/prometheus/snmp\_exporter/generator/mibs***/

Link Download:

* Cisco: <ftp://ftp>.cisco.com/pub/mibs/v2/v2.tar.gz
* Palo Alto PanOS 7.0 enterprise MIBs: <https://www.paloaltonetworks.com/content/dam/pan/en_US/assets/zip/technical-documentation/snmp-mib-modules/PAN-MIB-MODULES-7.0.zip>
* Synology: <https://global.download.synology.com/download/Document/Software/DeveloperGuide/Firmware/DSM/All/enu/Synology_MIB_File.zip>
* MikroTik: <http://download2.mikrotik.com/Mikrotik.mib>



Edit file generator.yml và thêm module cho router Cisco. Mặc định file generator.yml có rất nhiều thông tin trong này, chúng ta khuyến khị chỉ sử dụng những thông tin liên quan đến thiết bị cần monitor.

*#nano go/src/github.com/prometheus/snmp\_exporter/generator/generator.yml*

Nội dung file:

modules:

cisco:

walk:

- 1.3.6.1.2.1.2.2.1.1

- 1.3.6.1.2.1.2.2.1.2

- 1.3.6.1.2.1.2.2.1.10

- 1.3.6.1.2.1.2.2.1.13

- 1.3.6.1.2.1.2.2.1.14

- 1.3.6.1.2.1.2.2.1.16

- 1.3.6.1.2.1.2.2.1.19

- 1.3.6.1.2.1.2.2.1.2

- 1.3.6.1.2.1.2.2.1.20

- 1.3.6.1.2.1.2.2.1.5

- 1.3.6.1.2.1.2.2.1.7

- 1.3.6.1.2.1.2.2.1.8

- 1.3.6.1.2.1.31.1.1.1.1

- 1.3.6.1.2.1.31.1.1.1.18

- 1.3.6.1.4.1.9.9.48.1.1.1.5

- 1.3.6.1.4.1.9.9.48.1.1.1.6

- 1.3.6.1.2.1.1.5

- 1.3.6.1.4.1.9.1.1250

- 1.3.6.1.4.1.9.1.1251

lookups:

- source\_indexes: [ifIndex]

lookup: ifAlias

- source\_indexes: [ifIndex]

lookup: ifDescr

- source\_indexes: [ifIndex]

# Use OID to avoid conflict with Netscaler NS-ROOT-MIB.

lookup: 1.3.6.1.2.1.31.1.1.1.1 # ifName

overrides:

ifAlias:

ignore: true # Lookup metric

ifDescr:

ignore: true # Lookup metric

ifName:

ignore: true # Lookup metric

ifType:

type: EnumAsInfo

version: 2

max\_repetitions: 25

retries: 3

timeout: 10s

auth:

community: snmpmonitor

Trong đó  các dãy số này là SNMP OID:

***- 1.3.6.1.2.1.2.2.1.1***

***- 1.3.6.1.2.1.2.2.1.2***

Kiểm tra lại thông tin về OID những thiết bị của bạn để thay đổi cho chính xác:  <https://cric.grenoble.cnrs.fr/Administrateurs/Outils/MIBS/>

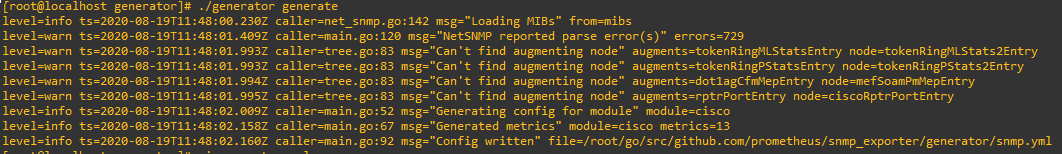
***version: 2***tương ứng với snmp v2

***community: snmpmonitor***là community khai báo trong phần enable  snmp của cisco.

Sau đó export mibs và generate ra file snmp.yml

#export MIBDIRS=mibs

#./generator generate



Như vậy chúng ta đã có file snmp.yml

Cài đặt **smnp\_exporter** lên prometheus server

Link download source snmp\_exporter tại đây: <https://github.com/prometheus/snmp_exporter/releases>

Giải nén và copy source đến thư mục **/usr/local/snmp\_exporter**

#tar -xvzf TEN-FILE.tar.gz

#mv snmp\_exporter\* /usr/local/snmp\_exporter

Tạo service trong systemd cho snmp\_exporter

*#vi /etc/systemd/system/snmp\_exporter.service*

Nôi dung file snmp\_exporter.service như sau:

*[Unit]*

*Description=Snmp\_exporter*

*Wants=network-online.target*

*After=network-online.target*

*[Service]*

*User=root*

*Group=root*

*Type=simple*

*ExecStart=/usr/local/snmp\_exporter/snmp\_exporter \*

*--config.file=/usr/local/snmp\_exporter/snmp.yml*

*[Install]*

*WantedBy=multi-user.target*

Copy file snmp.yml đến thư mục source của snmp\_exporter

*#cp -R /root/go/src/github.com/prometheus/snmp\_exporter/generator/snmp.yml /usr/local/snmp\_exporter/*

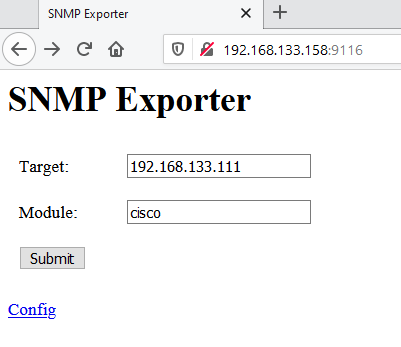
Restart và enable service

#systemctl restart snmp\_exporter.service

#systemctl status snmp\_exporter.service

#systemctl enable snmp\_exporter.service

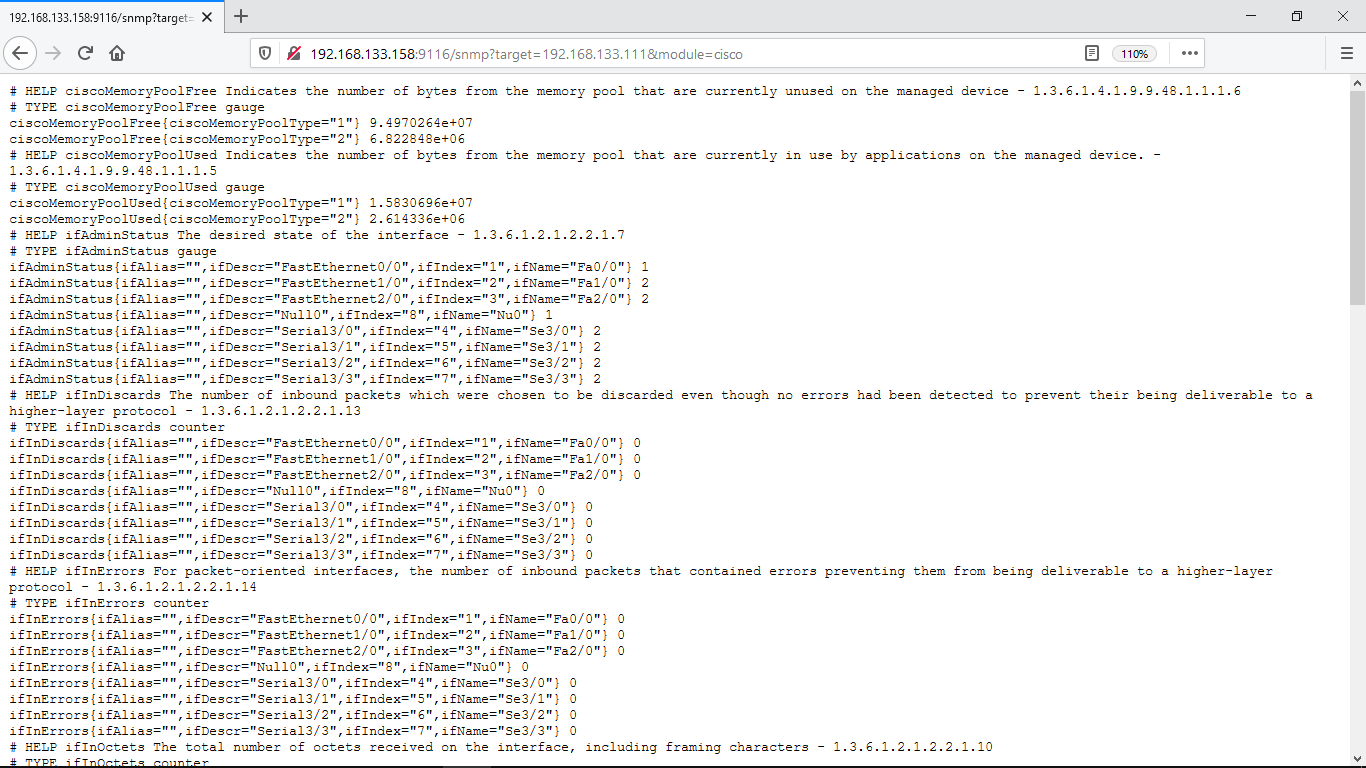
Mở port 9116/tcp. Truy cập vào **https://ipprometheus:9116** và điền các thông tin sau:



**Target:** là ip của cisco

**Module:** chính là module mà bạn tạo bằng generator cho cisco

Nếu xuất ra metric là OK.



Tạo job trong prometheus để giám sát cisco với nội dung job sau:

#vi /usr/local/prometheus/prometheus.yml

Nội dung như sau:

  - job\_name: ‘cisco’

    static\_configs:

      - targets: [‘192.168.133.111’]

        labels:

         hostname: R111

         device: cisco

         company: ctyTuine

    scrape\_interval: 3m

    scrape\_timeout : 3m

    metrics\_path: /snmp

    params:

      module: [cisco]

    relabel\_configs:

      - source\_labels: [\_\_address\_\_]

        target\_label: \_\_param\_target

      - source\_labels: [\_\_param\_target]

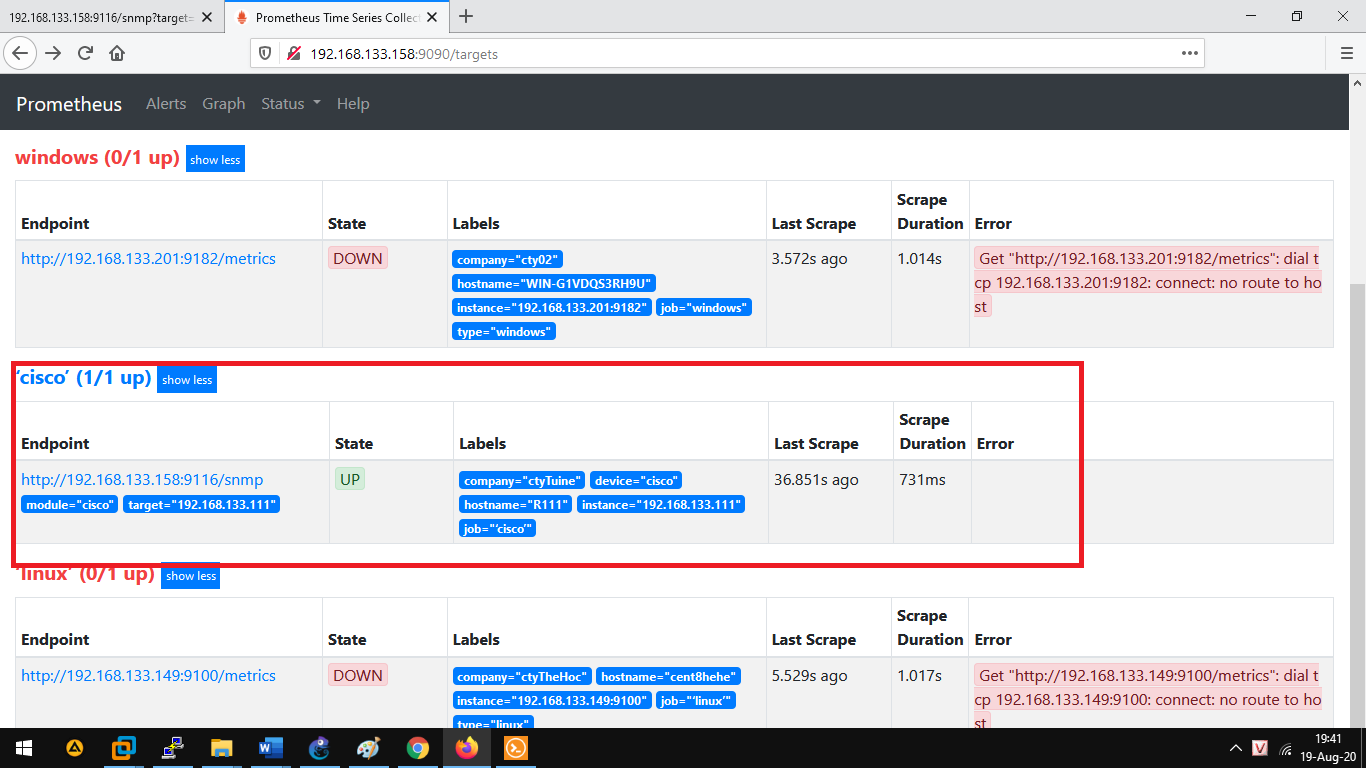
        target\_label: instance

      - target\_label: \_\_address\_\_

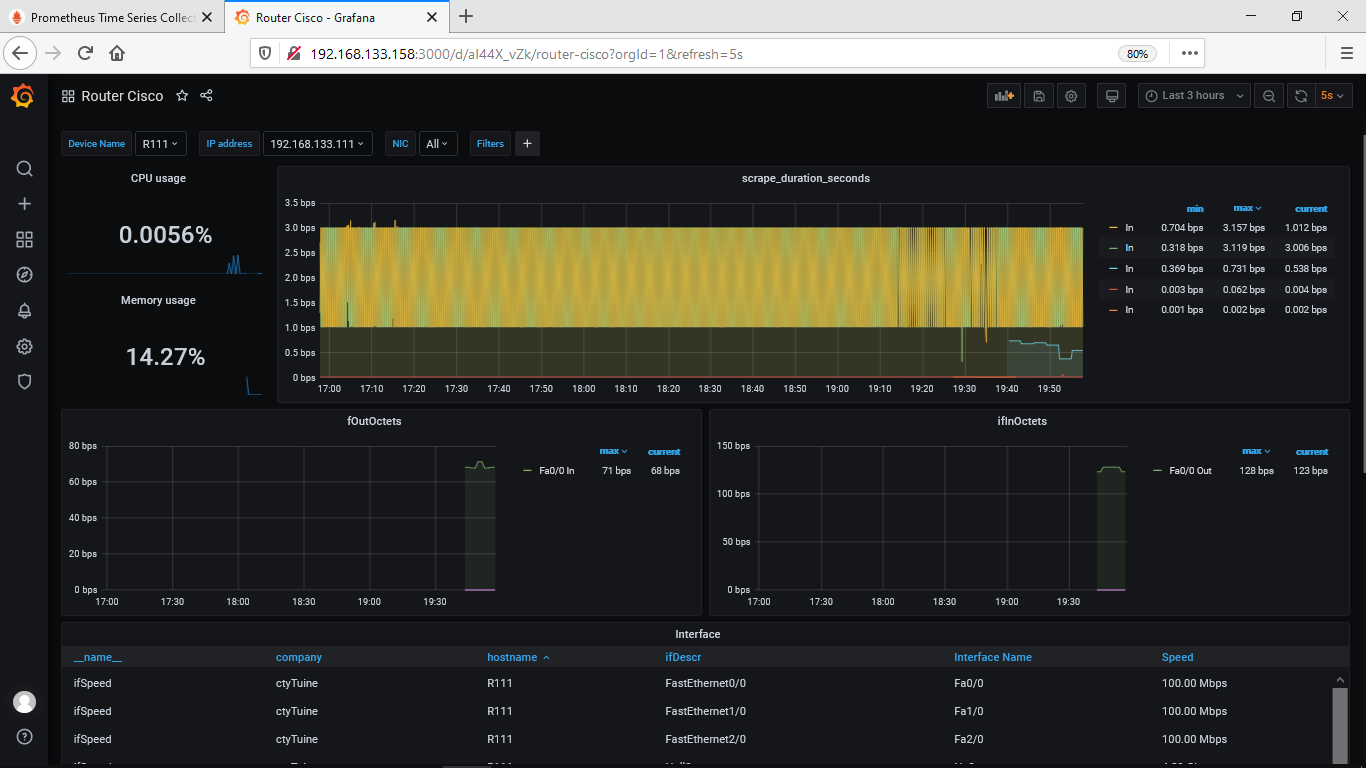
        replacement: 192.168.133.158:9116  # SNMP exporter

Tương tự nếu bạn muốn thêm nhiều device cisco vào, bạn chỉ cần chèn đoạn **target**vào là OK. Trong tất cả các job của prometheus mình đều thêm trường label vào để tiện query vào tạo dashboard trên grafana. Các bạn làm theo mình thì dash board sẽ không bị rổi.

Restart service và kiểm tra lại kết quả



Đăng nhập vào Grafana để tạo Dashboard để giám sát thiết bị Cisco



Như vậy tương tự cách của mình các bạn có thể ứng dụng nó vào để giám sát các thiết bị mạng của những hãng khác qua việc sử dụng snmp.

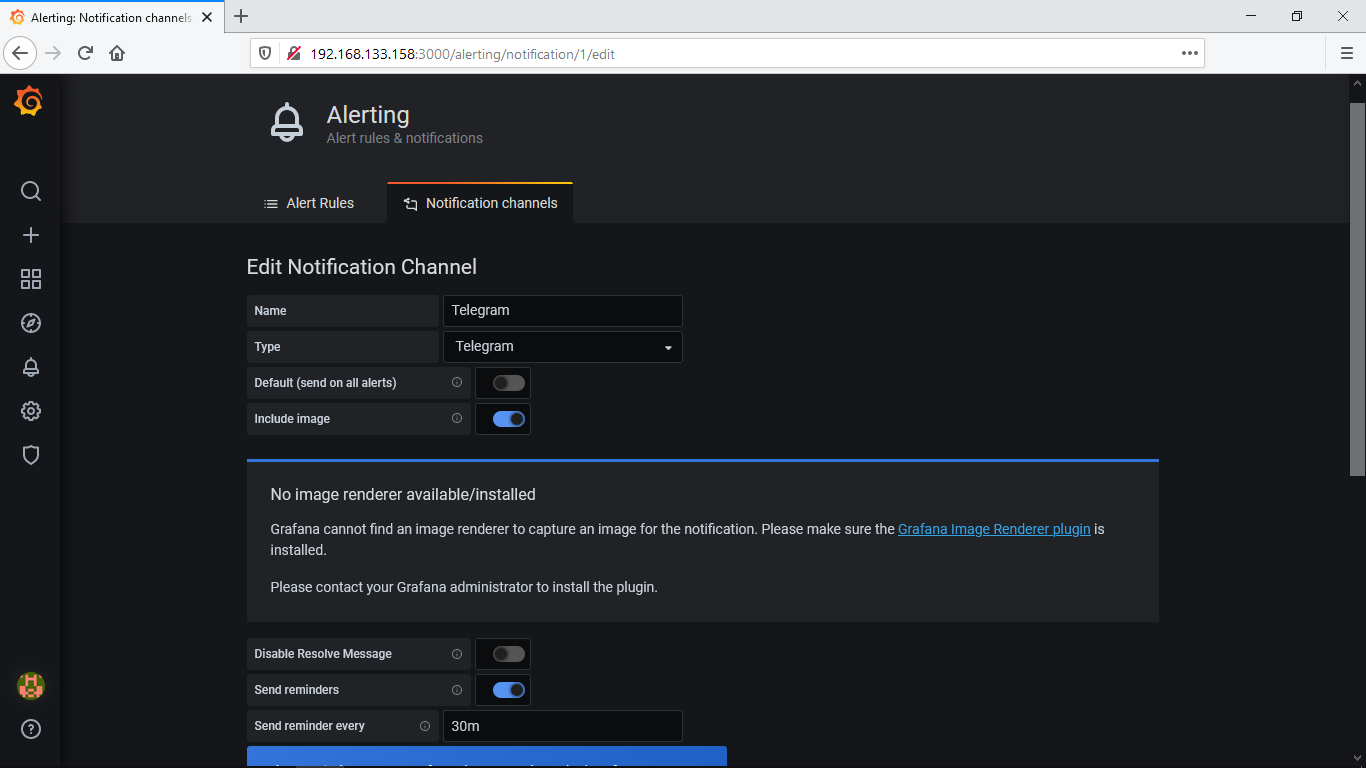
**6. Cấu hình Alert gửi tin nhắn qua Telegram**

Nếu bạn muốn gửi qua mail: <https://thriftly.io/docs/components/Thriftly-Deployment-Beyond-the-Basics/Metrics-Prometheus/Creating-Receiving-Email-Alerts.html>

Nhưng theo mình thấy trong task này thì Telegram nhanh và tiện hơn rất nhiều.

Vào Grafana > Alerting > Notification channels > New channel.

Chọn Type: Telegram. Bạn có thể chỉnh các thông số như là: default có gửi tất cả alert không, có hay không kèm hình ảnh (cần cài grafana-image-renderer), bật tắt tin báo OK, thời gian báo lại . . .



Tại đây yêu cầu 2 trường là BOT API TOKEN và Chat ID. Ta tiến hành tạo và lấy chúng như sau:

Mở Telegram tìm và chat với @BotFather

Chat

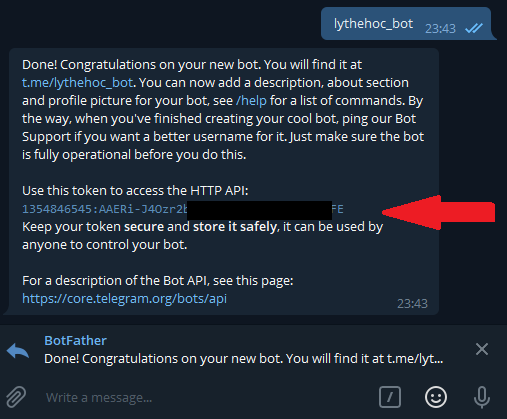
/newbot

Đặt tên bot, ví dụ:

Bot1

Đặt bots username (nên có \_bot ở cuối)

lythehoc\_bot



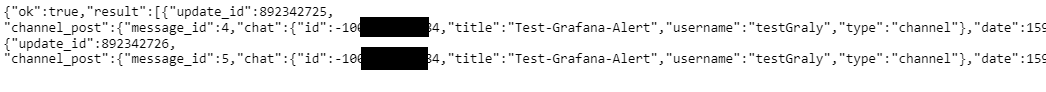
Token là dòng ngay mũi tên màu đỏ. Đến đây ta đã có BOT API TOKEN

Trên Telegram, tạo một channel mới (đây sẽ là nơi bạn nhận tin nhắn cảnh báo) trên Telegram sau đó mời con bot bạn vừa tạo vào channel này (cho nó quyền admin). Gửi ít nhất 1 tin nhắn trong channel, trong tin nhắn phải có từ “bot”. Copy BOT API TOKEN lúc nãy, Mở trình duyệt web và vào link sau:

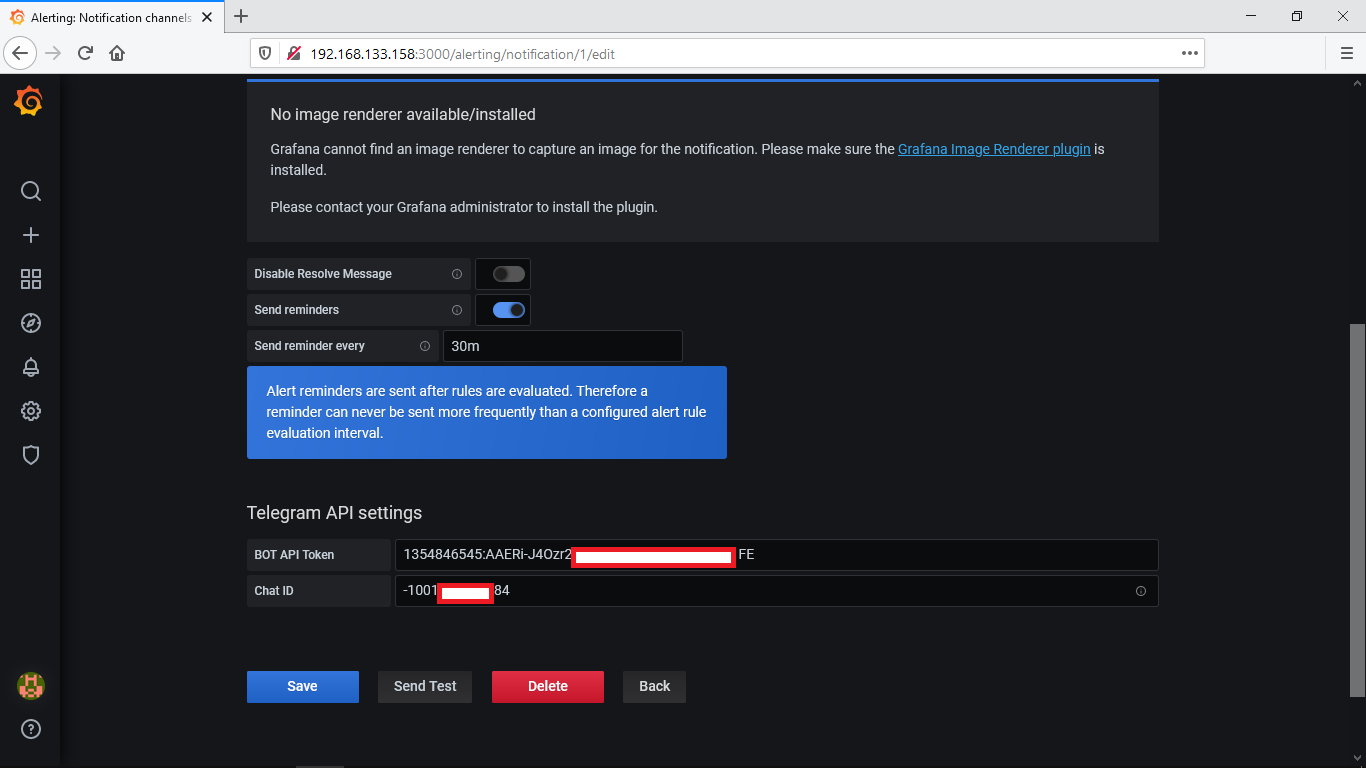
https://api.telegram.org/botTokenCuaBan/getUpdates

Thay TokenCuaBan bằng Token của bạn 😊

Kết quả trả về sẽ có cấu trúc như sau: {"ok":true,"result":[{"update\_id":BLA\_BLA\_BLA", chat":{"id":[CHAT\_ID],"title"



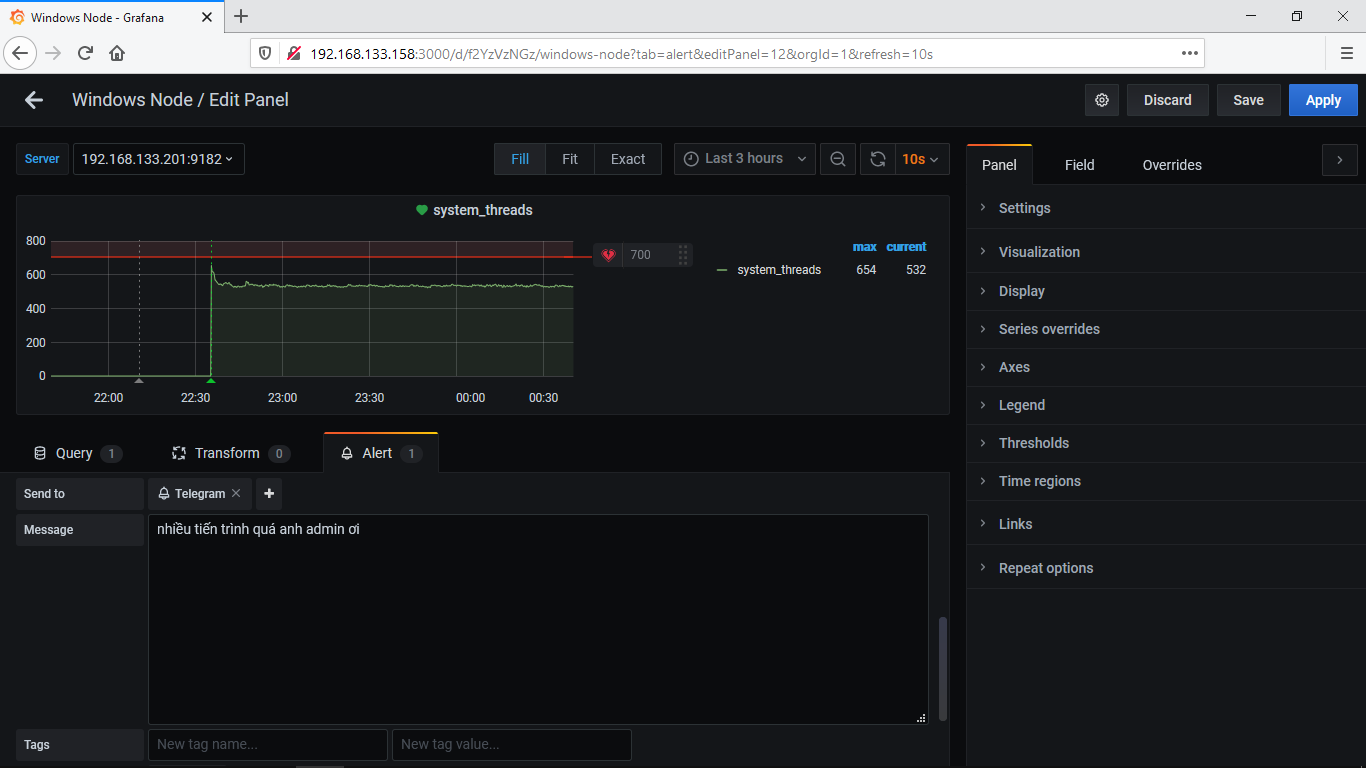
Copy Chat\_ID bỏ vào Grafana phải bao gốm cả dấu “-“ ở đầu:



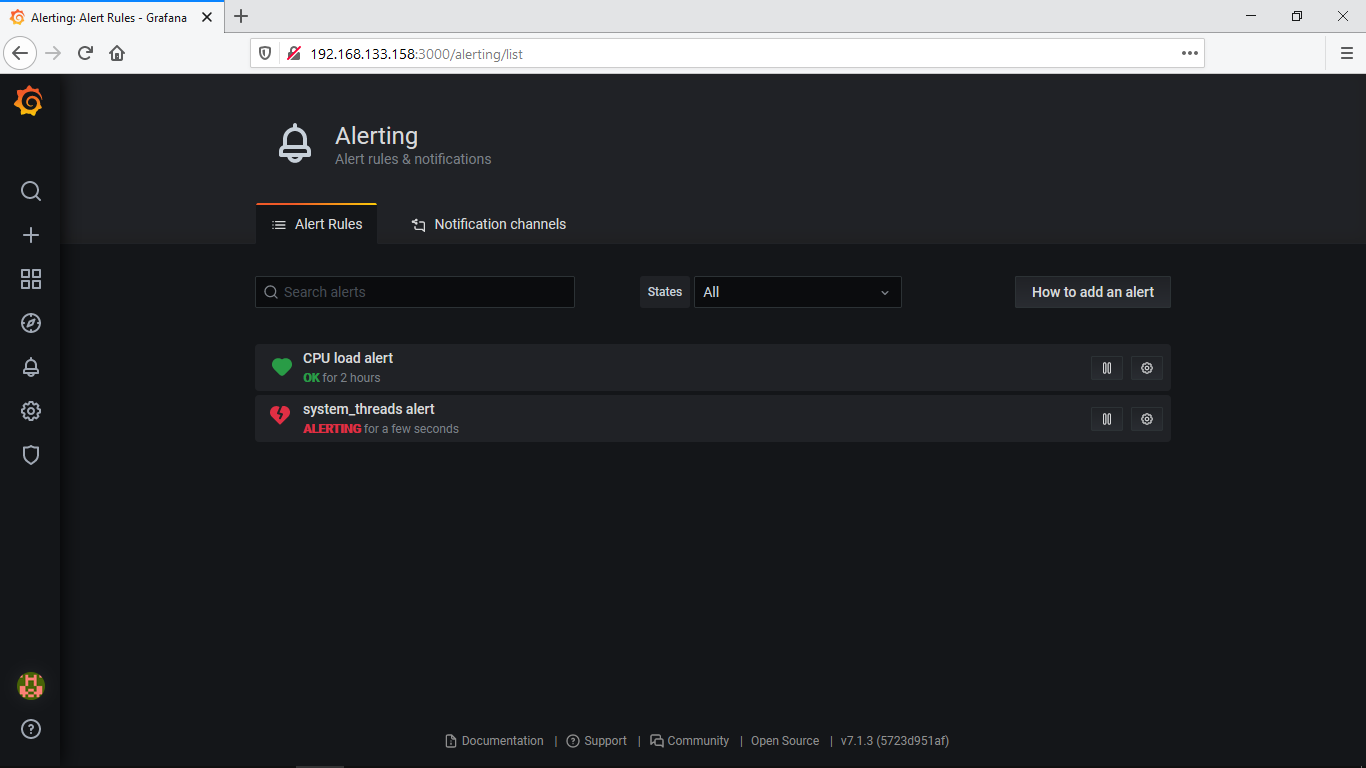
Click "Send Test". Nếu OK và nhận được tin nhắn test thì save



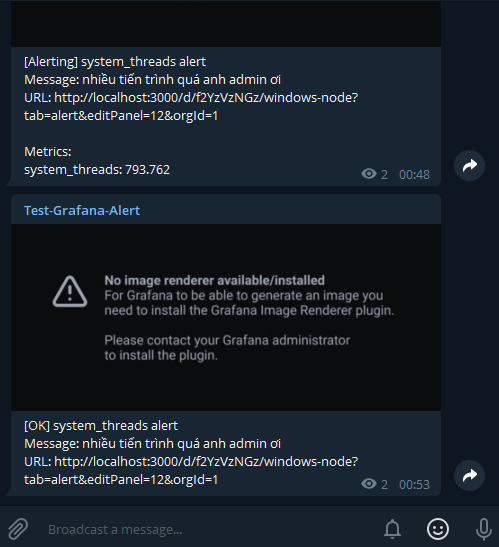
Tạo Alert như hướng dẫn phía trên, tại trường Send to ta chọn Telegram và chình sửa message.



Cảnh báo xuất hiện.



Kiểm tra tin nhắn thì có tin cảnh báo, Sau khi hệ thống hoạt động ổn định lại thì có tin báo OK. Vì lúc nãy ta có chọn gửi tin nhắn bao gồm hình ảnh nên nó gửi hình, nhưng không có hình vì chưa cài grafana-image-renderer.



Lệnh cài grafana-image-renderer

#grafana-cli plugins install grafana-image-renderer

“Minimum free memory recommendation is 16GB on the system doing the rendering”

**7. Các thiết bị khác**

Firewall Fortigate (snmp)

Firewall Pfsense (node\_exporter)

Vmware-vCenter (vmware\_exporter)

***\*Sau khi đã có metrics thì việc monitor có hiệu quả tối ưu hay không dựa vào cách bạn config Dashboard và Alert\****