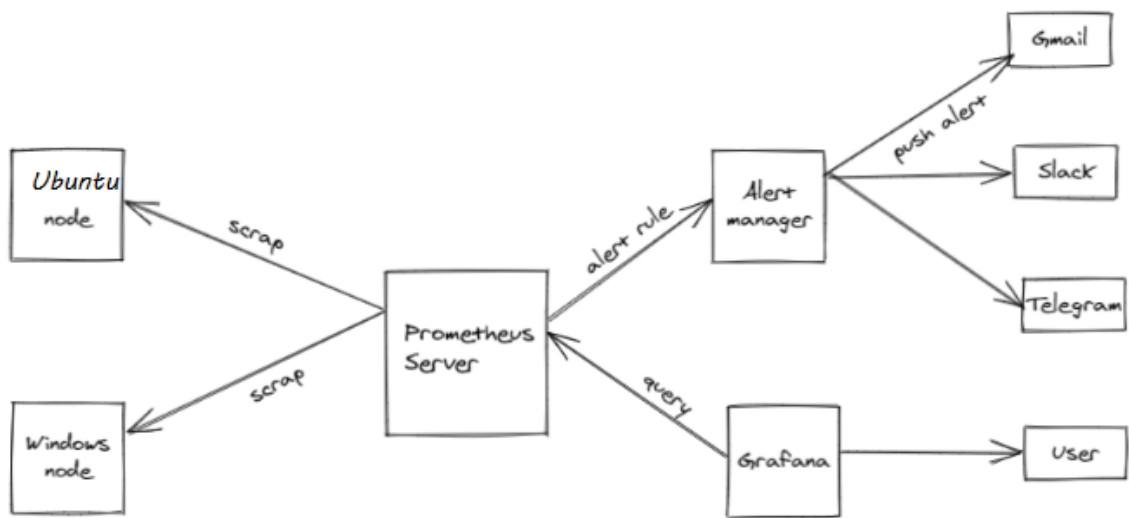
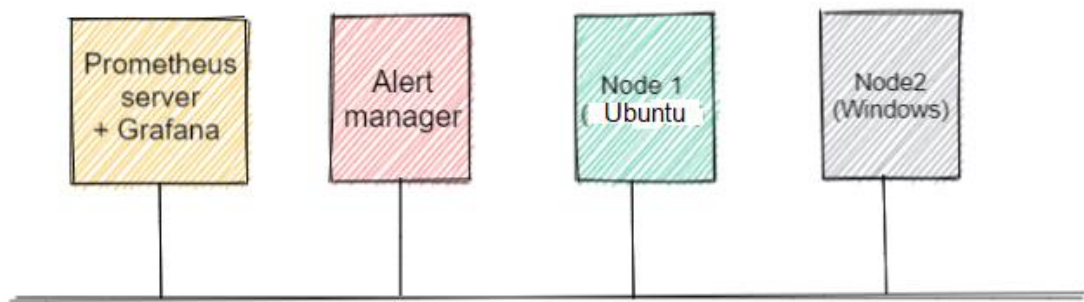


Mô hình tổng quát



Mô hình Lab



IP Planning

IP Planning			
Name	OS	IP	Cấu hình phần cứng
Prometheus Server + Grafana	Ubuntu Server 20.04	192.168.17.155	2cpu, 2g, 20gb
Alert Manager	Ubuntu Server 20.04	192.168.17.153	2cpu, 2g, 20gb
Node 1	Ubuntu Server 20.04	192.168.17.154	2cpu, 2g, 20gb

Node 2	Windows	192.168.5.249	2cpu, 2g, 20gb
--------	---------	---------------	----------------

## 1. Cài đặt cấu hình Prometheus server

Bước 1: Tải source code Prometheus

- wget  
<https://github.com/prometheus/prometheus/releases/download/v2.35.0/prometheus-2.35.0.linux-amd64.tar.gz>

Bước 2: Giải nén

- tar xvfz prometheus-2.35.0.linux-amd64.tar.gz

Bước 3: Di chuyển vào thư mục *prometheus-2.35.0.linux-amd64* và copy vào */usr/local/bin/prometheus*

- cd prometheus-2.35.0.linux-amd64
- cp -r . /usr/local/bin/Prometheus

Bước 4: Quản lý service bằng systemd

- sudo nano /etc/systemd/system/prometheus.service

[Unit]

Description=Prometheus Service

After=network.target

[Service]

Type=simple

ExecStart=/usr/local/bin/prometheus/prometheus --  
config.file=/usr/local/bin/prometheus/prometheus.yml

[Install]

WantedBy=multi-user.target

Bước 5: Chỉnh sửa scrape\_configs ( *Đổi localhost -> 192.168.17.155*)

- sudo vi /usr/local/bin/prometheus/prometheus.yml

- job\_name: "prometheus"

static\_configs:

- targets: ["192.168.17.155:9090"]

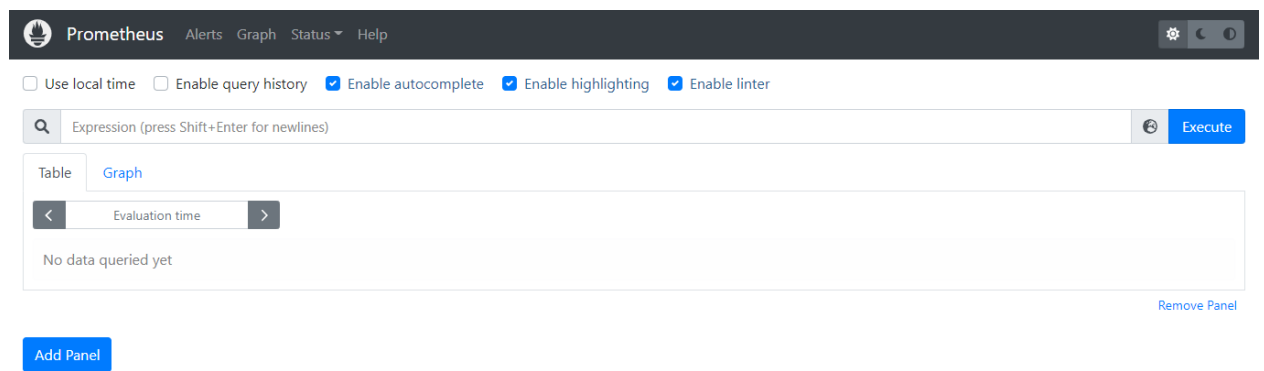
Bước 6: Khởi động Prometheus service

- systemctl daemon-reload
- sudo service prometheus start
- sudo service prometheus status
- sudo systemctl enable prometheus

Bước 7: Truy cập vào giao diện web Prometheus với đường link:

<http://192.168.17.155:9090/graph>

Giao diện của trang web



Metric của Prometheus Server có thể truy cập vào

<http://192.168.17.155:9090/metrics> để xem

```
# HELP go_gc_cycles_automatic_gc_cycles_total Count of completed GC cycles generated by the Go runtime.
# TYPE go_gc_cycles_automatic_gc_cycles_total counter
go_gc_cycles_automatic_gc_cycles_total 104
# HELP go_gc_cycles_forced_gc_cycles_total Count of completed GC cycles forced by the application.
# TYPE go_gc_cycles_forced_gc_cycles_total counter
go_gc_cycles_forced_gc_cycles_total 0
# HELP go_gc_cycles_total_gc_cycles_total Count of all completed GC cycles.
# TYPE go_gc_cycles_total_gc_cycles_total counter
go_gc_cycles_total_gc_cycles_total 104
# 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"} 6.5149e-05
go_gc_duration_seconds{quantile="0.25"} 0.000130671
go_gc_duration_seconds{quantile="0.5"} 0.000181139
go_gc_duration_seconds{quantile="0.75"} 0.000241226
go_gc_duration_seconds{quantile="1"} 0.01723198
go_gc_duration_seconds_sum 0.043846384
go_gc_duration_seconds_count 104
# HELP go_gc_heap_allocs_by_size_bytes_total Distribution of heap allocations by approximate size. Note that this does not include tiny objects as defined by
/go/heap/tiny/allocs:objects, only tiny blocks.
# TYPE go_gc_heap_allocs_by_size_bytes_total histogram
go_gc_heap_allocs_by_size_bytes_total_bucket{le="8.999999999999999"} 179732
go_gc_heap_allocs_by_size_bytes_total_bucket{le="24.999999999999999"} 7.633873e+06
go_gc_heap_allocs_by_size_bytes_total_bucket{le="64.999999999999999"} 9.934126e+06
go_gc_heap_allocs_by_size_bytes_total_bucket{le="144.999999999999999"} 1.0889562e+07
go_gc_heap_allocs_by_size_bytes_total_bucket{le="320.999999999999999"} 1.1362294e+07
go_gc_heap_allocs_by_size_bytes_total_bucket{le="704.999999999999999"} 1.138556e+07
go_gc_heap_allocs_by_size_bytes_total_bucket{le="1536.999999999999999"} 1.1402485e+07
go_gc_heap_allocs_by_size_bytes_total_bucket{le="3200.999999999999999"} 1.1409544e+07
go_gc_heap_allocs_by_size_bytes_total_bucket{le="6528.999999999999999"} 1.141591e+07
go_gc_heap_allocs_by_size_bytes_total_bucket{le="13568.999999999999999"} 1.141831e+07
go_gc_heap_allocs_by_size_bytes_total_bucket{le="27264.999999999999999"} 1.1421216e+07
go_gc_heap_allocs_by_size_bytes_total_bucket{le="+Inf"} 1.1424097e+07
go_gc_heap_allocs_by_size_bytes_total_sum 3.233434048e+09
go_gc_heap_allocs_by_size_bytes_total_count 1.1424097e+07
# HELP go_gc_heap_allocs_bytes_total Cumulative sum of memory allocated to the heap by the application.
# TYPE go_gc_heap_allocs_bytes_total counter
go_gc_heap_allocs_bytes_total 3.233434048e+09
# HELP go_gc_heap_allocs_objects_total Cumulative count of heap allocations triggered by the application. Note that this does not include tiny objects as defined by
/go/heap/tiny/allocs:objects, only tiny blocks.
```

## 2. Cài đặt Grafana

Bước 1:

wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -

Bước 2:

sudo add-apt-repository "deb https://packages.grafana.com/oss/deb stable main"

Bước 3:

sudo apt update

Bước 4: Cài đặt Grafana

sudo apt install grafana

Bước 5: Dùng systemctl để khởi động Grafana-server

sudo systemctl start grafana-server

Bước 6: Kiểm tra trạng thái

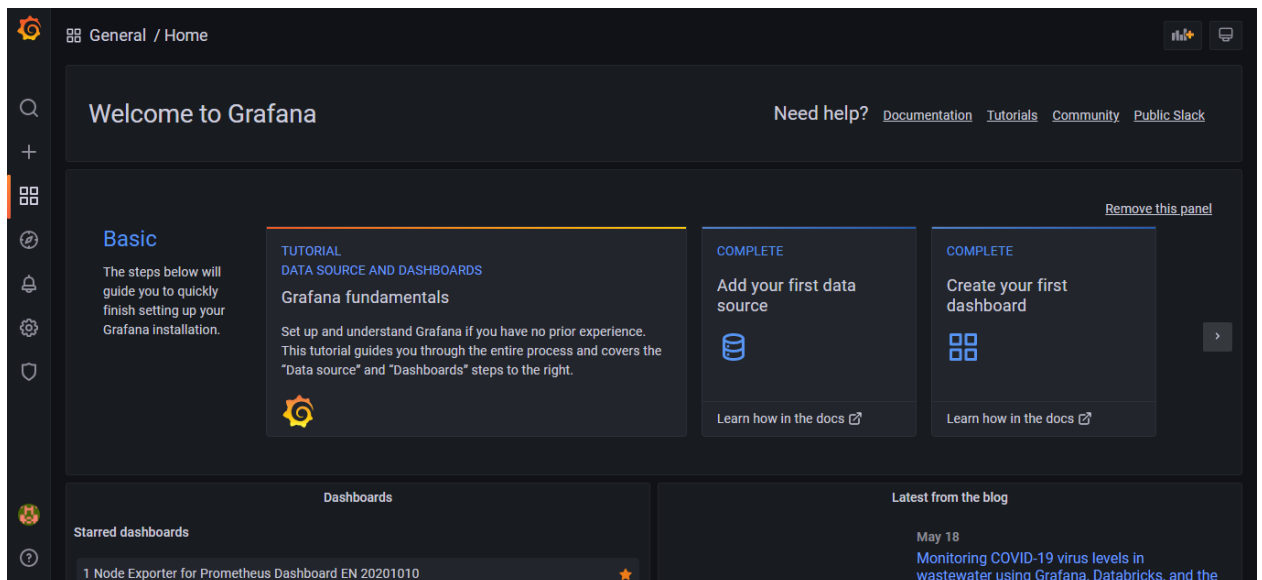
sudo systemctl status grafana-server

Bước 7: Kích hoạt dịch vụ khởi động cùng OS

sudo systemctl enable grafana-server

Bước 8: Truy cập vào giao diện Grafana với đường link:

<http://192.168.17.155:3000/>



### 3. Cài đặt Node Exporter thu nhập metric trên Ubuntu Server 20.04

Bước 1: Tải source code

wget

[https://github.com/prometheus/node\\_exporter/releases/download/v1.3.1/node\\_exporter-1.3.1.linux-amd64.tar.gz](https://github.com/prometheus/node_exporter/releases/download/v1.3.1/node_exporter-1.3.1.linux-amd64.tar.gz)

Bước 2: Giải nén

tar xzf node\_exporter-1.3.1.linux-amd64.tar.gz

Bước 3: Copy vào thư mục /usr/local/bin/

cp node\_exporter-1.3.1.linux-amd64/node\_exporter /usr/local/bin/node\_exporter

Bước 4: Quản lí service bằng system

- sudo nano /etc/systemd/system/node-exporter.service

[Unit]

Description=Prometheus Node Exporter Service

After=network.target

[Service]

Type=simple

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

[Install]

WantedBy=multi-user.target

Bước 5: Thêm scrape config mới cho node exporter

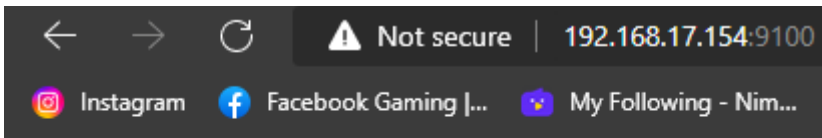
- job\_name: 'node-exporter'

static\_configs:

- targets: ['192.168.17.154:9100']

Bước 6: Khởi động Prometheus service

- systemctl daemon-reload
- sudo service node-exporter start
- sudo service node-exporter status
- sudo systemctl enable node-exporter



# Node Exporter

## Metrics

```
# 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"} 4.416e-05
go_gc_duration_seconds{quantile="0.25"} 0.000183032
go_gc_duration_seconds{quantile="0.5"} 0.000245959
go_gc_duration_seconds{quantile="0.75"} 0.000339337
go_gc_duration_seconds{quantile="1"} 0.007293988
go_gc_duration_seconds_sum 0.203087925
go_gc_duration_seconds_count 489
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 9
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info{version="go1.17.3"} 1
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 2.76572e+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 1.048871112e+09
# 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 1.550951e+06
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 8.052587e+06
# HELP go_memstats_gc_cpu_fraction The fraction of this program's available CPU time used by the GC since the program started.
# TYPE go_memstats_gc_cpu_fraction gauge
go_memstats_gc_cpu_fraction 7.364488184519274e-05
# 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 5.287544e+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 2.76572e+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 7.880704e+06
# HELP go_memstats_heap_inuse_bytes Number of heap bytes that are in use.
# TYPE go_memstats_heap_inuse_bytes gauge
```

## 4. Cài đặt Alert Manager

Bước 1: Tải source code và giải nén

- wget <https://github.com/prometheus/alertmanager/releases/download/v0.24.0/alertmanager-0.24.0.linux-amd64.tar.gz>
- tar xvfz alertmanager-0.24.0.linux-amd64.tar.gz

Bước 2: CD vào thư mục alertmanager-0.24.0.linux-amd64

- cd alertmanager-0.24.0.linux-amd64

Bước 3: Copy vào thư mục /usr/local/bin/

- cp -r . /usr/local/bin/alertmanager

Bước 4: Quản lý service bằng system

- sudo nano /etc/systemd/system/alertmanager.service

[Unit]

Description=Prometheus Alert Manager Service

After=network.target

[Service]

Type=simple

ExecStart=/usr/local/bin/alertmanager/alertmanager \\\n --config.file=/usr/local/bin/alertmanager/alertmanager.yml

[Install]

WantedBy=multi-user.target

Bước 5: Khởi động và kiểm tra trạng thái của dịch vụ

- sudo service alertmanager start
- sudo service alertmanager status
- sudo systemctl enable alertmanager

Truy cập vào địa chỉ <http://192.168.17.153:9093/#/status>

[Alertmanager](#) [Alerts](#) [Silences](#) [Status](#) [Help](#)

## Status

**Uptime:** 2022-05-19T03:34:20.525Z

## Cluster Status

**Name:** 01G3D62W39Z6VRD53ADCR0RSJA

**Status:** ready

**Peers:**

- **Name:** 01G3D62W39Z6VRD53ADCR0RSJA
- **Address:** 192.168.17.153:9094

### 4.1 Tạo Alerting Rules (Vào lại máy Prometheus Server)

Bước 1: Di chuyển vào thư mục /usr/local/bin/ và tạo file prometheus\_rules.yml

- cd /usr/local/bin/prometheus
- sudo nano prometheus\_rules.yml

groups:

- name: custom\_rules

rules:

- record: node\_memory\_MemFree\_percent

expr: 100 - (100 \* node\_memory\_MemFree\_bytes /  
node\_memory\_MemTotal\_bytes)

- record: node\_filesystem\_free\_percent

expr: 100 \* node\_filesystem\_free\_bytes{mountpoint="/" } /  
node\_filesystem\_size\_bytes{mountpoint="/" }

- Copy thêm rules tại đường link bên dưới

[https://github.com/ducta1911/Prometheus/blob/main/prometheus\\_rules.yml](https://github.com/ducta1911/Prometheus/blob/main/prometheus_rules.yml)

Bước 2: Check rules

./promtool check rules prometheus\_rules.yml

Nếu thấy Success là thành công

Bước 3: Vào Prometheus.yml thêm prometheus\_rules.yml vào phần rule\_files

- sudo nano /usr/local/bin/prometheus/prometheus.yml

rule\_files:

# - "first\_rules.yml"

# - "second\_rules.yml"

- "prometheus\_rules.yml"

Bước 4: Restart lại dịch vụ và kiểm tra trạng thái

- sudo service prometheus restart
- sudo service prometheus status

Vào địa chỉ <http://192.168.17.155:9090/alerts> để kiểm tra cái rules vừa thêm




Prometheus
Alerts
Graph
Status ▾
Help

✓ Inactive (5)
✓ Pending (0)
✓ Firing (0)

Filter by name or labels

/usr/local/bin/prometheus/prometheus\_rules.yml > alert.rules

▼ InstanceDown (0 active)

```

name: InstanceDown
expr: up == 0
for: 1m
labels:
  severity: critical
annotations:
  description: {{ $labels.instance }} of job {{ $labels.job }} has been down for more than 1 minutes.
  summary: Endpoint {{ $labels.instance }} down

```

> HostOutOfMemory (0 active)

> HostOutOfDiskSpace (0 active)

> HostHighCpuLoad (0 active)

> DiskSpaceFree10Percent (0 active)

## 4.2 Cài đặt Alert Manager tích hợp cảnh báo qua Slack

Bước 1:

- `sudo nano /usr/local/bin/alertmanager/alertmanager.yml`

Copy paste file cấu hình ở đường link

<https://github.com/ducta1911/Prometheus/blob/main/alertmanager.yml>

Vào Slack -> Workspace name -> Setting & administration -> Manage apps-> tìm “Incoming WebHooks” -> Edit configurations-> Copy webhook url và channel nhận cảnh báo vào file configs

### Integration Settings

#### Post to Channel

Messages that are sent to the incoming webhook will be posted here.

#prometheusalert

[or create a new channel](#)

#### Webhook URL

Send your JSON payloads to this URL.

[https://hooks.slack.com/services/T005VU103CW/B005FMDU105C0/45B1U7E1A...](#)

Kiểm tra config bằng amtool

- `/usr/local/bin/alertmanager/amtool check-config`  
`/usr/local/bin/alertmanager/alertmanager.yml`

Nếu tất cả ok, khởi động lại dịch vụ

- `sudo service alertmanager restart`
- `sudo service alertmanager status`

Bước 2:

- `sudo nano /usr/local/bin/prometheus/prometheus.yml`

# Alertmanager configuration

alerting:

alertmanagers:

- static\_configs:
  - targets:
    - 192.168.17.153:9093

Kiểm tra config bằng promtool

- `/usr/local/bin/prometheus/promtool check config /usr/local/bin/prometheus/prometheus.yml`
- `sudo service prometheus restart`
- `sudo service prometheus status`

Bước 3: Thêm scrape target trong file Prometheus.yml

- job\_name: 'alert-manager'

static\_configs:

- targets: ['192.168.17.153:9093']

Restart và kiểm tra lại dịch vụ

- `sudo service prometheus restart`
- `sudo service prometheus status`

#### 4.3 Test cảnh báo Instance Down

- Dừng dịch vụ Node exporter

```
root@node1:~#  
root@node1:~# sudo service node-exporter stop  
root@node1:~#
```

- Tab Alert báo về

PrometheusAlertsGraphStatus▼Help

Inactive (4)

Pending (1)

Firing (0)

Filter by name or labels

Show annotations

/usr/local/bin/prometheus/prometheus\_rules.yml > alert.rules

inactivepending (1)

▼ InstanceDown (1 active)

name: InstanceDown

expr: up == 0

for: 1m

labels:

severity: critical

annotations:

description: {{ \$labels.instance }} of job {{ \$labels.job }} has been down for more than 1 minutes.

summary: Endpoint {{ \$labels.instance }} down

Labels	State	Active Since	Value
alertname=InstanceDowninstance=192.168.17.154:9100job=node-exporterseverity=critical	PENDING	2022-05-19T09:57:44.874570689Z	0

PrometheusAlertsGraphStatus▼Help

Inactive (4)

Pending (0)

Firing (1)

Filter by name or labels

Show annotations

/usr/local/bin/prometheus/prometheus\_rules.yml > alert.rules

inactivefiring (1)

▼ InstanceDown (1 active)

name: InstanceDown

expr: up == 0

for: 1m

labels:

severity: critical

annotations:

description: {{ \$labels.instance }} of job {{ \$labels.job }} has been down for more than 1 minutes.

summary: Endpoint {{ \$labels.instance }} down

Labels	State	Active Since	Value
alertname=InstanceDowninstance=192.168.17.154:9100job=node-exporterseverity=critical	FIRING	2022-05-19T09:57:44.874570689Z	0

- Vào Tab Targets thấy node-exporter đã down

PrometheusAlertsGraphStatus▼Help

AllUnhealthyCollapse All

Filter by endpoint or labels

alert-manager (1/1 up)show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://192.168.17.153:9093/metrics	UP	instance="192.168.17.153:9093" job="alert-manager"	11.955s ago	10.742ms	

node-exporter (0/1 up)show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://192.168.17.154:9100/metrics	DOWN	instance="192.168.17.154:9100" job="node-exporter"	13.914s ago	0.974ms	Get "http://192.168.17.154:9100/metrics": dial tcp 192.168.17.154:9100: connect: connection refused

- Qua Slack đã nhận được thông báo node-exporter down

#prometheusalert ▾


+ Add a bookmark

minutes. Details:


- alertname: InstanceDown
- instance: 192.168.5.249:9182

Show more

---

 **Alertmanager** APP 4:58 PM

[FIRING:1] InstanceDown for node-exporter (instance="192.168.17.154:9100", job="node-exporter", severity="critical")

 Graph

Alert details:

Alert: - critical

Description: 192.168.17.154:9100 of job node-exporter has been down for more than 1 minutes. Details:

- alertname: InstanceDown
- instance: 192.168.17.154:9100

Show more

- Bật dịch vụ Node-exporter

```
root@node1:~# sudo service node-exporter start
^[[Aroot@node1
root@node1:~# sudo service node-exporter status
• node-exporter.service - Prometheus Node Exporter Service
  Loaded: loaded (/etc/systemd/system/node-exporter.service; enabled; vendor
  Active: active (running) since Thu 2022-05-19 10:06:20 UTC; 8s ago
  Main PID: 9554 (node_exporter)
  Tasks: 5 (limit: 2237)
  Memory: 6.0M
  CGroup: /system.slice/node-exporter.service
          └─9554 /usr/local/bin/node_exporter

May 19 10:06:20 node1 node_exporter[9554]: ts=2022-05-19T10:06:20.074Z caller=n
May 19 10:06:20 node1 node_exporter[9554]: ts=2022-05-19T10:06:20.074Z caller=n
May 19 10:06:20 node1 node_exporter[9554]: ts=2022-05-19T10:06:20.074Z caller=n
May 19 10:06:20 node1 node_exporter[9554]: ts=2022-05-19T10:06:20.074Z caller=n
```

- Target up

## Targets

All Unhealthy Collapse All



Filter by endpoint or labels

alert-manager (1/1 up) [show less](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
<a href="http://192.168.17.153:9093/metrics">http://192.168.17.153:9093/metrics</a>	UP	<code>instance="192.168.17.153:9093"</code> <code>job="alert-manager"</code>	3.871s ago	5.246ms	

node-exporter (1/1 up) [show less](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
<a href="http://192.168.17.154:9100/metrics">http://192.168.17.154:9100/metrics</a>	UP	<code>instance="192.168.17.154:9100"</code> <code>job="node-exporter"</code>	5.833s ago	25.588ms	

- Slack thông báo node-exporter up

# prometheusalert

+ Add a bookmark

1 minutes. Details:

- alertname: InstanceDown
- instance: 192.168.17.154:9100

[Show more](#)

Alertmanager APP 5:06 PM

[RESOLVED] InstanceDown for node-exporter (instance="192.168.17.154:9100", job="node-exporter", severity="critical")

Graph

Alert details:

Alert: - critical

Description: 192.168.17.154:9100 of job node-exporter has been down for more than 1 minutes. Details:

- alertname: InstanceDown
- instance: 192.168.17.154:9100

[Show more](#)

## 5. Cài đặt Node 2 – Windows exporter

Bước 1: Tải windows exporter bản mới nhất tại

[https://github.com/prometheus-community/windows\\_exporter/releases](https://github.com/prometheus-community/windows_exporter/releases)

**v0.18.1** Latest

## Bug fixes

- Fixed broken whitelist/blacklist flag functionality for **iis** collector #937

### Assets 7

sha256sums.txt	396 Bytes	Feb 05, 2022
windows_exporter-0.18.1-386.exe	17.1 MB	Feb 05, 2022
windows_exporter-0.18.1-386.msi	8.75 MB	Feb 05, 2022
<b>windows_exporter-0.18.1-amd64.exe</b>	18.2 MB	Feb 05, 2022
windows_exporter-0.18.1-amd64.msi	8.96 MB	Feb 05, 2022
Source code (zip)		Feb 02, 2022
Source code (tar.gz)		Feb 02, 2022

## Bước 2: Chạy file vừa tải về

```
C:\Users\mrduc\Downloads\Programs\windows_exporter-0.18.1-amd64.exe
time="2022-05-19T14:35:46+07:00" level=warning msg="No where-clause specified for service collector. This will generate a very large number of metrics!" source="service.go:48"
time="2022-05-19T14:35:46+07:00" level=info msg="Enabled collectors: os, service, system, textfile, cpu, cs, logical_disk, net" source="exporter.go:348"
time="2022-05-19T14:35:46+07:00" level=info msg="Starting windows_exporter (version=0.18.1, branch=heads/tags/v0.18.1, revision=e07b2053af3ca708db3489f41e2fcd9941860f4)" source="exporter.go:400"
time="2022-05-19T14:35:46+07:00" level=info msg="Build context (go=go1.17.6, user=runneradmin@fv-az177-985, date=20220205-08:18:35)" source="exporter.go:401"
time="2022-05-19T14:35:46+07:00" level=info msg="Starting server on :9182" source="exporter.go:404"
time="2022-05-19T14:35:46+07:00" level=info msg="TLS is disabled." source="gokit_adapter.go:38"
time="2022-05-19T14:35:46+07:00" level=error msg="Error reading textfile collector directory \"C:\\\\Users\\\\mrduc\\\\D
ownloads\\\\\\\\Programs\\\\\\\\textfile_inputs\\\\\": open C:\\\\Users\\\\mrduc\\\\Downloads\\\\Programs\\\\textfile_inputs: The system canno
t find the file specified." source="textfile.go:249"
time="2022-05-19T14:35:54+07:00" level=error msg="Error reading textfile collector directory \"C:\\\\Users\\\\mrduc\\\\D
ownloads\\\\\\\\Programs\\\\\\\\textfile_inputs\\\\\": open C:\\\\Users\\\\mrduc\\\\Downloads\\\\Programs\\\\textfile_inputs: The system canno
t find the file specified." source="textfile.go:249"
time="2022-05-19T14:36:04+07:00" level=error msg="Error reading textfile collector directory \"C:\\\\Users\\\\mrduc\\\\D
ownloads\\\\\\\\Programs\\\\\\\\textfile_inputs\\\\\": open C:\\\\Users\\\\mrduc\\\\Downloads\\\\Programs\\\\textfile_inputs: The system canno
t find the file specified." source="textfile.go:249"
time="2022-05-19T14:36:14+07:00" level=error msg="Error reading textfile collector directory \"C:\\\\Users\\\\mrduc\\\\D
ownloads\\\\\\\\Programs\\\\\\\\textfile_inputs\\\\\": open C:\\\\Users\\\\mrduc\\\\Downloads\\\\Programs\\\\textfile_inputs: The system canno
t find the file specified." source="textfile.go:249"
time="2022-05-19T14:36:24+07:00" level=error msg="Error reading textfile collector directory \"C:\\\\Users\\\\mrduc\\\\D
ownloads\\\\\\\\Programs\\\\\\\\textfile_inputs\\\\\": open C:\\\\Users\\\\mrduc\\\\Downloads\\\\Programs\\\\textfile_inputs: The system canno
t find the file specified." source="textfile.go:249"
time="2022-05-19T14:36:34+07:00" level=error msg="Error reading textfile collector directory \"C:\\\\Users\\\\mrduc\\\\D
ownloads\\\\\\\\Programs\\\\\\\\textfile_inputs\\\\\": open C:\\\\Users\\\\mrduc\\\\Downloads\\\\Programs\\\\textfile_inputs: The system canno
t find the file specified." source="textfile.go:249"
time="2022-05-19T14:36:44+07:00" level=error msg="Error reading textfile collector directory \"C:\\\\Users\\\\mrduc\\\\D
ownloads\\\\\\\\Programs\\\\\\\\textfile_inputs\\\\\": open C:\\\\Users\\\\mrduc\\\\Downloads\\\\Programs\\\\textfile_inputs: The system canno
```

Bước 3: Truy cập vào trang <http://192.168.5.249:9182/metrics> sẽ thấy được metric của node 2:

```
← → ↻ ⚠ Not secure | 192.168.5.249:9182/metrics
Instagram Facebook Gaming [...] My Following - Nim... YouTube bit.ly/tiengruoi | Xe... (2) 35. How To Conf... SQL Server Always... Configure Transacti... Roundcube

# HELP go_gc_cycles_automatic_gc_cycles_total Count of completed GC cycles generated by the Go runtime.
# TYPE go_gc_cycles_automatic_gc_cycles_total counter
go_gc_cycles_automatic_gc_cycles_total 1544
# HELP go_gc_cycles_forced_gc_cycles_total Count of completed GC cycles forced by the application.
# TYPE go_gc_cycles_forced_gc_cycles_total counter
go_gc_cycles_forced_gc_cycles_total 0
# HELP go_gc_cycles_total_gc_cycles_total Count of all completed GC cycles.
# TYPE go_gc_cycles_total_gc_cycles_total counter
go_gc_cycles_total_gc_cycles_total 1544
# 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"} 0
go_gc_duration_seconds{quantile="0.25"} 0
go_gc_duration_seconds{quantile="0.5"} 0
go_gc_duration_seconds{quantile="0.75"} 0
go_gc_duration_seconds{quantile="1"} 0.0018685
go_gc_duration_seconds_sum 0.2013205
go_gc_duration_seconds_count 1544
# HELP go_gc_heap_allocs_by_size_bytes_total Distribution of heap allocations by approximate size. Note that this does not include tiny objects as defined by
# TYPE go_gc_heap_allocs_by_size_bytes_total histogram
go_gc_heap_allocs_by_size_bytes_total bucket[le="8.999999999999999"] 2.163269e+06
go_gc_heap_allocs_by_size_bytes_total bucket[le="24.999999999999999"] 8.5395429e+07
go_gc_heap_allocs_by_size_bytes_total bucket[le="64.999999999999999"] 1.2645937e+08
go_gc_heap_allocs_by_size_bytes_total bucket[le="144.999999999999999"] 1.35334495e+08
go_gc_heap_allocs_by_size_bytes_total bucket[le="320.999999999999999"] 1.35552604e+08
go_gc_heap_allocs_by_size_bytes_total bucket[le="704.999999999999999"] 1.35595816e+08
go_gc_heap_allocs_by_size_bytes_total bucket[le="1536.999999999999999"] 1.35615871e+08
go_gc_heap_allocs_by_size_bytes_total bucket[le="3200.999999999999999"] 1.35641119e+08
go_gc_heap_allocs_by_size_bytes_total bucket[le="6528.999999999999999"] 1.35657385e+08
go_gc_heap_allocs_by_size_bytes_total bucket[le="13568.999999999999999"] 1.35667838e+08
go_gc_heap_allocs_by_size_bytes_total bucket[le="27264.999999999999999"] 1.35673542e+08
go_gc_heap_allocs_by_size_bytes_total bucket[le="+Inf"] 1.3568293e+08
go_gc_heap_allocs_by_size_bytes_total sum 6.593493736e+09
go_gc_heap_allocs_by_size_bytes_total count 1.3568293e+08
# HELP go_gc_heap_allocs_bytes_total Cumulative sum of memory allocated to the heap by the application.
# TYPE go_gc_heap_allocs_bytes_total counter
go_gc_heap_allocs_bytes_total 6.593493736e+09
# HELP go_gc_heap_allocs_objects_total Cumulative count of heap allocations triggered by the application. Note that this does not include tiny objects as defined by
# TYPE go_gc_heap_allocs_objects_total counter
```

Bước 4: Vào máy Prometheus Server – Add thêm scrape target cho windows exporter

scrape\_configs:

```
# ...
```

```
- job_name: 'node_2'
```

```
  scrape_interval: 10s
```

```
  static_configs:
```


```
    - targets: ['192.168.5.249:9182']
```

Bước 5: Khởi động lại dịch vụ

- `sudo service prometheus restart`

Bước 6: Query metric trên Prometheus Server

Truy cập vào <http://192.168.17.155:9090/graph> sử dụng câu query cơ bản  
{instance="192.168.5.249:9182", job="node\_2"}

 Prometheus Alerts Graph Status ▾ Help


☐ Use local time

☐ Enable query history


☒ Enable autocomplete

☒ Enable highlighting

☒ Enable linter



{instance="192.168.5.249:9182", job="node\_2"}

 Execute

Table

Graph

Load time: 566ms Resolution: 14s Result series: 7747

<

Evaluation time

>

Notice: Showing more than 1000 series, turning off label formatting for performance reasons.

go_gc_cycles_automatic_gc_cycles_total(instance="192.168.5.249:9182", job="node_2")	1601
go_gc_cycles_forced_gc_cycles_total(instance="192.168.5.249:9182", job="node_2")	0
go_gc_cycles_total_gc_cycles_total(instance="192.168.5.249:9182", job="node_2")	1601
go_gc_duration_seconds(instance="192.168.5.249:9182", job="node_2", quantile="0")	0
go_gc_duration_seconds(instance="192.168.5.249:9182", job="node_2", quantile="0.25")	0
go_gc_duration_seconds(instance="192.168.5.249:9182", job="node_2", quantile="0.5")	0
go_gc_duration_seconds(instance="192.168.5.249:9182", job="node_2", quantile="0.75")	0
go_gc_duration_seconds(instance="192.168.5.249:9182", job="node_2", quantile="1")	0.0019992
go_gc_duration_seconds_count(instance="192.168.5.249:9182", job="node_2")	1601
go_gc_duration_seconds_sum(instance="192.168.5.249:9182", job="node_2")	0.3022852

Như vậy Prometheus server đã thực hiện scrape và lưu trữ metrics thành công metric từ Node 2 Windows.