

SISTEM MONITORING RESOURCE VIRTUAL MACHINE MENGGUNAKAN NODE EXPORTER, PROMETHEUS & GRAFANA

Diajukan untuk Memenuhi Tugas Mata Kuliah

”Workshop Administrasi Jaringan”

Dosen: Ferry Astika



Oleh :

Muhamad Aldy Nugroho (3122600020)

Wildan Ardana (3122600028)

Daniar Oktavian Dwiputra (3122600020)

2 D4 Teknik Informatika A

Semester 4

D4 Teknik Informatika

Departemen Teknik Informatika dan Komputer

Politeknik Elektronika Negeri Surabaya

A. ABSTRAKSI

Pemantauan infrastruktur merupakan aspek penting dalam memastikan kelancaran operasi dan kinerja sistem. Dalam era virtualisasi, pemantauan VM (Virtual Machine) menjadi semakin krusial untuk memastikan ketersediaan, kinerja, dan efisiensi sumber daya. Makalah ini menghadirkan solusi pemantauan VM yang efektif dengan memanfaatkan Prometheus dan Grafana.

Prometheus, sistem pengumpulan data time-series, berperan dalam mengumpulkan metrik penting dari VM, seperti penggunaan CPU, memori, disk, dan jaringan. Metrik-metrik ini kemudian disimpan dalam database Prometheus untuk analisis dan visualisasi. Grafana, platform visualisasi data, berperan dalam menampilkan metrik-metrik yang dikumpulkan Prometheus dalam bentuk dashboard yang informatif dan mudah dipahami.

Penggunaan Prometheus dan Grafana bersama-sama memungkinkan pemantauan VM yang komprehensif dan real-time. Dashboard Grafana dapat disesuaikan untuk menampilkan metrik-metrik yang relevan dengan kebutuhan spesifik, seperti pemantauan kinerja aplikasi, pemantauan kesehatan sistem, dan identifikasi potensi bottleneck.

Makalah ini membahas secara mendalam arsitektur dan implementasi solusi pemantauan VM menggunakan Prometheus dan Grafana. Manfaat dan keunggulan solusi ini dibahas, termasuk skalabilitas, fleksibilitas, dan kemudahan penggunaan.

B. PENDAHULUAN

Prometheus dan Grafana adalah dua alat yang sering digunakan untuk kebutuhan monitoring dan visualisasi data. Prometheus adalah sistem monitoring open-source yang mengumpulkan dan menyimpan metrik dari berbagai sumber dalam format time-series. Metrik ini sangat penting untuk memantau performa, kesehatan, dan ketersediaan sistem. Sementara itu, Grafana adalah platform visualisasi yang memungkinkan pengguna untuk membuat dashboard interaktif, sehingga memudahkan dalam memantau dan menganalisis data yang dikumpulkan oleh Prometheus.

Implementasi monitoring dengan Prometheus dan Grafana pada Virtual Machine (VM) memberikan beberapa keuntungan utama, yaitu:

1. **Deteksi Masalah:** Dengan monitoring yang baik, masalah dalam sistem dapat terdeteksi lebih awal sebelum menjadi gangguan yang lebih besar. Prometheus dapat mengirimkan notifikasi ketika metrik tertentu melewati batas yang ditentukan, memungkinkan tim untuk segera mengambil tindakan korektif.
2. **Visualisasi Data:** Grafana menyediakan visualisasi data yang intuitif dan interaktif melalui berbagai jenis grafik dan dashboard. Ini memudahkan pengguna untuk

memahami data yang dikumpulkan, melihat tren, dan melakukan analisis lebih mendalam.

3. **Open-Source dan Skalabilitas:** Kedua alat ini bersifat open-source, yang berarti dapat digunakan dan dimodifikasi sesuai kebutuhan tanpa biaya lisensi. Selain itu, baik Prometheus maupun Grafana dirancang untuk skalabilitas, sehingga dapat digunakan untuk memantau infrastruktur dari skala kecil hingga besar.
4. **Visibilitas:** Implementasi monitoring ini memberikan visibilitas penuh terhadap performa dan kesehatan VM. Dengan dashboard yang dapat dikustomisasi, pengguna dapat memantau metrik penting secara real-time, sehingga memiliki pemahaman yang lebih baik tentang kondisi sistem mereka.

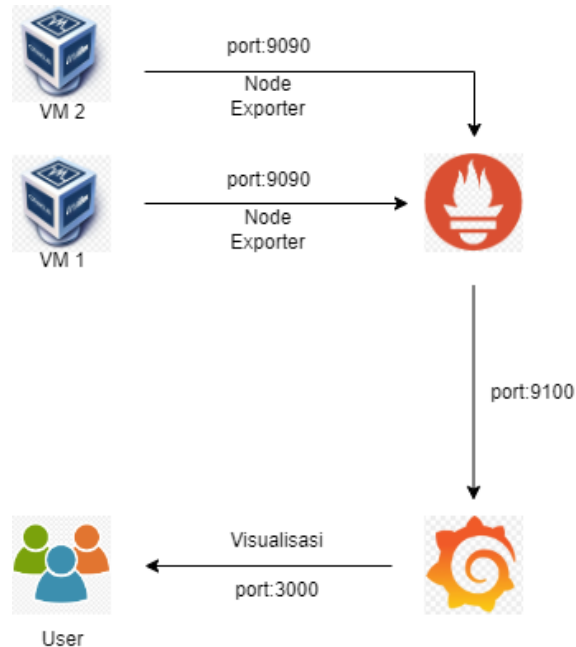
Dengan menggunakan Prometheus dan Grafana, organisasi dapat meningkatkan keandalan dan efisiensi infrastruktur mereka, memastikan bahwa sistem berjalan optimal dan siap menangani permintaan pengguna.

C. RUANG LINGKUP

1. Monitoring VM
 - Proyek ini berfokus pada monitoring Virtual Machines (VM) yang berjalan di VirtualBox.
2. Komponen Monitoring
 - Prometheus: Digunakan untuk mengumpulkan dan menyimpan data metrik dari VM.
 - Node Exporter: Diinstal di setiap VM untuk mengekspor metrik sistem seperti CPU, memori, disk, dan jaringan.
 - Grafana: Digunakan untuk visualisasi data metrik yang dikumpulkan oleh Prometheus.
3. Monitoring
 - CPU Usage: Memantau persentase penggunaan CPU.
 - Memory Usage: Memantau penggunaan RAM dan SWAP.
 - Disk Usage: Memantau penggunaan filesystem root.
4. Visualisasi dan Dashboard
 - Membuat dashboard di Grafana dengan berbagai panel untuk menampilkan metrik yang dikumpulkan.
 - Menyediakan visualisasi metrik seperti penggunaan CPU, memori, disk, dan beban sistem secara real-time.
5. Tujuan Proyek
 - Memastikan bahwa semua VM terpantau dengan baik.

- Mengidentifikasi potensi masalah kinerja atau beban pada VM.

D. DESAIN SISTEM



1. Virtual Machine (Debian)
 - Sistem operasi Debian berjalan pada mesin virtual. Mesin ini adalah target yang akan dimonitor.
2. Prometheus
 - Prometheus berfungsi sebagai alat pengumpul data (data collection). Prometheus diinstal pada mesin virtual atau server terpisah.
 - Prometheus dikonfigurasi untuk mengumpulkan metrik dari mesin virtual melalui port 9090.
3. Node Exporter
 - Node Exporter adalah agen yang berjalan pada mesin virtual Debian. Ia mengumpulkan data metrik seperti penggunaan CPU dan RAM dan mengirimkannya ke Prometheus melalui port 9100.
4. Grafana
 - Grafana adalah alat visualisasi data yang mengambil data dari Prometheus
 - Dengan Grafana, administrator jaringan dapat membuat dashboard yang menampilkan metrik penggunaan RAM dan CPU secara real-time.

Alur Kerja Sistem

1. Pengumpulan Data oleh Node Exporter
 - Node Exporter mengumpulkan data metrik dari mesin virtual Debian, termasuk informasi tentang penggunaan RAM dan CPU.
 - Data metrik ini kemudian dikirimkan ke Prometheus melalui port 9100.
2. Pengumpulan dan Penyimpanan Data oleh Prometheus
 - Prometheus mengumpulkan data metrik dari Node Exporter pada interval waktu yang telah ditentukan.
 - Data yang dikumpulkan disimpan dalam database internal Prometheus.
3. Visualisasi Data oleh Grafana
 - Grafana mengambil data metrik yang telah dikumpulkan oleh Prometheus.
 - Administrator dapat membuat berbagai dashboard dan grafik untuk memvisualisasikan data penggunaan RAM dan CPU secara real-time.

E. TAHAPAN PELAKSANAAN

Nomor	Tanggal	Keterangan
1.	Senin, 27 Mei 2024	Membuat project charter
2.	Selasa, 28 Mei 2024	Instalansi Prometheus dan Node Exporter
3.	Kamis, 30 Mei 2024	Instalansi Grafana
4.	Jumat, 31 Mei 2024	Mengimplementasikan monitoring dengan Prometheus dan Grafana Mengamati hasil monitoring
5.	Minggu, 2 Juni 2024	Membuat laporan akhir

F. SISTEM TESTING

Pengujian Prometheus

1. Konfigurasi Target VM di Prometheus
 - Tambahkan konfigurasi target VM dalam file ``prometheus.yml`` pada bagian ``scrape_configs``. Contoh:

```
global:
  scrape_interval: 15s
  evaluation_interval: 15s

scrape_configs:
  - job_name: "prometheus"
    static_configs:
      - targets: ["localhost:9090"]
  - job_name: "node-exporter"
    static_configs:
      - targets: ["localhost:9100"]
```

2. Verifikasi Instalasi dan Konfigurasi Prometheus
 - Setelah melakukan konfigurasi, restart Prometheus:
`sudo systemctl status prometheus`
 - Buka browser dan akses Prometheus melalui **`http://localhost:9090`**. Jika berhasil, akan terbuka jendela Prometheus.

Verifikasi Pengumpulan Metrik dari VM

1. Buka Halaman Prometheus:
 - Akses Prometheus di **`http://localhost:9090`**
2. Verifikasi Status Target:
 - Klik tab Status > Targets.
 - Pastikan target VM muncul di daftar dan statusnya UP. Ini menandakan bahwa Prometheus berhasil mengumpulkan metrik dari VM.

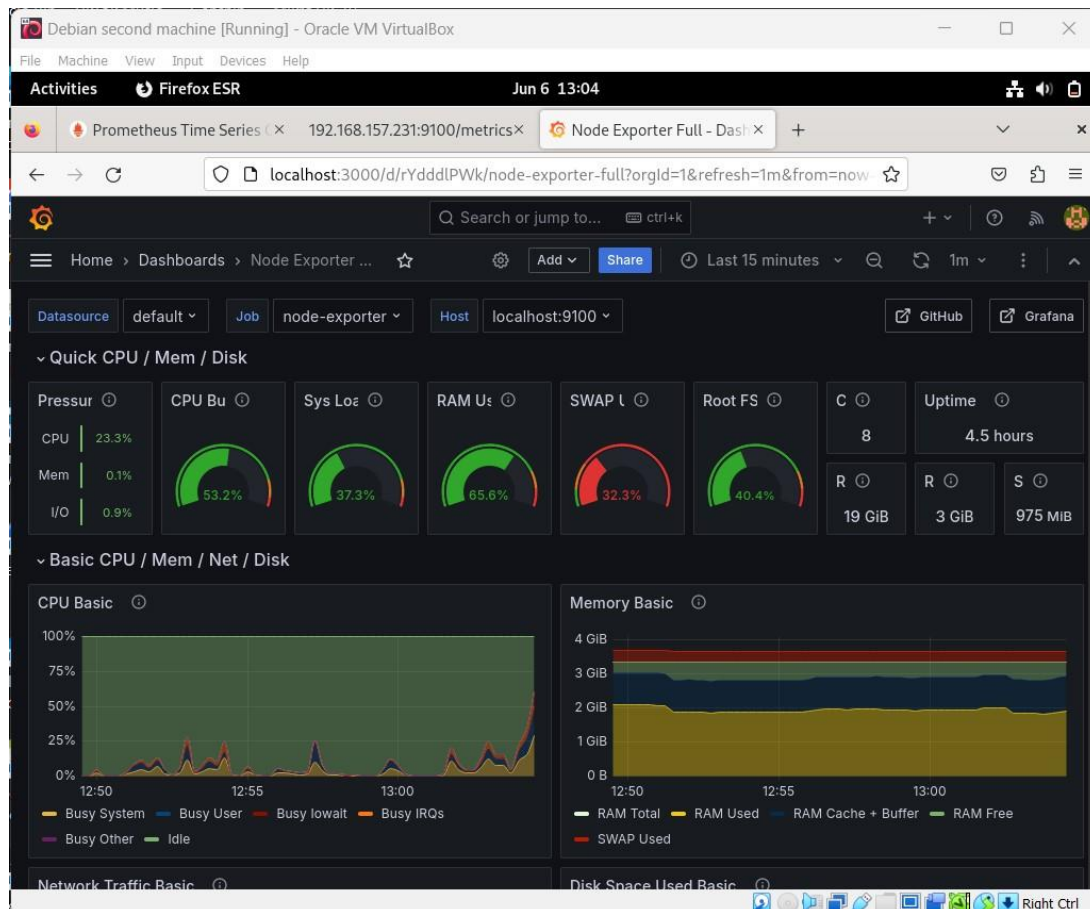
Setup Grafana untuk Visualisasi Data

1. Akses Grafana:
 - Setelah instalasi dan konfigurasi, akses Grafana melalui <http://localhost:3000>

2. Tambahkan Prometheus sebagai Data Source:
 - Masuk ke Grafana.
 - Navigasi ke **Configuration > Data Sources**.
 - Klik **Add data source** dan pilih **Prometheus**.
 - Isi URL Prometheus, misalnya **http://localhost:9090**, dan klik **Save & Test** untuk memastikan koneksi berhasil.
3. Buat Dashboard Baru untuk Metrik VM:
 - Masuk ke **Create > Dashboard**.
 - Tambahkan panel baru dan pilih data source Prometheus.

Verifikasi Visualisasi Data di Grafana

1. Periksa Metrik di Dashboard Grafana:
 - Pastikan metrik dari VM muncul di dashboard.
 - Verifikasi bahwa data yang ditampilkan sesuai dengan metrik yang diharapkan, seperti penggunaan CPU, memori, disk I/O, dan lain-lain.

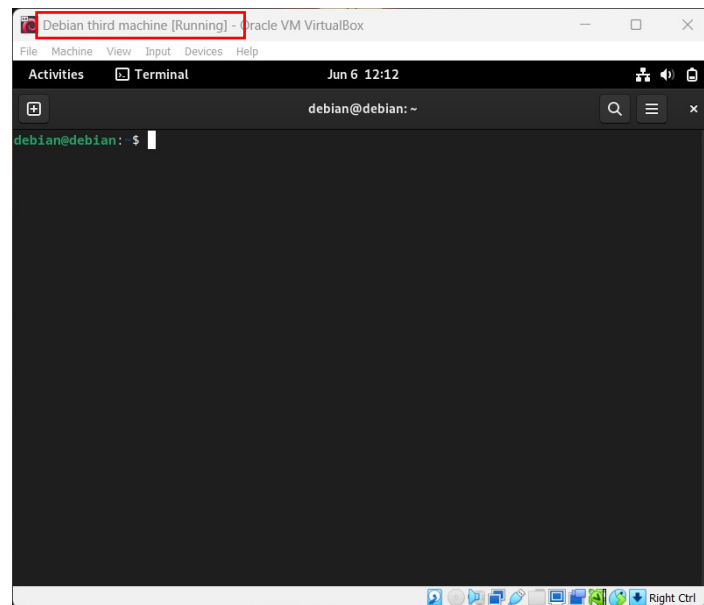


Tambahan :

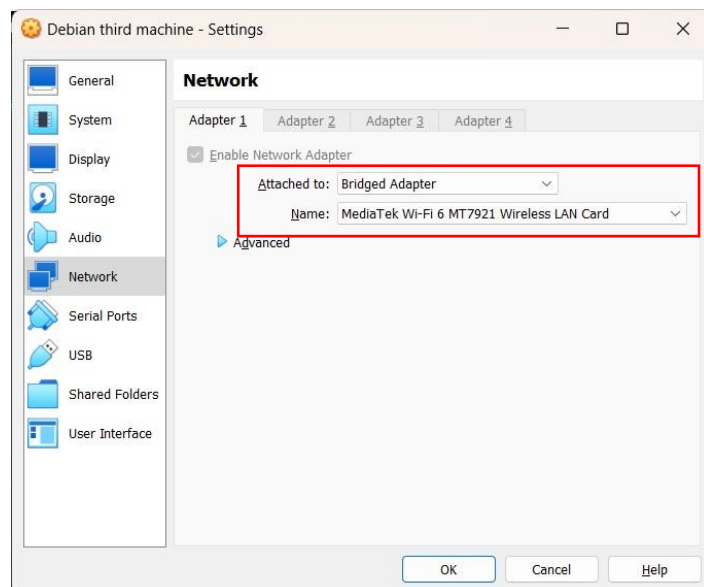
1. Tambah 1 Virtual Machine lagi untuk dimonitor
2. Darimana Node Exporter mendapatkan informasi system
3. Arti dari CPU Busy System, CPU Busy User, CPU Busy Lowait, CPU Busy IRQs. CPU Busy Other
4. Ubah grafik menjadi Line tanpa fill

1. Tambah 1 Virtual Machine lagi untuk dimonitor

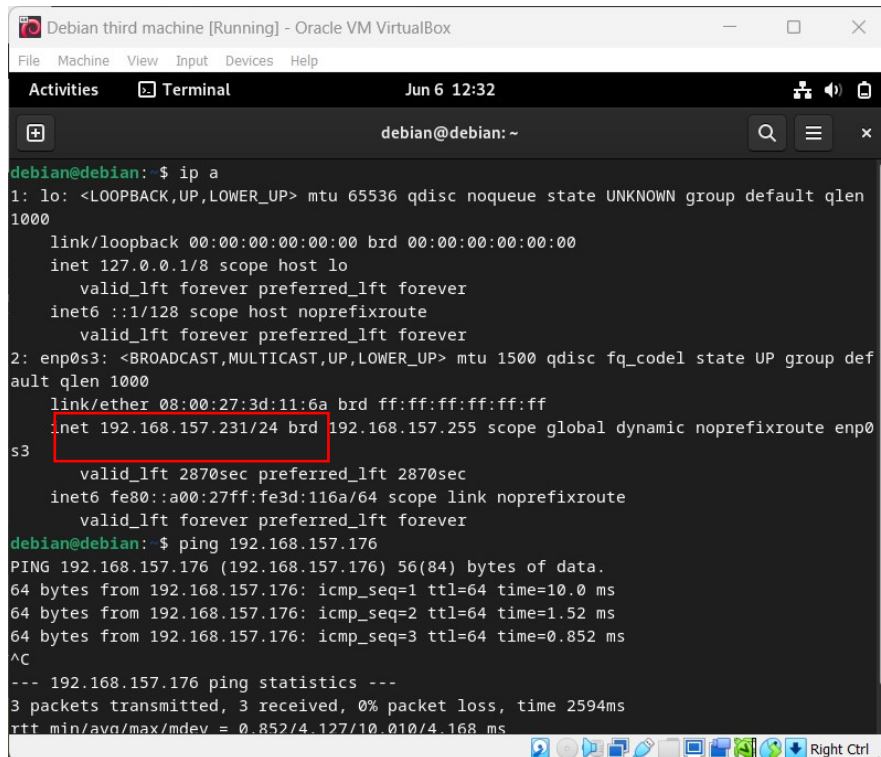
- a. Buat 1 buah Virtual Machine lagi (kali ini saya beri nama `debian third machine`)



- b. Ubah network settings menjadi **Bridged adapter**

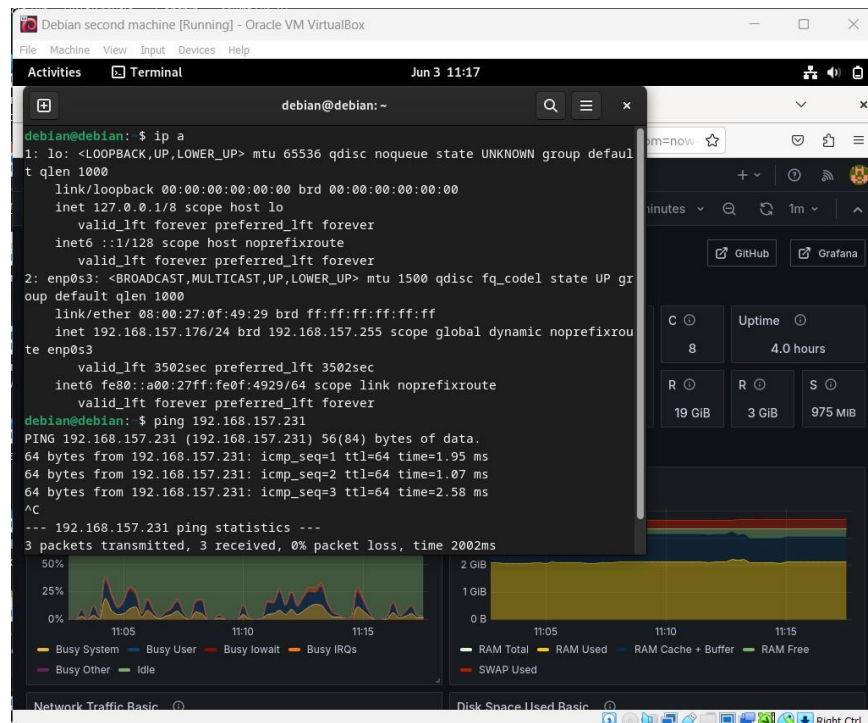


- c. Cek apakah VM yang baru tadi sudah mendapatkan ip address menggunakan command ``ip a``

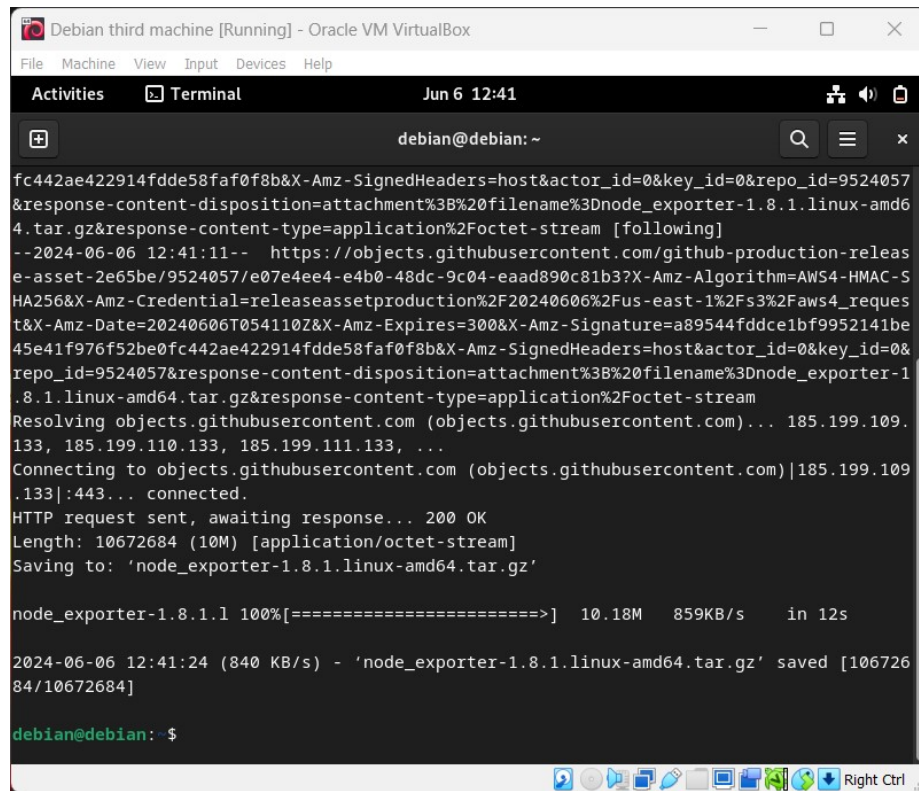


```
Debian third machine [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Jun 6 12:32
debian@debian: ~
debian@debian:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:3d:11:6a brd ff:ff:ff:ff:ff:ff
    inet 192.168.157.231/24 brd 192.168.157.255 scope global dynamic noprefixroute enp0s3
        valid_lft 2870sec preferred_lft 2870sec
    inet6 fe80::a00:27ff:fe3d:116a/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
debian@debian:~$ ping 192.168.157.176
PING 192.168.157.176 (192.168.157.176) 56(84) bytes of data:
64 bytes from 192.168.157.176: icmp_seq=1 ttl=64 time=10.0 ms
64 bytes from 192.168.157.176: icmp_seq=2 ttl=64 time=1.52 ms
64 bytes from 192.168.157.176: icmp_seq=3 ttl=64 time=0.852 ms
^C
--- 192.168.157.176 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2594ms
rtt min/avg/max/mdev = 0.852/4.127/10.010/4.168 ms
```

- d. Jika sudah dapat ip address, coba lakukan ping antar VM untuk memastikan kedua VM bisa saling berkomunikasi



- e. Install node exporter pada VM kedua melalui web resmi Prometheus



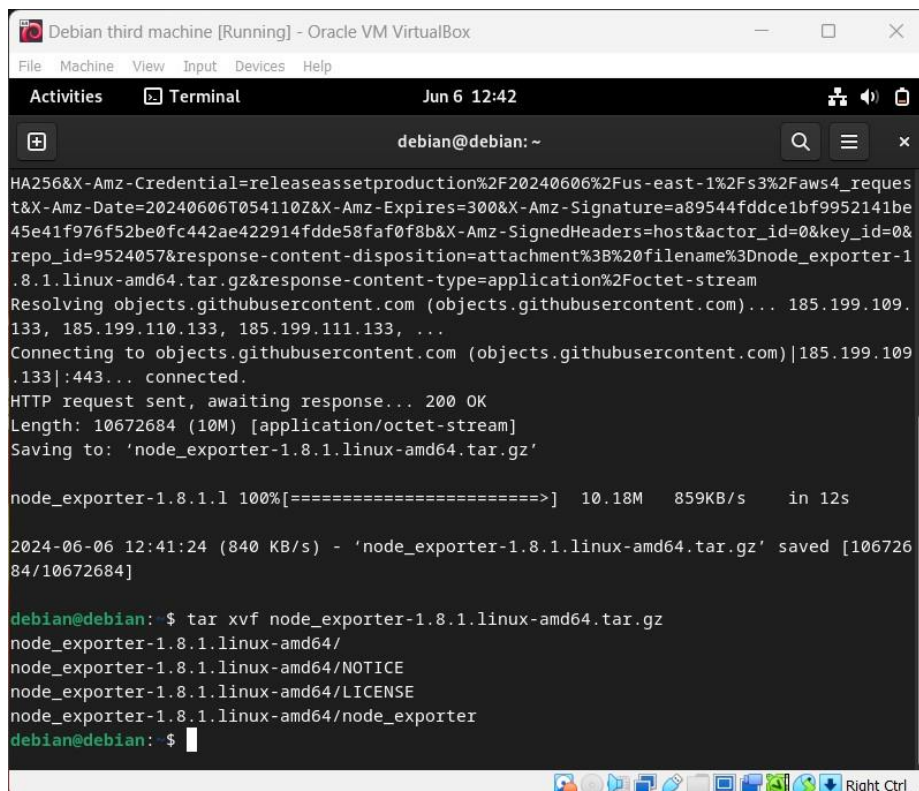
```
fc442ae422914fdde58faf0f8b&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=9524057
&response-content-disposition=attachment%3B%20filename%3Dnode_exporter-1.8.1.linux-amd6
4.tar.gz&response-content-type=application%2Foctet-stream [following]
--2024-06-06 12:41:11-- https://objects.githubusercontent.com/github-production-releas
e-asset-2e65be/9524057/e07e4ee4-e4b0-48dc-9c04-eaad890c81b3?X-Amz-Algorithm=AWS4-HMAC-S
HA256&X-Amz-Credential=releaseassetproduction%2F20240606%2Fus-east-1%2Fs3%2Faws4_reques
t&X-Amz-Date=20240606T054110Z&X-Amz-Expires=300&X-Amz-Signature=a89544fddce1bf9952141be
45e41f976f52be0fc442ae422914fdde58faf0f8b&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&
repo_id=9524057&response-content-disposition=attachment%3B%20filename%3Dnode_exporter-1
.8.1.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.109.
133, 185.199.110.133, 185.199.111.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.109
.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10672684 (10M) [application/octet-stream]
Saving to: 'node_exporter-1.8.1.linux-amd64.tar.gz'

node_exporter-1.8.1.l 100%[=====] 10.18M 859KB/s in 12s

2024-06-06 12:41:24 (840 KB/s) - 'node_exporter-1.8.1.linux-amd64.tar.gz' saved [106726
84/10672684]

debian@debian:~$
```

- f. Extract file node exporter yang sudah di download tadi, gunakan `tar xvf`



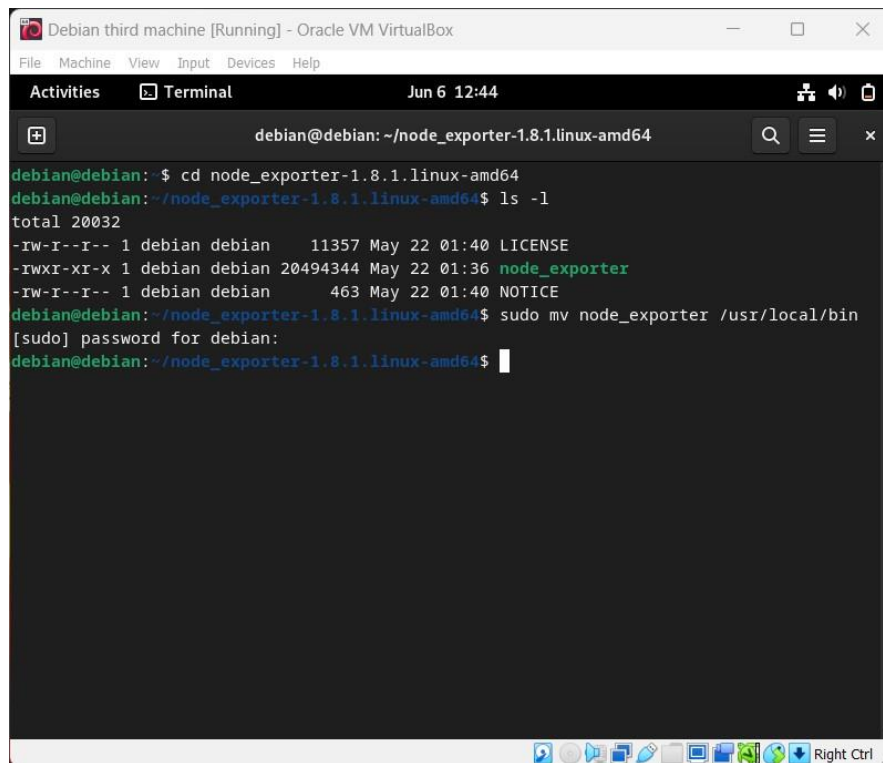
```
HA256&X-Amz-Credential=releaseassetproduction%2F20240606%2Fus-east-1%2Fs3%2Faws4_reques
t&X-Amz-Date=20240606T054110Z&X-Amz-Expires=300&X-Amz-Signature=a89544fddce1bf9952141be
45e41f976f52be0fc442ae422914fdde58faf0f8b&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&
repo_id=9524057&response-content-disposition=attachment%3B%20filename%3Dnode_exporter-1
.8.1.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.109.
133, 185.199.110.133, 185.199.111.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.109
.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10672684 (10M) [application/octet-stream]
Saving to: 'node_exporter-1.8.1.linux-amd64.tar.gz'

node_exporter-1.8.1.l 100%[=====] 10.18M 859KB/s in 12s

2024-06-06 12:41:24 (840 KB/s) - 'node_exporter-1.8.1.linux-amd64.tar.gz' saved [106726
84/10672684]

debian@debian:~$ tar xvf node_exporter-1.8.1.linux-amd64.tar.gz
node_exporter-1.8.1.linux-amd64/
node_exporter-1.8.1.linux-amd64/NOTICE
node_exporter-1.8.1.linux-amd64/LICENSE
node_exporter-1.8.1.linux-amd64/node_exporter
debian@debian:~$
```

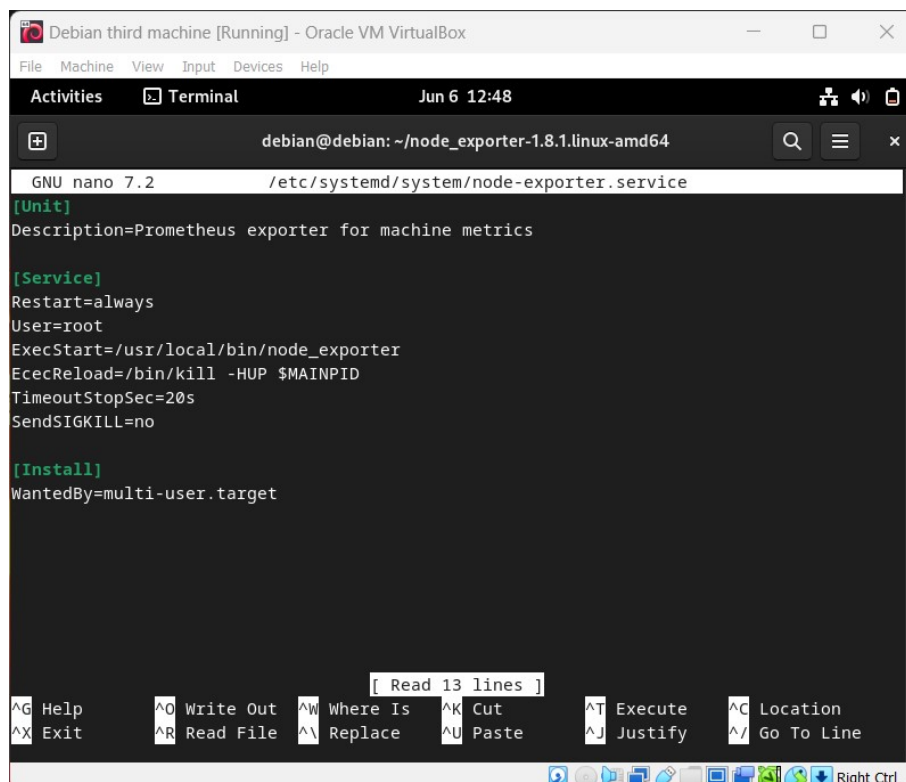
- g. Pindah ke direktori hasil extract node exporter dan pindahkan `node_exporter` ke `/usr/local/bin` untuk mempermudah organisasi



```
Debian third machine [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Jun 6 12:44
debian@debian: ~/node_exporter-1.8.1.linux-amd64

debian@debian:~$ cd node_exporter-1.8.1.linux-amd64
debian@debian:~/node_exporter-1.8.1.linux-amd64$ ls -l
total 20032
-rw-r--r-- 1 debian debian 11357 May 22 01:40 LICENSE
-rwxr-xr-x 1 debian debian 20494344 May 22 01:36 node_exporter
-rw-r--r-- 1 debian debian 463 May 22 01:40 NOTICE
debian@debian:~/node_exporter-1.8.1.linux-amd64$ sudo mv node_exporter /usr/local/bin
[sudo] password for debian:
debian@debian:~/node_exporter-1.8.1.linux-amd64$
```

- h. Buat service baru di `/etc/systemd/system/node-exporter.service`



```
Debian third machine [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Jun 6 12:48
debian@debian: ~/node_exporter-1.8.1.linux-amd64

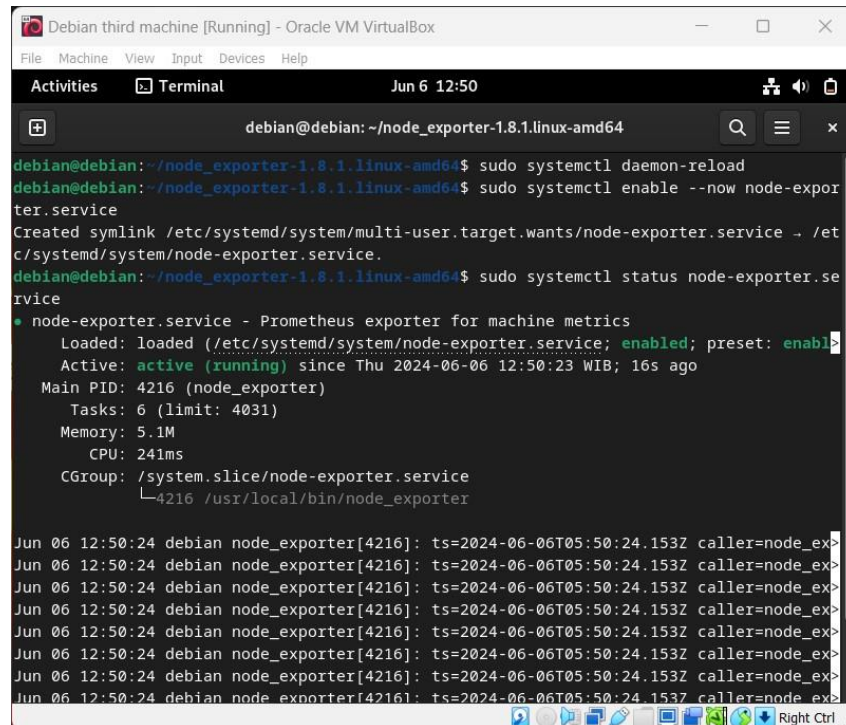
GNU nano 7.2 /etc/systemd/system/node-exporter.service
[Unit]
Description=Prometheus exporter for machine metrics

[Service]
Restart=always
User=root
ExecStart=/usr/local/bin/node_exporter
ExecReload=/bin/kill -HUP $MAINPID
TimeoutStopSec=20s
SendSIGKILL=no

[Install]
WantedBy=multi-user.target

[ Read 13 lines ]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line
```

- i. Reload daemon, lalu nyalakan service yang sudah kita buat tadi, dan cek status apakah service sudah menyala

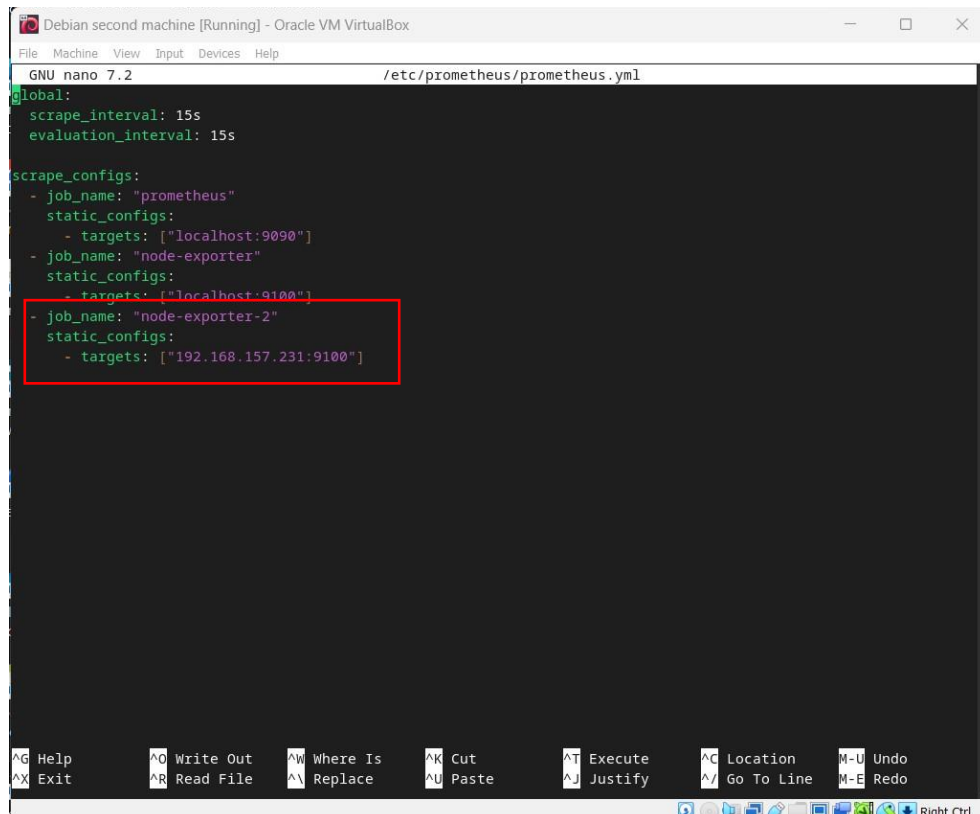


```
Debian third machine [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Jun 6 12:50
debian@debian: ~/node_exporter-1.8.1.linux-amd64

debian@debian:~/node_exporter-1.8.1.linux-amd64$ sudo systemctl daemon-reload
debian@debian:~/node_exporter-1.8.1.linux-amd64$ sudo systemctl enable --now node-exporter.service
Created symlink /etc/systemd/system/multi-user.target.wants/node-exporter.service - /etc/systemd/system/node-exporter.service.
debian@debian:~/node_exporter-1.8.1.linux-amd64$ sudo systemctl status node-exporter.service
* node-exporter.service - Prometheus exporter for machine metrics
   Loaded: loaded (/etc/systemd/system/node-exporter.service; enabled; preset: enabled)
   Active: active (running) since Thu 2024-06-06 12:50:23 WIB; 16s ago
     Main PID: 4216 (node_exporter)
        Tasks: 6 (limit: 4031)
       Memory: 5.1M
          CPU: 241ms
      CGroup: /system.slice/node-exporter.service
              └─4216 /usr/local/bin/node_exporter

Jun 06 12:50:24 debian node_exporter[4216]: ts=2024-06-06T05:50:24.153Z caller=node_exporter
Jun 06 12:50:24 debian node_exporter[4216]: ts=2024-06-06T05:50:24.153Z caller=node_exporter
Jun 06 12:50:24 debian node_exporter[4216]: ts=2024-06-06T05:50:24.153Z caller=node_exporter
Jun 06 12:50:24 debian node_exporter[4216]: ts=2024-06-06T05:50:24.153Z caller=node_exporter
Jun 06 12:50:24 debian node_exporter[4216]: ts=2024-06-06T05:50:24.153Z caller=node_exporter
Jun 06 12:50:24 debian node_exporter[4216]: ts=2024-06-06T05:50:24.153Z caller=node_exporter
Jun 06 12:50:24 debian node_exporter[4216]: ts=2024-06-06T05:50:24.153Z caller=node_exporter
Jun 06 12:50:24 debian node_exporter[4216]: ts=2024-06-06T05:50:24.153Z caller=node_exporter
```

- j. Pindah ke VM 1 lalu tambahkan job baru (node-exporter-2) di **prometheus.yml**



```
Debian second machine [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
GNU nano 7.2 /etc/prometheus/prometheus.yml
global:
  scrape_interval: 15s
  evaluation_interval: 15s

scrape_configs:
  - job_name: "prometheus"
    static_configs:
      - targets: ["localhost:9090"]
  - job_name: "node-exporter"
    static_configs:
      - targets: ["localhost:9100"]
  - job_name: "node-exporter-2"
    static_configs:
      - targets: ["192.168.157.231:9100"]

AG Help      AQ Write Out  AW Where Is  AK Cut       AT Execute   AC Location  MU Undo
AX Exit      AR Read File  AL Replace   AU Paste     AJ Justify   AV Go To Line ME Redo
```


- k. Restart service prometheus pada VM 1 dan cek status nya

```
Debian second machine [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
debian@debian: ~$ sudo systemctl restart prometheus.service
debian@debian: ~$ sudo systemctl status prometheus.service
● prometheus.service - Prometheus
   Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; preset: enabled)
   Active: active (running) since Thu 2024-06-06 13:02:23 WIB; 5s ago
     Docs: https://prometheus.io/docs/introduction/overview/
    Main PID: 10686 (prometheus)
      Tasks: 13 (limit: 4046)
     Memory: 59.7M
        CPU: 2.797s
    CGroup: /system.slice/prometheus.service
           └─10686 /usr/local/bin/prometheus --config.file /etc/prometheus/prometheus.yml --storage.tsdb.path /var/lib/prometheus/

Jun 06 13:02:25 debian prometheus[10686]: ts=2024-06-06T06:02:25.986Z caller=head.go:783 level=info component=tsdb
Jun 06 13:02:26 debian prometheus[10686]: ts=2024-06-06T06:02:26.058Z caller=head.go:783 level=info component=tsdb
Jun 06 13:02:26 debian prometheus[10686]: ts=2024-06-06T06:02:26.059Z caller=head.go:783 level=info component=tsdb
Jun 06 13:02:26 debian prometheus[10686]: ts=2024-06-06T06:02:26.059Z caller=head.go:820 level=info component=tsdb
Jun 06 13:02:26 debian prometheus[10686]: ts=2024-06-06T06:02:26.066Z caller=main.go:1150 level=info fs_type=Ext4
Jun 06 13:02:26 debian prometheus[10686]: ts=2024-06-06T06:02:26.066Z caller=main.go:1153 level=info msg="TSDB init complete"
Jun 06 13:02:26 debian prometheus[10686]: ts=2024-06-06T06:02:26.066Z caller=main.go:1335 level=info msg="Loading configuration file"
Jun 06 13:02:26 debian prometheus[10686]: ts=2024-06-06T06:02:26.077Z caller=main.go:1372 level=info msg="Completed loading of configuration file"
Jun 06 13:02:26 debian prometheus[10686]: ts=2024-06-06T06:02:26.078Z caller=main.go:1114 level=info msg="Server is ready to receive connections."
Jun 06 13:02:26 debian prometheus[10686]: ts=2024-06-06T06:02:26.078Z caller=manager.go:163 level=info component=manager

debian@debian: ~$
```

- l. Cek apakah job baru (node-exporter-2) sudah bisa diakses dari Prometheus di VM 1

Debian second machine [Running] - Oracle VM VirtualBox

Activities Firefox ESR Jun 6 13:03

Prometheus Time Series 192.168.157.231:9100/metrics Node Exporter Full - Dasl

localhost:9090/targets?search=

Prometheus Alerts Graph Status Help

Targets

All scrape pools All Unhealthy Collapse All Filter by endpoint or label Unknown Unhealthy Healthy

node-exporter (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9100/metrics	UP	instance="localhost:9100" job="node-exporter"	6.518s ago	55.243ms	

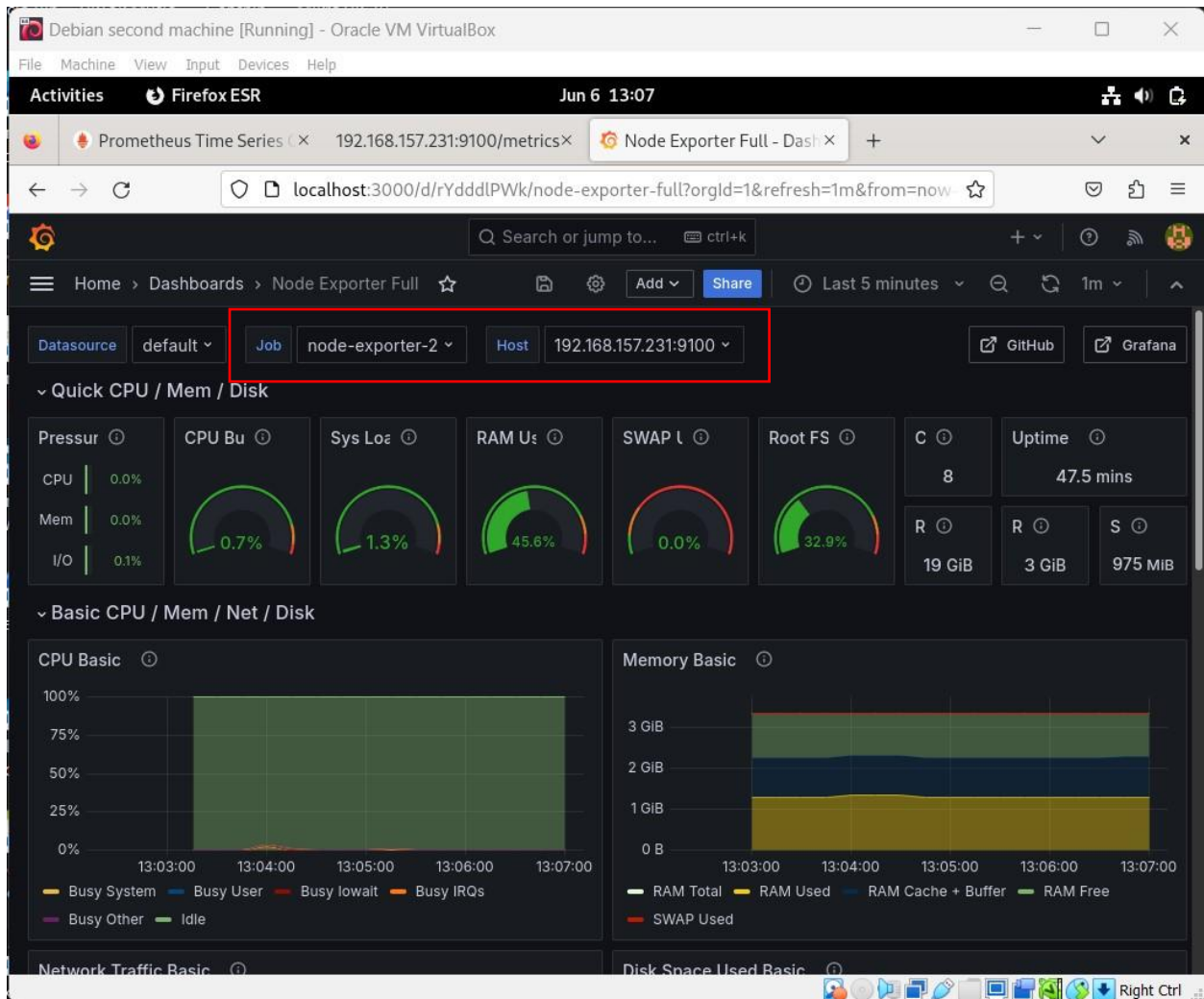
node-exporter-2 (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://192.168.157.231:9100/metrics	UP	instance="192.168.157.231:9100" job="node-exporter-2"	5.134s ago	139.492ms	

prometheus (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090"	7.005s ago	0.722ms	

m. Ubah job pada visualisasi Grafana di VM 1 ke job (node-exporter-2) VM 2



2. Darimana Node Exporter mendapatkan informasi system?

Node Exporter banyak menggunakan file sistem virtual di /proc dan /sys, yang menyediakan informasi runtime tentang sistem operasi dan perangkat keras. Beberapa contoh:

- **/proc/meminfo:** Memberikan informasi tentang penggunaan memori.
- **/proc/cpuinfo:** Menyediakan detail tentang CPU.
- **/proc/stat:** Menyediakan statistik tentang penggunaan CPU.
- **/proc/diskstats:** Menyediakan statistik tentang penggunaan disk.
- **/proc/net/dev:** Memberikan informasi tentang statistik jaringan.
- **/sys/class/net/:** Memberikan informasi tentang interface jaringan.

3. Arti dari CPU Busy System, CPU Busy User, CPU Busy Lowait, CPU Busy IRQs. CPU Busy Other

Dalam konteks monitoring sistem dan visualisasi menggunakan Grafana dengan data yang mungkin dikumpulkan oleh Node Exporter, metrik CPU Busy biasanya menunjukkan persentase waktu yang dihabiskan CPU dalam berbagai keadaan. Berikut adalah penjelasan tentang setiap metrik yang disebutkan:

CPU Busy System

- **CPU Busy System** mengindikasikan waktu yang dihabiskan CPU untuk menjalankan tugas-tugas kernel atau sistem operasi. Ini mencakup waktu yang dihabiskan oleh CPU untuk menangani panggilan sistem dan tugas-tugas yang membutuhkan interaksi dengan perangkat keras melalui kernel.

CPU Busy User

- **CPU Busy User** mengindikasikan waktu yang dihabiskan CPU untuk menjalankan proses-proses pengguna di tingkat user space. Ini mencakup waktu yang dihabiskan untuk menjalankan aplikasi dan proses non-kernel.

CPU Busy IOWait

- **CPU Busy IOWait** mengindikasikan waktu yang dihabiskan CPU menunggu operasi I/O (Input/Output) untuk selesai. Ini biasanya terjadi ketika CPU menunggu untuk membaca atau menulis data ke disk atau perangkat jaringan. Selama waktu ini, CPU tidak aktif memproses tugas lain tetapi tetap dianggap "sibuk" karena menunggu I/O.

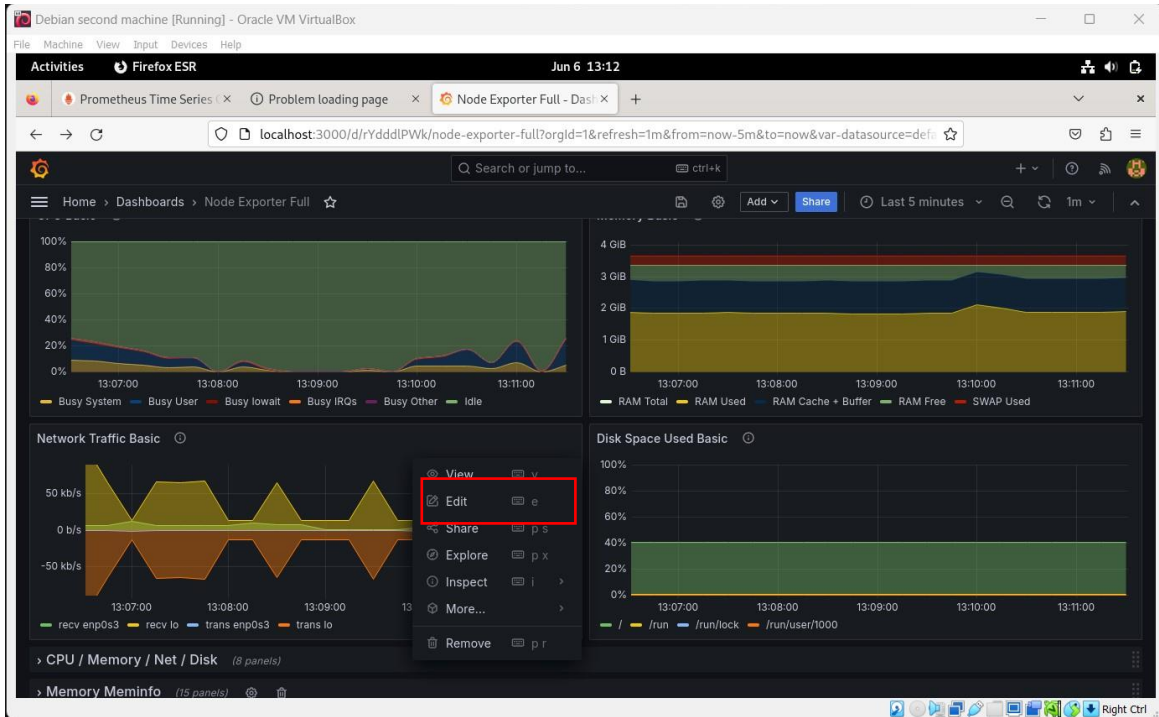
CPU Busy IRQs

- **CPU Busy IRQs** (Interrupt Requests) mengindikasikan waktu yang dihabiskan CPU menangani interrupt dari perangkat keras. Interrupts adalah sinyal yang dikirim oleh perangkat keras untuk memberitahu CPU bahwa perangkat membutuhkan perhatian, seperti perangkat input (keyboard, mouse), perangkat jaringan, atau timer.

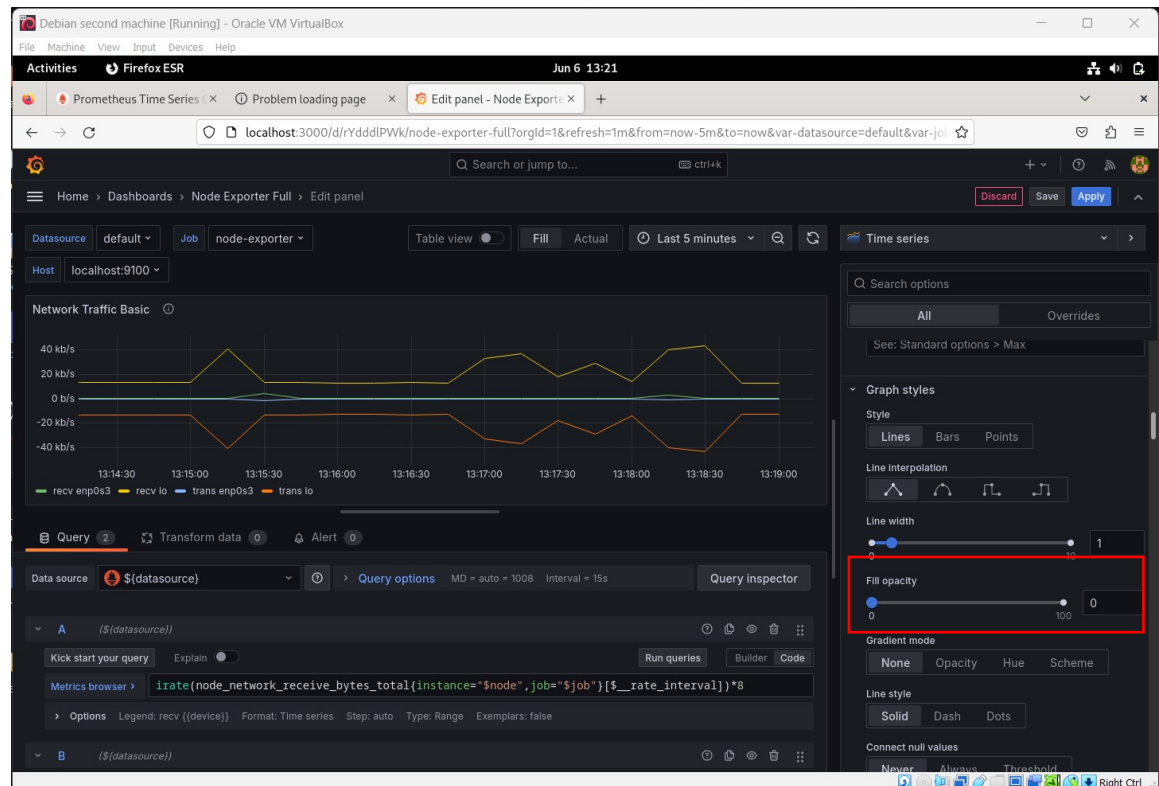
CPU Busy Other

- **CPU Busy Other** biasanya mengacu pada waktu yang dihabiskan CPU dalam keadaan lain yang tidak termasuk dalam kategori yang telah disebutkan sebelumnya. Ini mungkin mencakup waktu yang dihabiskan dalam proses idle atau waktu yang tidak bisa diklasifikasikan secara spesifik dalam metrik lainnya.

4. Ubah grafik menjadi Line tanpa fill
- a. Masuk ke page edit pada grafik yang dipilih



- b. Pada bagian kanan, ubah fill opacity menjadi 0



c. Contoh pada grafik yang lain

