

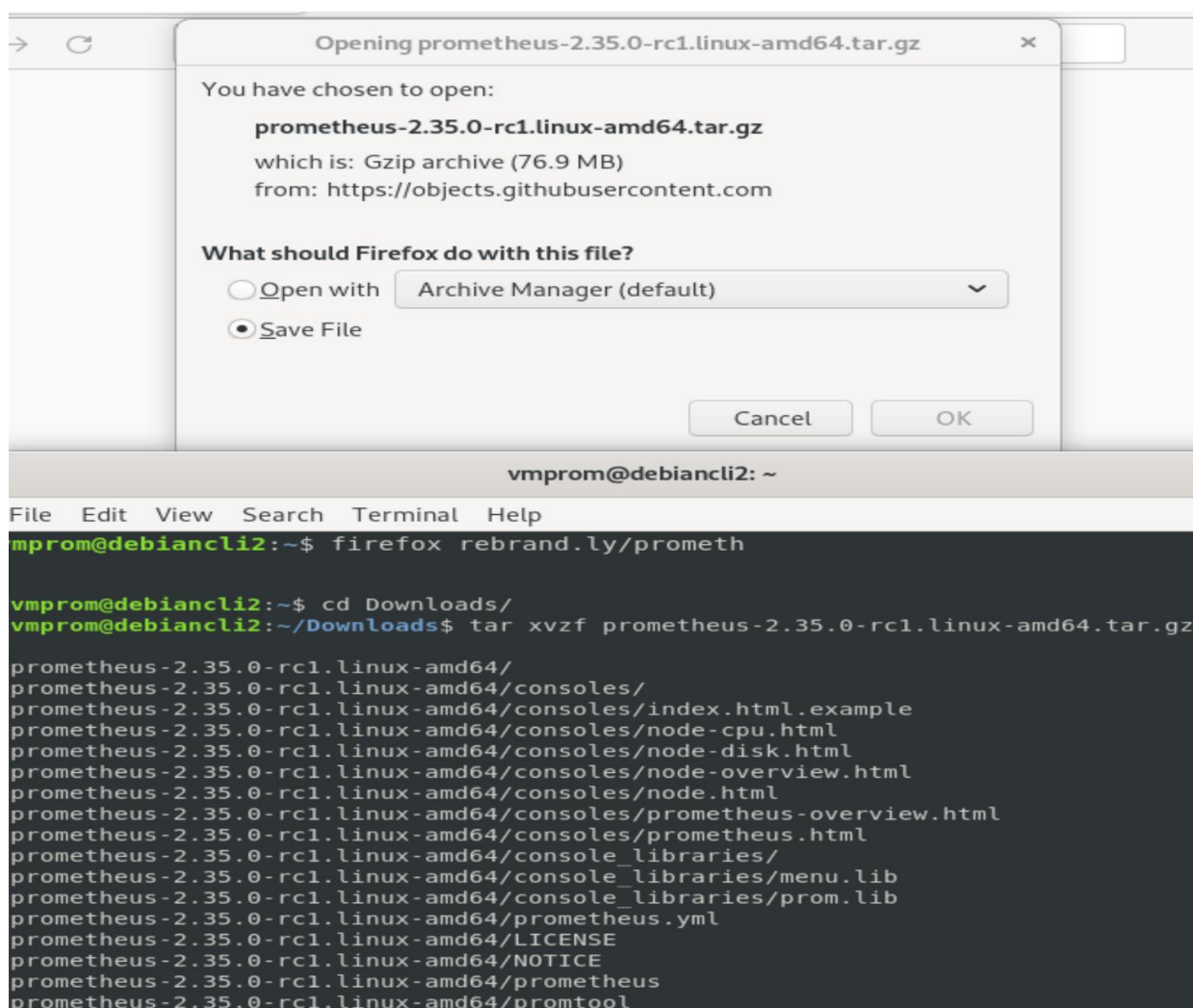
## Mise en place d'un serveur de supervision

### *I – Installation et configuration de Prometheus sur Debian avec interface graphique (obligatoire)*

Dans les éléments qui vont particulièrement nous intéresser, nous retiendrons les briques suivantes :

- **Prometheus server** : c'est le moteur permettant de stocker les métriques collectées ;
- **Jobs exporters** : il s'agit d'agents sur lesquels Prometheus viendra collecter les différentes métriques ;
- **Web UI** : restitution des métriques à l'utilisateur ;
- **Alertmanager** : gestionnaire d'envoi des alertes aux utilisateurs.

On commence par télécharger le tar.gz de Prometheus puis on le décompresse.



On déplacera le code de Prometheus dans le bon répertoire pour suivre les bonnes pratiques Linux et en suite on créera le fichier conf pour récupérer des informations du système des machines qu'on veut. (.111 = Ubuntu | .112 = Windows)

\*Node Exporter est le programme qui récupère des metrics (infos du système) et vas les rendre disponible par simple requête curl. Il utilise le port 9100 par défaut.

\*Mysql Exporter est le programme qui récupère des metrics (infos de base de données). Il utilise le port 9104 par défaut.

\*Windows Exporter est le programme qui récupère des metrics (infos du système). Il utilise le port 9182 par défaut.

```
vmprom@debiancli2:~/Downloads$ sudo mv prometheus-2.35.0-rc1.linux-amd64 /usr/share/prometheus
[sudo] password for vmprom:
vmprom@debiancli2:~/Downloads$ sudo nano /usr/share/prometheus/prometheus_config.yml
```

```
GNU nano 3.2 /usr/share/prometheus/prometheus_config.yml

scrape_configs:
- job_name: 'nodeexporter'
  scrape_interval: 5s
  static_configs:
    - targets: [ "192.168.1.111:9100" ]
- job_name: 'mysqlexporter'
  scrape_interval: 5s
  static_configs:
    - targets: [ "192.168.1.111:9104" ]
- job_name: 'windowsexporter'
  scrape_interval: 5s
  static_configs:
    - targets: [ "192.168.1.112:9182" ]
```

On créera également un utilisateur Promehteus dédié.

```
vmprom@debiancli2:~/Downloads$ sudo useradd -d /usr/share/prometheus -s /bin/false prometheus
vmprom@debiancli2:~/Downloads$ sudo mkdir -p /var/lib/prometheus/data
vmprom@debiancli2:~/Downloads$ sudo chown prometheus:prometheus /var/lib/prometheus/data
vmprom@debiancli2:~/Downloads$ sudo chown -R prometheus:prometheus /usr/share/prometheus
```

Ensuite, on fera un script dans systemd pour lancer automatiquement Prometheus.

```
GNU nano 3.2 /etc/systemd/system/prometheus.service

[Unit]
Description=Prometheus Server
Documentation=https://prometheus.io/docs/introduction/overview/
After=network-online.target

[Service]
User=prometheus
Restart=on-failure
WorkingDirectory=/usr/share/prometheus
ExecStart=/usr/share/prometheus/prometheus --config.file=/usr/share/prometheus/prometheus_config.yml

[Install]
WantedBy=multi-user.target
```

Enfin, on mettra à jour les services et on lancera la commande « enable » pour que Prometheus soit activé à chaque démarrage du système.

```
vmpr@debiancli2:~/Downloads$ sudo systemctl daemon-reload
o systemctl enable prometheus
sudo systemctl start prometheus
sudo systemctl status prometheusvmpr@debiancli2:~/Downloads$ sudo systemctl enable prometheus
Created symlink /etc/systemd/system/multi-user.target.wants/prometheus.service → /etc/systemd/system/prometheus.service.
vmpr@debiancli2:~/Downloads$ sudo systemctl start prometheus
vmpr@debiancli2:~/Downloads$ sudo systemctl status prometheus
● prometheus.service - Prometheus Server
   Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2022-04-17 16:46:32 CEST; 5s ago
     Docs: https://prometheus.io/docs/introduction/overview/
  Main PID: 3511 (prometheus)
    Tasks: 6 (limit: 4689)
   Memory: 19.5M
   CGroup: /system.slice/prometheus.service
           └─3511 /usr/share/prometheus/prometheus --config.file=/usr/share/prometheus/prometheus_config.yml

Apr 17 16:46:32 debiancli2 prometheus[3511]: ts=2022-04-17T14:46:32.449Z caller=head.go:536 level=info component=tsdb ms
Apr 17 16:46:32 debiancli2 prometheus[3511]: ts=2022-04-17T14:46:32.449Z caller=head.go:542 level=info component=tsdb ms
Apr 17 16:46:32 debiancli2 prometheus[3511]: ts=2022-04-17T14:46:32.452Z caller=tls_config.go:195 level=info component=w
Apr 17 16:46:32 debiancli2 prometheus[3511]: ts=2022-04-17T14:46:32.452Z caller=head.go:613 level=info component=tsdb ms
Apr 17 16:46:32 debiancli2 prometheus[3511]: ts=2022-04-17T14:46:32.453Z caller=head.go:619 level=info component=tsdb ms
Apr 17 16:46:32 debiancli2 prometheus[3511]: ts=2022-04-17T14:46:32.455Z caller=main.go:978 level=info fs_type=EXT4_SUPE
Apr 17 16:46:32 debiancli2 prometheus[3511]: ts=2022-04-17T14:46:32.455Z caller=main.go:981 level=info msg="TSDB started
Apr 17 16:46:32 debiancli2 prometheus[3511]: ts=2022-04-17T14:46:32.455Z caller=main.go:1162 level=info msg="Loading con
Apr 17 16:46:32 debiancli2 prometheus[3511]: ts=2022-04-17T14:46:32.456Z caller=main.go:1199 level=info msg="Completed l
Apr 17 16:46:32 debiancli2 prometheus[3511]: ts=2022-04-17T14:46:32.456Z caller=main.go:930 level=info msg="Server is re
lines 1-20/20 (END)
```

## II – Installation de Node Exporter et Mysql Exporter sur Ubuntu

La première chose que nous ferons est de définir une adresse IP statique sur notre machine virtuelle Ubuntu. On tapera la commande `IP -c link` pour savoir le nom de notre interface.

```
vmnode@vmnode-VirtualBox:~/Bureau$ ip -c link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
   T group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP m
   ode DEFAULT group default qlen 1000
    link/ether 08:00:27:4e:78:89 brd ff:ff:ff:ff:ff:ff
```

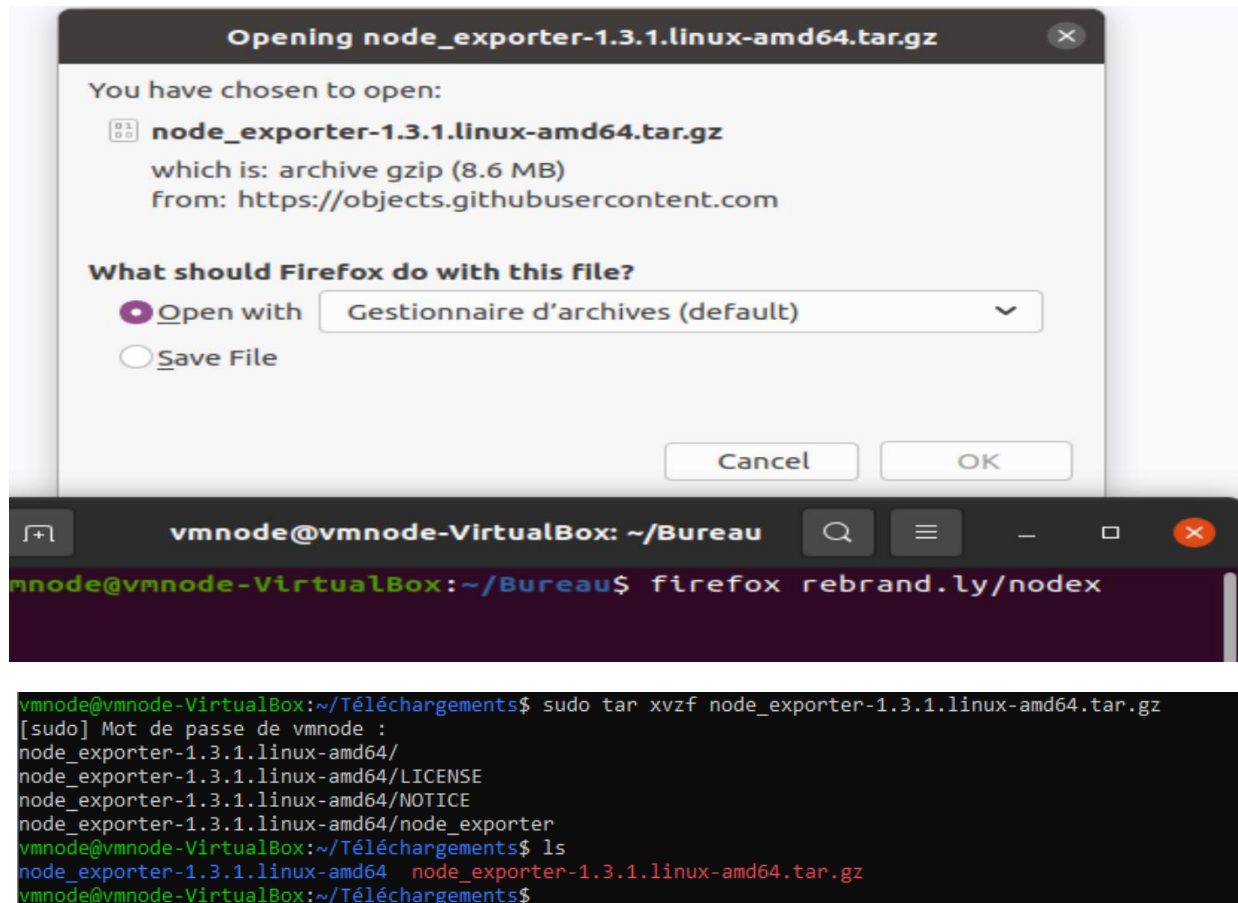
Ignorez l'interface `lo`, nous n'avons pas besoin. Ce qu'on utilisera est l'interface 2.

L'étape suivante consiste à ouvrir le fichier d'interfaces et à définir l'adresse IP de l'interface. « `sudo nano /etc/netplan/01...` » et « `sudo netplan apply` » pour actualiser.

```
GNU nano 4.8 /etc/netplan/01-network-manager-all.yaml
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    enp0s3:
      dhcp4: yes
      dhcp6: no
      addresses: [192.168.1.111/24]
      gateway4: 192.168.1.254
```

En réalité, on n'installera pas vraiment Node Exporter, on crée juste un script systemd qui va lancer la commande.

On commence encore une fois par télécharger le tar.gz de Node Exporter et puis on le décompresse.



On déplacera le dossier node\_exporter dans le dossier /usr/local/bin pour suivre les bonnes pratiques Linux. Et on crée également un utilisateur dédié pour Node Exporter.

```
vmnode@vmnode-VirtualBox:~/Téléchargements$ sudo mv node_exporter-1.3.1.linux-amd64 /usr/local/bin
[sudo] Mot de passe de vmnode :
vmnode@vmnode-VirtualBox:~/Téléchargements$ useradd -rs /bin/false node_exporter
useradd: Permission denied.
useradd : impossible de verrouiller /etc/passwd ; veuillez réessayer plus tard.
vmnode@vmnode-VirtualBox:~/Téléchargements$ sudo useradd -rs /bin/false node_exporter
```

Nous allons maintenant créer le script systemd pour NE.

```
GNU nano 4.8 /etc/systemd/system/node_exporter.service
[Unit]
Description=Node Exporter
After=network.target

[Service]
User=node_exporter
Group=node_exporter
Type=simple
ExecStart=/usr/local/bin/node_exporter-1.3.1.linux-amd64/node_exporter

[Install]
WantedBy=multi-user.target
```

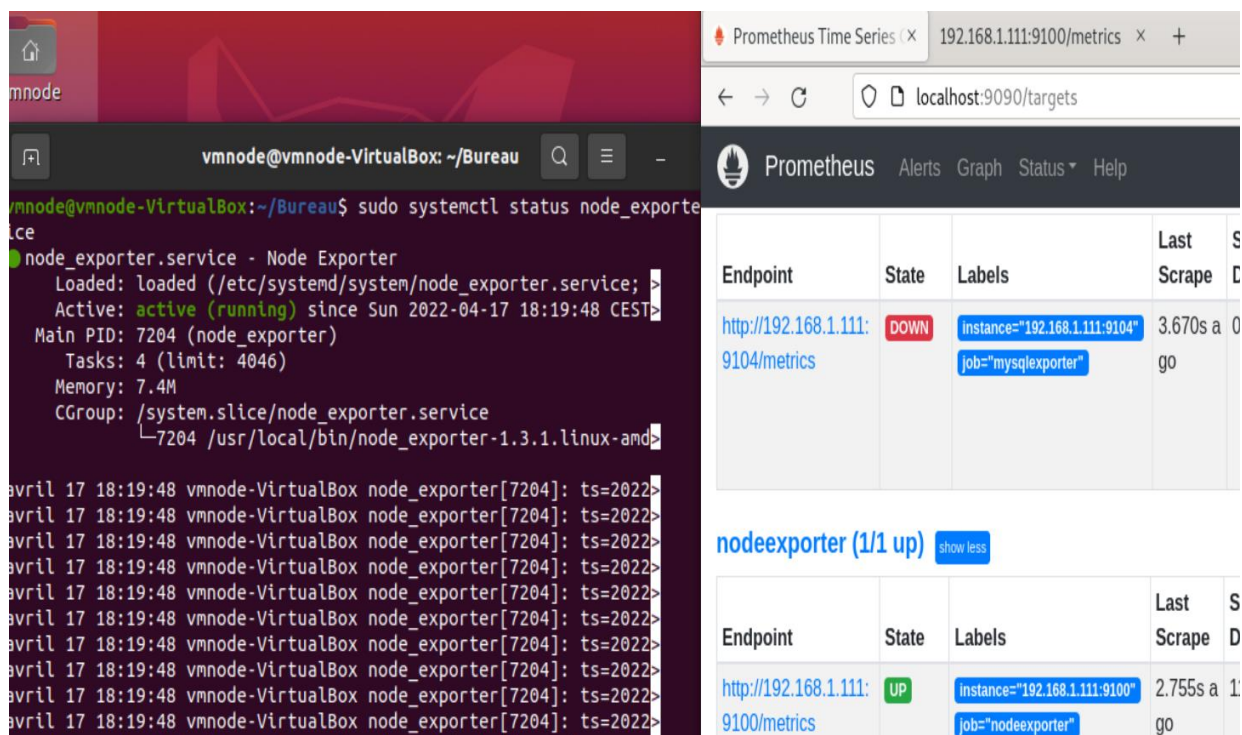


Il faut recharger le daemon et l'ajouter au service de démarrage.

```
vmnode@vmnode-VirtualBox:~/Téléchargements$ sudo systemctl daemon-reload
o systemctl start node_exporter
sudo systemctl status node_exporter
vmnode@vmnode-VirtualBox:~/Téléchargements$ sudo systemctl enable node_exporter
vmnode@vmnode-VirtualBox:~/Téléchargements$ sudo systemctl status node_exporter
● node_exporter.service - Node Exporter
   Loaded: loaded (/etc/systemd/system/node_exporter.service; disabled; vendor preset: enabled)
   Active: active (running) since Sun 2022-04-17 18:19:48 CEST; 36ms ago
     Main PID: 7204 (node_exporter)
       Tasks: 3 (limit: 4046)
      Memory: 2.1M
    CGroup: /system.slice/node_exporter.service
            └─7204 /usr/local/bin/node_exporter-1.3.1.linux-amd64/node_exporter

avril 17 18:19:48 vmnode-VirtualBox node_exporter[7204]: ts=2022-04-17T16:19:48.766Z caller=node_exporter.go:115 level=
avril 17 18:19:48 vmnode-VirtualBox node_exporter[7204]: ts=2022-04-17T16:19:48.766Z caller=node_exporter.go:115 level=
avril 17 18:19:48 vmnode-VirtualBox node_exporter[7204]: ts=2022-04-17T16:19:48.766Z caller=node_exporter.go:115 level=
avril 17 18:19:48 vmnode-VirtualBox node_exporter[7204]: ts=2022-04-17T16:19:48.766Z caller=node_exporter.go:115 level=
avril 17 18:19:48 vmnode-VirtualBox node_exporter[7204]: ts=2022-04-17T16:19:48.767Z caller=node_exporter.go:115 level=
avril 17 18:19:48 vmnode-VirtualBox node_exporter[7204]: ts=2022-04-17T16:19:48.767Z caller=node_exporter.go:115 level=
avril 17 18:19:48 vmnode-VirtualBox node_exporter[7204]: ts=2022-04-17T16:19:48.767Z caller=node_exporter.go:115 level=
avril 17 18:19:48 vmnode-VirtualBox node_exporter[7204]: ts=2022-04-17T16:19:48.767Z caller=node_exporter.go:199 level=
avril 17 18:19:48 vmnode-VirtualBox node_exporter[7204]: ts=2022-04-17T16:19:48.768Z caller=tls_config.go:195 level=inf
log file: udo systemctl enable node_exporter
```

On peut maintenant superviser la machine virtuelle Ubuntu depuis Prometheus sur Debian (« localhost:9090/targets »).



The screenshot displays the terminal output of the `systemctl status node_exporter` command and the Prometheus web interface. The terminal shows that the `node_exporter` service is active and running. The Prometheus web interface shows the target status for `192.168.1.111:9100/metrics`.

**Terminal Output:**

```
vmnode@vmnode-VirtualBox:~/Bureau$ sudo systemctl status node_exporter
● node_exporter.service - Node Exporter
   Loaded: loaded (/etc/systemd/system/node_exporter.service; >
   Active: active (running) since Sun 2022-04-17 18:19:48 CEST>
     Main PID: 7204 (node_exporter)
       Tasks: 4 (limit: 4046)
      Memory: 7.4M
    CGroup: /system.slice/node_exporter.service
            └─7204 /usr/local/bin/node_exporter-1.3.1.linux-amd>

avril 17 18:19:48 vmnode-VirtualBox node_exporter[7204]: ts=2022>
avril 17 18:19:48 vmnode-VirtualBox node_exporter[7204]: ts=2022>
avril 17 18:19:48 vmnode-VirtualBox node_exporter[7204]: ts=2022>
avril 17 18:19:48 vmnode-VirtualBox node_exporter[7204]: ts=2022>
avril 17 18:19:48 vmnode-VirtualBox node_exporter[7204]: ts=2022>
avril 17 18:19:48 vmnode-VirtualBox node_exporter[7204]: ts=2022>
avril 17 18:19:48 vmnode-VirtualBox node_exporter[7204]: ts=2022>
avril 17 18:19:48 vmnode-VirtualBox node_exporter[7204]: ts=2022>
avril 17 18:19:48 vmnode-VirtualBox node_exporter[7204]: ts=2022>
```

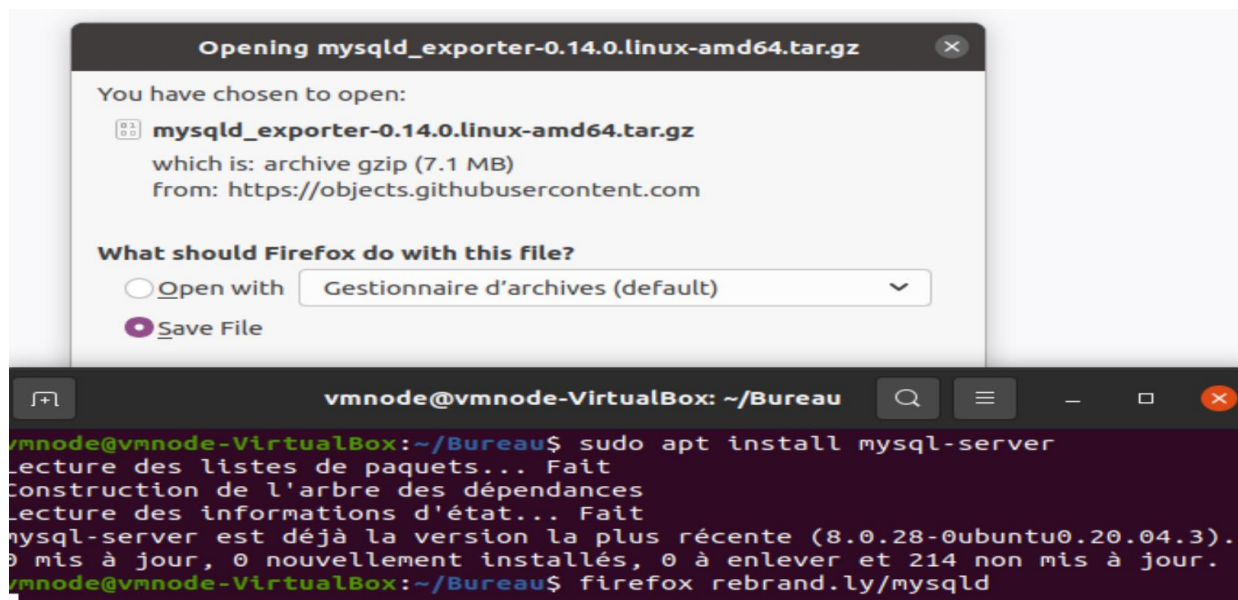
**Prometheus Web Interface:**

Endpoint: <http://192.168.1.111:9104/metrics> State: DOWN Labels: instance="192.168.1.111:9104", job="mysqlexporter" Last Scrape: 3.670s ago

nodeexporter (1/1 up) [show less](#)

Endpoint: <http://192.168.1.111:9100/metrics> State: UP Labels: instance="192.168.1.111:9100", job="nodeexporter" Last Scrape: 2.755s ago

Ensuite, on installe mysql et mysql\_exporter sur Ubuntu.



Pour l'installation de mysql\_exporter, on créera un dossier pour notre exporter et on ajoutera un utilisateur dédié.

```
vmnode@vmnode-VirtualBox:~/Téléchargements$ sudo mkdir /opt/prometheus_exporters
[sudo] Mot de passe de vmnode :
vmnode@vmnode-VirtualBox:~/Téléchargements$ sudo useradd -d /opt/prometheus_exporters -s /bin/false prometheus
```

On décompresse ensuite le tar.gz dans le « /opt/prometheus\_exporters ».

```
vmnode@vmnode-VirtualBox:~/Téléchargements$ sudo tar xzf mysqld_exporter-0.14.0.linux-amd64.tar.gz -C /opt/prometheus_exporters/
vmnode@vmnode-VirtualBox:~/Téléchargements$
```

Il nous faut un utilisateur dédié avec les bons droits pour la base de données.

```
mysql> create user 'mysql'@'localhost' identified by 'mysql';
Query OK, 0 rows affected (0,01 sec)

mysql> grant replication client on *.* to 'mysql'@'localhost'
-> ;
Query OK, 0 rows affected (0,06 sec)

mysql> grant select on performance_schema.* to 'mysql'@'localhost';
Query OK, 0 rows affected (0,01 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0,00 sec)
```

Pour pouvoir récupérer des infos de la base de donnée, il nous faut l'identifiant et le mdp d'un utilisateur pour mysql exporter.

```
vmnode@vmnode-VirtualBox: ~  
GNU nano 4.8 /opt/prometheus_exporters/.my.cnf  
[client]  
user=mysql  
password=mysql
```

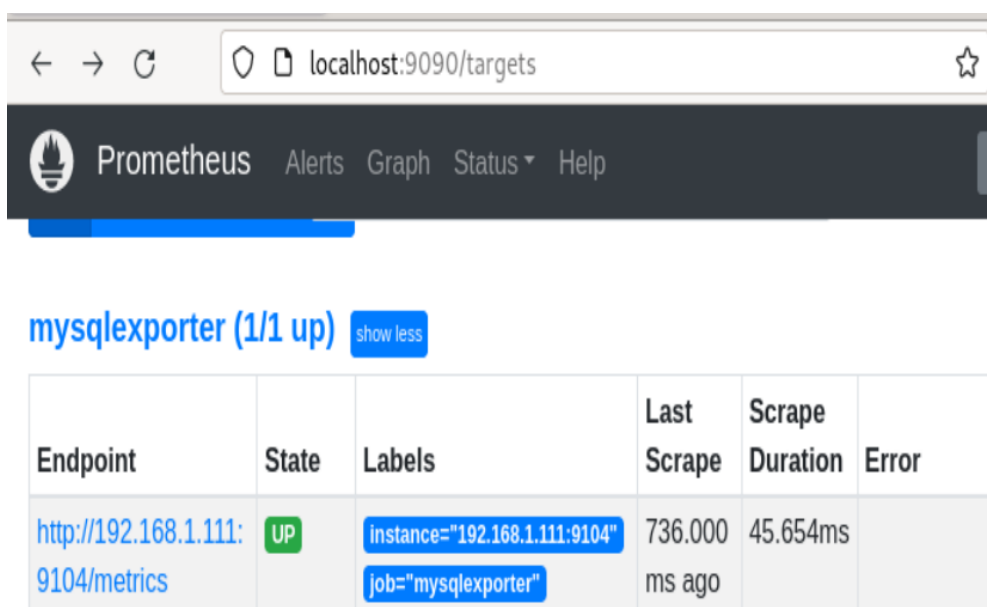
Nous allons maintenant créer le script systemd pour Mysql Exporter.

```
vmnode@vmnode-VirtualBox: ~  
GNU nano 4.8 /lib/systemd/system/mysql_exporter.service  
[Unit]  
Description=MySQL Exporter  
User=prometheus  
  
[Service]  
Type=simple  
Restart=always  
ExecStart=/opt/prometheus_exporters/mysqld_exporter-0.14.0.linux-amd64/mysqld_exporter \\  
--config.my-cnf /opt/prometheus_exporters/.my.cnf \\  
--collect.auto_increment.columns \\  
--collect.binlog_size \\  
--collect.info_schema.tables \\  
--collect.engine_innodb_status \\  
--collect.engine_tokudb_status \\  
--collect.global_status \\  
--web.listen-address=0.0.0.0:9104  
  
[Install]  
WantedBy=multi-user.target
```

Enfin, on recharge le daemon et puis on démarre notre service. Pensez à enable le service également.

```
vmnode@vmnode-VirtualBox:~$ sudo systemctl daemon-reload  
^[[Avmnode@vmnode-VirtualBox:~$ sudo systemctl start mysql_exporter.service  
vmnode@vmnode-VirtualBox:~$ sudo systemctl enable mysql_exporter.service  
vmnode@vmnode-VirtualBox:~$ sudo systemctl status mysql_exporter.service  
● mysql_exporter.service - MySQL Exporter  
   Loaded: loaded (/lib/systemd/system/mysql_exporter.service; enabled; vendor preset: enabled)  
   Active: active (running) since Sun 2022-04-17 20:36:16 CEST; 18s ago  
     Main PID: 9746 (mysqld_exporter)  
       Tasks: 3 (limit: 4046)  
      Memory: 6.0M  
     CGroup: /system.slice/mysql_exporter.service  
             └─9746 /opt/prometheus_exporters/mysqld_exporter-0.14.0.linux-amd64/mysqld_exporter --config.my-cnf
```

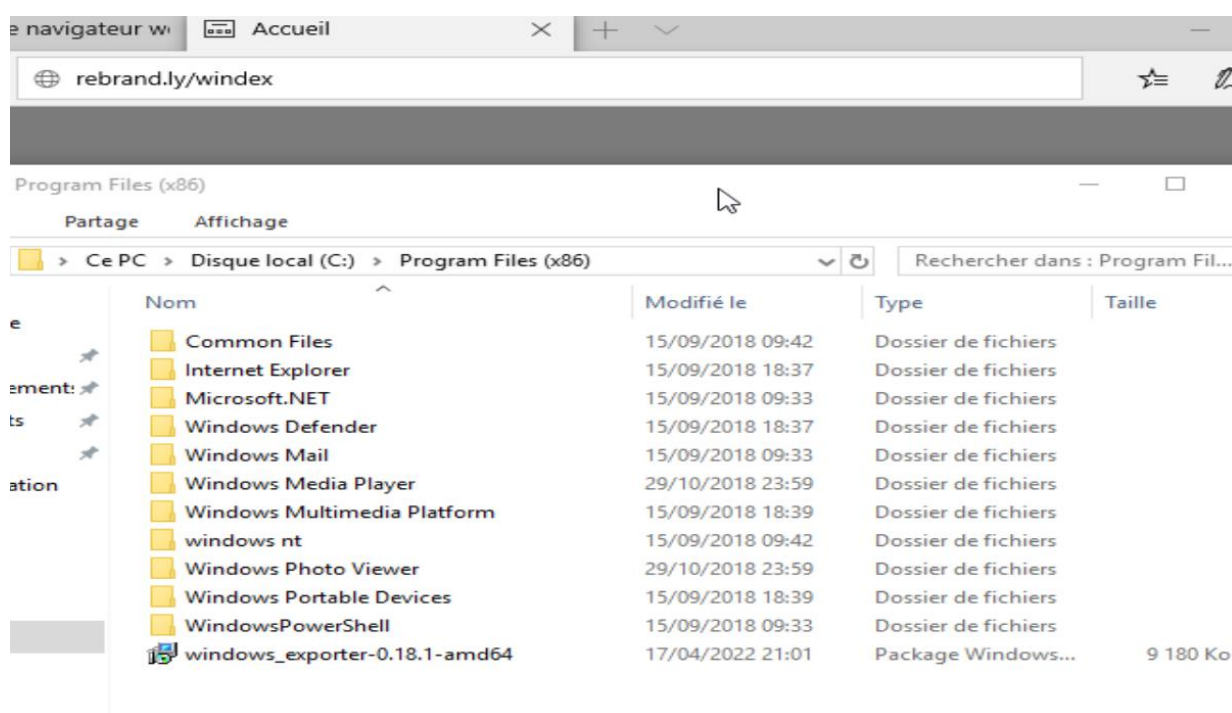
On peut maintenant superviser les bases de données d'Ubuntu depuis Prometheus sur Debian (« localhost:9090/targets »).



Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
<a href="http://192.168.1.111:9104/metrics">http://192.168.1.111:9104/metrics</a>	UP	<code>instance="192.168.1.111:9104"</code> <code>job="mysqlexporter"</code>	736.000 ms ago	45.654ms	

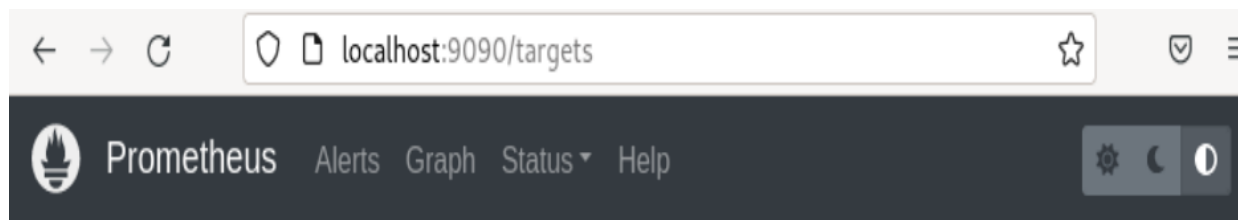
### III – Installation de Windows Exporter sur Windows 10

L'installation de Windows Exporter est tout simple, il suffit de télécharger le .msi (rebrand.ly/windex) et puis l'exécuter.





On peut maintenant superviser la machine virtuelle Windows 10 depuis Prometheus sur Debian (« localhost:9090/targets »).



### nodeexporter (0/1 up) [show less](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
<a href="http://192.168.1.111:9100/metrics">http://192.168.1.111:9100/metrics</a>	DOWN	<code>instance="192.168.1.111:9100"</code> <code>job="nodeexporter"</code>	10.7s ago	5.0s	Get "http://192.168.1.111:9100/metrics": context deadline exceeded

### windowsexporter (1/1 up) [show less](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
<a href="http://192.168.1.112:9182/metrics">http://192.168.1.112:9182/metrics</a>	UP	<code>instance="192.168.1.112:9182"</code> <code>job="windowsexporter"</code>	4.295s ago	1.77s	

## IV – Installation de Grafana et la mise en place des dashboard

Nous téléchargeons la clé GPG de Grafana d'abord avec wget, puis acheminez la sortie à apt-key. Cela ajoutera la clé à la liste des clés de confiance de votre installation APT, ce qui permettra de télécharger et de vérifier le paquet Grafana signé par GPG :

```
vmprom@debiancli2:~$ wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -  
[sudo] password for vmprom:  
OK  
vmprom@debiancli2:~$
```

Ajoutez ensuite le référentiel Grafana à vos sources APT et puis actualiser votre cache APT pour mettre à jour vos liste de paquets.

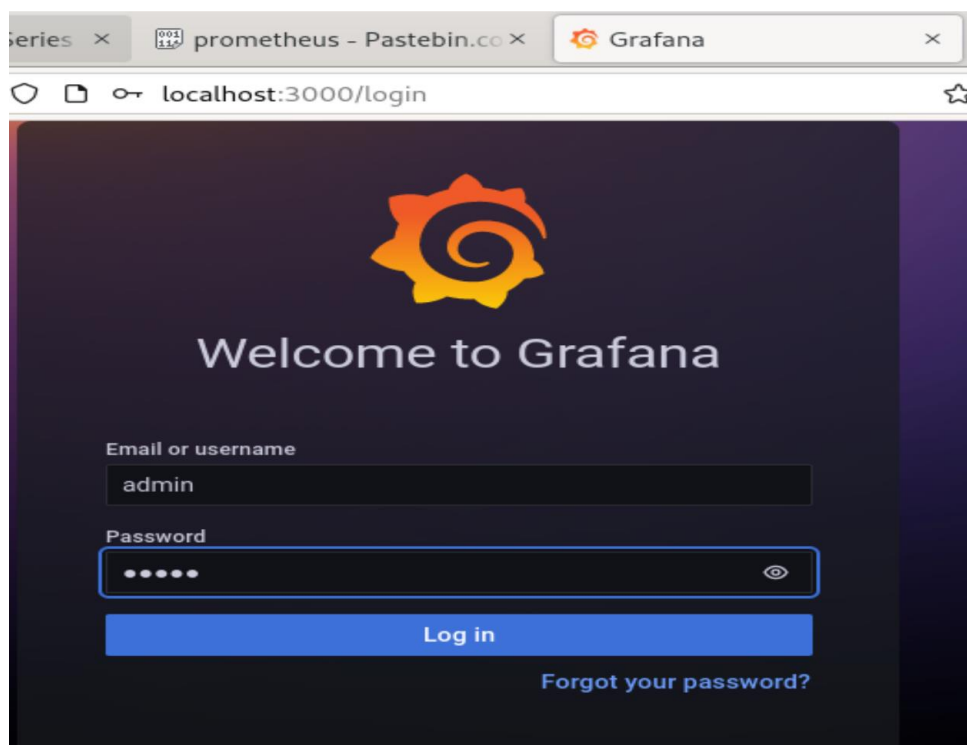
```
vmprom@debiancli2:~$ sudo add-apt-repository "deb https://packages.grafana.com/oss/deb stable main"
vmprom@debiancli2:~$ sudo apt update
```

Installation de Grafana et puis activation du service pour démarrer automatiquement Grafana au boot

```
vmprom@debiancli2:~$ sudo apt install grafana
```

```
vmprom@debiancli2:~$ sudo systemctl start grafana-server
systemctl status grafana-servervmprom@debiancli2:~$ sudo systemctl status grafana-server
● grafana-server.service - Grafana instance
   Loaded: loaded (/lib/systemd/system/grafana-server.service; disabled; vendor preset: enabled)
   Active: active (running) since Fri 2022-04-22 16:44:26 CEST; 2s ago
     Docs: http://docs.grafana.org
    Main PID: 4193 (grafana-server)
      Tasks: 5 (limit: 4689)
     Memory: 23.7M
    CGroup: /system.slice/grafana-server.service
            └─4193 /usr/sbin/grafana-server --config=/etc/grafana/grafana.ini --pidfile=/run/grafana/grafana-server.pid -
Apr 22 16:44:28 debiancli2 grafana-server[4193]: logger=migrator t=2022-04-22T16:44:28.36+0200 lvl=info msg="Executing m
Apr 22 16:44:28 debiancli2 grafana-server[4193]: logger=migrator t=2022-04-22T16:44:28.36+0200 lvl=info msg="Executing m
Apr 22 16:44:28 debiancli2 grafana-server[4193]: logger=migrator t=2022-04-22T16:44:28.37+0200 lvl=info msg="Executing m
Apr 22 16:44:28 debiancli2 grafana-server[4193]: logger=migrator t=2022-04-22T16:44:28.38+0200 lvl=info msg="Executing m
Apr 22 16:44:28 debiancli2 grafana-server[4193]: logger=migrator t=2022-04-22T16:44:28.39+0200 lvl=info msg="Executing m
Apr 22 16:44:28 debiancli2 grafana-server[4193]: logger=migrator t=2022-04-22T16:44:28.39+0200 lvl=info msg="Executing m
Apr 22 16:44:28 debiancli2 grafana-server[4193]: logger=migrator t=2022-04-22T16:44:28.4+0200 lvl=info msg="Executing m
Apr 22 16:44:28 debiancli2 grafana-server[4193]: logger=migrator t=2022-04-22T16:44:28.42+0200 lvl=info msg="Executing m
Apr 22 16:44:28 debiancli2 grafana-server[4193]: logger=migrator t=2022-04-22T16:44:28.42+0200 lvl=info msg="Executing m
Apr 22 16:44:28 debiancli2 grafana-server[4193]: logger=migrator t=2022-04-22T16:44:28.44+0200 lvl=info msg="Executing m
vmprom@debiancli2:~$ sudo systemctl enable grafana-server
```

Entrez « localhost :3000 » pour accéder à l'interface de Grafana, l'identifiant et le mot de passe est « admin » par défaut.



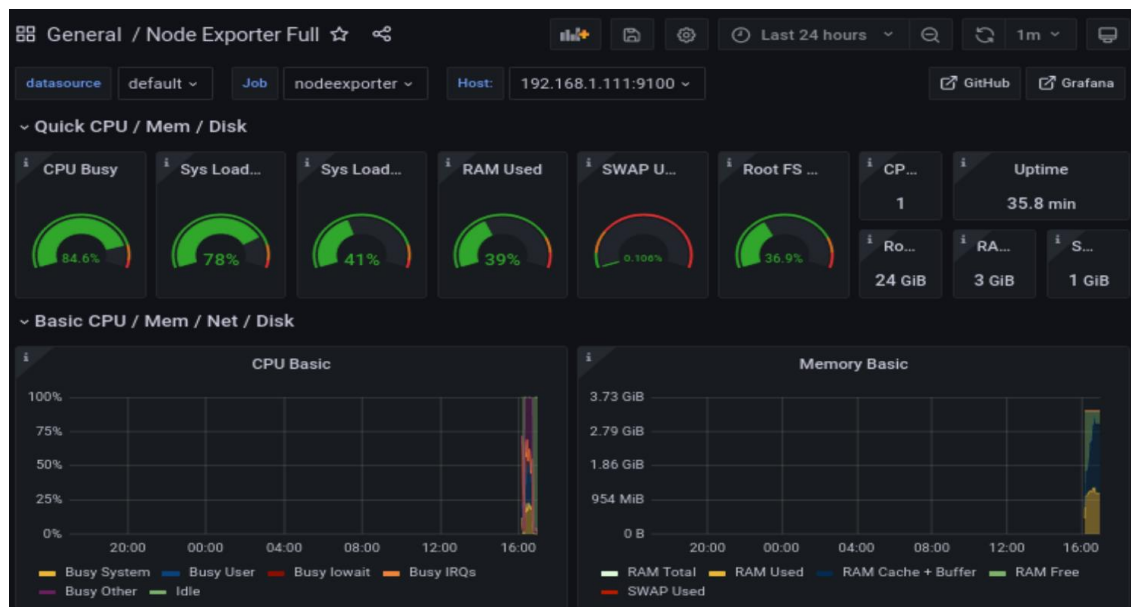
Il faut maintenant ajouter Prometheus dans la Configuration/Data sources, il faut juste préciser l'URL et puis la sauvegarder.

The screenshot shows the 'Data Sources / Prometheus' configuration page in Grafana. The page title is 'Data Sources / Prometheus' with a subtitle 'Type: Prometheus'. Below the title is a 'Settings' dropdown menu. A blue box with an information icon contains the text: 'Configure your Prometheus data source below. Or skip the effort and get Prometheus (and Loki) as fully-managed, scalable, and hosted data sources from Grafana Labs with the free-forever Grafana Cloud plan.' Below this, there is a 'Name' field with the value 'Prometheus' and a 'Default' toggle switch that is turned on. Under the 'HTTP' section, there is a 'URL' field with the value 'http://localhost:9090' and an 'Access' dropdown menu with the value 'Server (default)'. A 'Help' link is visible at the bottom right.

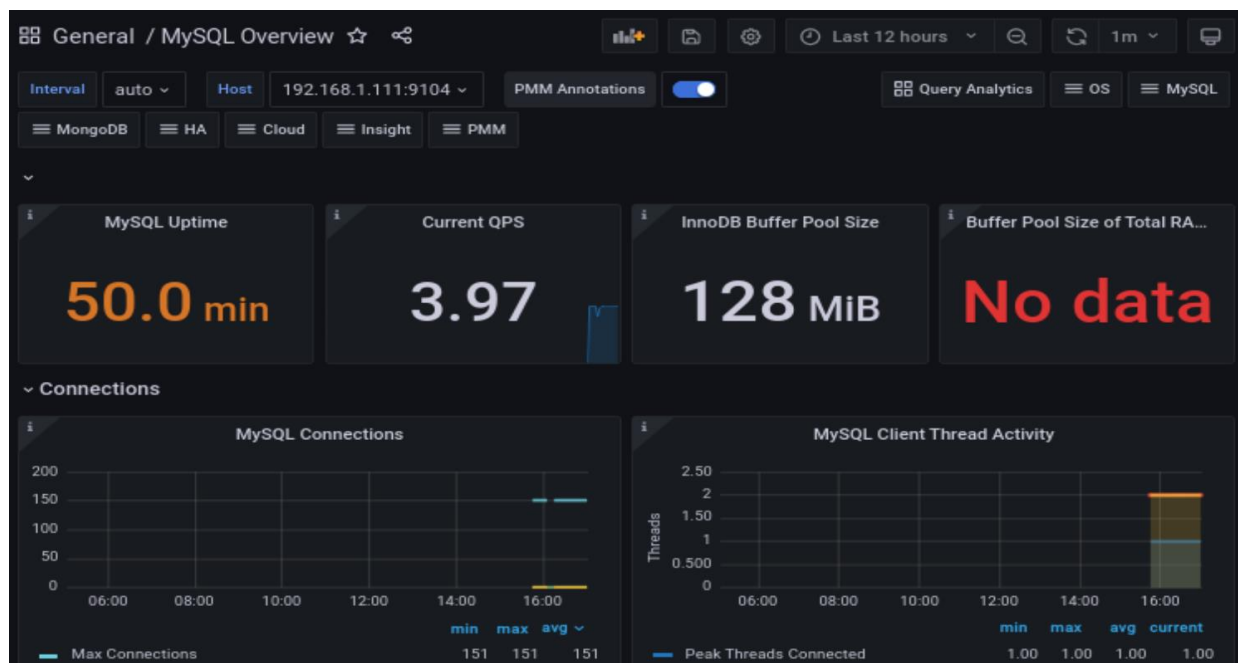
Ensuite, nous allons importer des dashboards venant du site grafana.com, donc « 1860 » est un dashboard pour le Node Exporter, « 7362 » pour le MySQL exporter, et « 14694 » pour le Windows Exporter.

The screenshot shows the 'Import' page in Grafana. The page title is 'Import' with a subtitle 'Import dashboard from file or Grafana.com'. Below the title is a blue button labeled 'Upload JSON file'. Below this, there is a section 'Import via grafana.com' with a text input field containing the value '1860' and a blue 'Load' button. Below this, there is a section 'Importing dashboard from Grafana.com' with a table showing the dashboard details: 'Published by' is 'rfraile' and 'Updated on' is '2022-02-08 20:51:42'. Below the table is a section 'Options' with a 'Name' field containing 'Node Exporter Full', a 'Folder' dropdown menu with the value 'General', and a 'Unique identifier (UID)' field containing 'rYdddIPWk' with a blue 'Change uid' button. Below this, there is a 'Prometheus' section with a dropdown menu showing 'Prometheus'. At the bottom, there are 'Import' and 'Cancel' buttons.

On peut maintenant superviser la machine virtuelle Ubuntu, et avoir plus d'informations sur son état et son comportement comme le nombre de stockage et de RAM qu'il reste.



On peut également superviser les bases de données de la machine Ubuntu.



Documentation réalisée par Aaron PESCASIO.

