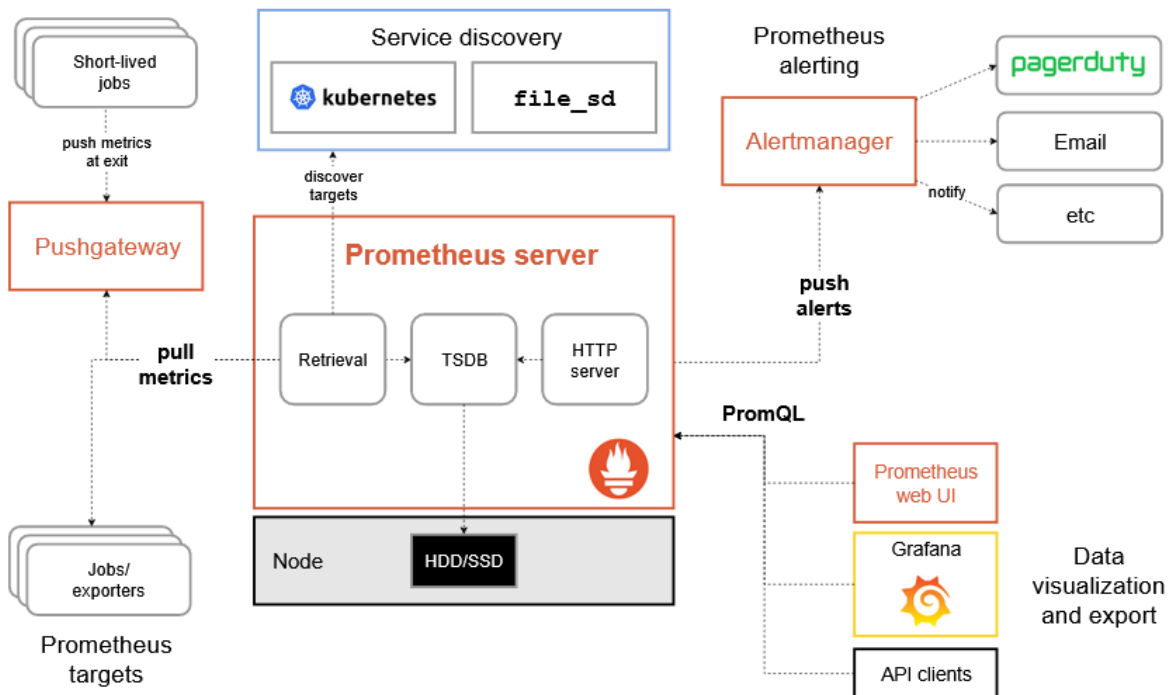


Prometheus-Node exporter-Grafana:



Fonctionnement de prometheus

Prometheus collecte les données en provenances des clients nommés *Exporters* installés sur les instances distantes.

Il interroge à intervalle régulier ces agents pour récupérer des métriques (données).

Il existe différents Exporter :

- Des **Third-party Exporters**, permettant de récupérer des métriques de tout type.
L'un des Exporters important est le System Exporter permettant de récupérer toutes les métriques systèmes ;
- Des **Blackbox Exporters** qui permettent de récupérer des métriques à travers différents protocoles ;
- D'autres outils comme **Telegraf** permettant d'exposer des données à Prometheus ;
- Créer ses propres Exporters si ceux fournis par défaut ne correspondent pas aux besoins.

Précision :

Superviser une infrastructure complexe qui comporte plusieurs niveaux d'isolations, le mode *Pull* devient problématique.

Au lieu d'utiliser le mode *Push* qui vient à l'esprit naturellement, **Prometheus** fournit un **proxy** permettant de conserver le modèle *Pull*, et de superviser tous les systèmes derrière le(s) **Firewall(s)**.

Système d'Alerting dans Prometheus :

Prometheus possède un module nommé Alertmanager qui est capable de transmettre les alarmes à différents types de média (Mail, *Pager Duty*, etc...)

Ces fonctionnalités :

1. Grouper les alertes : une seule notification, regroupant toutes les alertes des N dernières secondes / minutes, est envoyée ;
2. Rendre muettes les alertes : ne pas envoyer la même alerte pendant une période déterminée
3. Haute disponibilité : Alertmanager supporte la configuration en mode Cluster permettant la haute disponibilité.

Scalabilité et haute disponibilité :

Définition :

Scalabilité : fait référence à la capacité d'augmenter ou de réduire la quantité de ressources informatiques selon les besoins pour répondre à l'évolution de la demande.

Essence même du cloud.

Procédure mise en place du Prometheus Serveur

Serveur Prometheus

Installation :

Téléchargement de l'archive de prometheus.

<https://prometheus.io/download/>.

Operating system

popular

Architecture

amd64

prometheus

The Prometheus monitoring system and time series database. [prometheus/prometheus](#)

2.33.0-rc.1 / 2022-01-19

Pre-release

Release notes

File name	OS	Arch	Size	SHA256 Checksum
prometheus-2.33.0-rc.1.darwin-amd64.tar.gz	darwin	amd64	72.35 MiB	c6a77b50809952783cfa0a8ce6fd8a577f4ff8b52bdd19fa6929ac93fc83b27d
prometheus-2.33.0-rc.1.linux-amd64.tar.gz	linux	amd64	72.30 MiB	bd08609dec35f3e82d0412541c961f963417cf4b2bec9d43be95ccce35cd4a04

Extraction du fichier zip tar xfvz (suivis du nom de l'archive).

```
ubuntu@ubuntu2004:~/Downloads$ tar xfvz prometheus-2.33.0-rc.1.linux-amd64.tar.gz
prometheus-2.33.0-rc.1.linux-amd64/
```

Se déplacer dans le dossier download ou a été extraie prometheus.

-Puis le lancer.

```
ubuntu@ubuntu2004:~/Downloads$ ./prometheus-2.33.0-rc.1.linux-amd64/prometheus --
config.file=prometheus-2.33.0-rc.1.linux-amd64/prometheus.yml
```

Après avoir lancer prometheus il est joignable en local grace a l'url : <http://localhost:9090>

Configuration de prometheus :

Creation d'un fichier : faire une copie du fichier prometheus.yml (si cest sans le node exporter)

Installation d'un node exporter :

- Retourner sur la même url, pour y chercher le node exporter.

node_exporter

Exporter for machine metrics [prometheus/node_exporter](#)

1.3.1 / 2021-12-01 Release notes				
File name	OS	Arch	Size	SHA256 Checksum
node_exporter-1.3.1.darwin-amd64.tar.gz	darwin	amd64	4.22 MiB	9e954a08597f3ee7a503f010801aaec3dae054b0ec49a641ea99836e7f87e99
node_exporter-1.3.1.linux-amd64.tar.gz	linux	amd64	8.61 MiB	60f3002c2dd3900667e4ba65ea2e1fb03f4a4ba026cca375f15a0390ff850949

- Extraire l'archive

```
ubuntu@ubuntu2004:~/Downloads$ tar xfvz node_exporter-1.3.1.linux-amd64.tar.gz
node_exporter-1.3.1.linux-amd64/
node_exporter-1.3.1.linux-amd64/LICENSE
node_exporter-1.3.1.linux-amd64/NOTICE
node_exporter-1.3.1.linux-amd64/node_exporter
ubuntu@ubuntu2004:~/Downloads$ ls
data
node_exporter-1.3.1.linux-amd64
node_exporter-1.3.1.linux-amd64.tar.gz
passPDF.pdf
passZIP.zip
prometheus-2.33.0-rc.1.linux-amd64
prometheus-2.33.0-rc.1.linux-amd64.tar.gz
savedstate.sav
```

Création du service prometheus

```
sudo mv (la version du prometheus installer) /usr/share/prometheus
```

#Permet de de placer le serveur dans le bon repertoire

```
sudo mv ~/prometheus.yml /usr/share/prometheus/prometheus_config.yml
```

(prometheus.yml, sagit du fichier créé précédement)

Création d'un utilisateur prometheus :

```
sudo useradd -d /usr/share/prometheus -s /bin/false prometheus
```

```
sudo mkdir -p /var/lib/prometheus/data
```

```
sudo chown prometheus:prometheus /var/lib/prometheus/data
```

```
sudo chown -R prometheus: prometheus /usr/share/prometheus
```

Création du script pour lancer prometheus de manière automatique.

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

```
GNU nano 4.8 /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/

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

Mise à jour des services :

```
ubuntu@ubuntu2004:~$ sudo systemctl daemon-reload
```

Activation de prometheus au démarrage

```
ubuntu@ubuntu2004:~$ sudo systemctl enable prometheus
Created symlink /etc/systemd/system/multi-user.target.wants/prometheus.service -
/etc/systemd/system/prometheus.service.
```

Lancement du service prometheus

```
loan@loan-ubuntu:~$ sudo systemctl start prometheus
```

Etat du service

```
loan@loan-ubuntu:~$ sudo systemctl status prometheus
● prometheus.service - Prometheus Server
   Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; vendor pre>
   Active: active (running) since Tue 2022-04-26 10:25:47 CEST; 7min ago
     Docs: https://prometheus.io/docs/introduction/overview/
    Main PID: 625 (prometheus)
      Tasks: 7 (limit: 2290)
     Memory: 71.8M
    CGroup: /system.slice/prometheus.service
            └─625 /usr/share/prometheus/prometheus --config.file=/usr/share/pr>

avril 26 10:25:51 loan-ubuntu prometheus[625]: ts=2022-04-26T08:25:51.121Z call>
avril 26 10:25:51 loan-ubuntu prometheus[625]: ts=2022-04-26T08:25:51.121Z call>
avril 26 10:25:51 loan-ubuntu prometheus[625]: ts=2022-04-26T08:25:51.121Z call>
avril 26 10:25:51 loan-ubuntu prometheus[625]: ts=2022-04-26T08:25:51.123Z call>
avril 26 10:25:51 loan-ubuntu prometheus[625]: ts=2022-04-26T08:25:51.123Z call>
avril 26 10:25:51 loan-ubuntu prometheus[625]: ts=2022-04-26T08:25:51.123Z call>
avril 26 10:25:51 loan-ubuntu prometheus[625]: ts=2022-04-26T08:25:51.124Z call>
avril 26 10:25:51 loan-ubuntu prometheus[625]: ts=2022-04-26T08:25:51.124Z call>
avril 26 10:26:08 loan-ubuntu prometheus[625]: ts=2022-04-26T08:26:08.826Z call>
avril 26 10:26:08 loan-ubuntu prometheus[625]: ts=2022-04-26T08:26:08.828Z call>
lines 1-20/20 (END)...skipping...
```

Lancement du node exporter

Le node exporter est un agent disposé sur les machines d'un réseau permettant d'envoyer des données collectées au serveur de supervision.

Effectuer une recherche depuis le terminale

```
loan@debianminatchy:~$ firefox https://prometheus.io/download/.
```

Télécharger l'archive node_exporter

node_exporter

Exporter for machine metrics [prometheus/node_exporter](#)

1.3.1 / 2021-12-01 [Release notes](#)

File name	OS	Arch	Size	SHA256 Checksum
node_exporter-1.3.1.darwin-amd64.tar.gz	darwin	amd64	4.22 MiB	9e954a08597f3ee7a503f010801aacc3dae
node_exporter-1.3.1.linux-amd64.tar.gz	linux	amd64	8.61 MiB	68f3802c2dd3980667e4ba65ea2e1fb03f4

Extraire l'archive

```
loan@debianminatchy:~/Téléchargements$ tar xfvz node_exporter-1.3.1.linux-amd64.tar.gz
```

! Bien y renseigner la version de l'archive téléchargé !

```
ubuntu@ubuntu2004:~/Downloads$ ./node_exporter-1.3.1.linux-amd64/node_exporter
```

Creation du service node_exporter

(Le service node exporter permet de lancer la commande automatiquement afin de lancer node exporter automatiquement à chaque démarrage de la machine cela évite de le faire manuellement à chaque fois que la machine démarre)

Bonne pratique :

Déplacer node_exporter dans le bon dossier :

```
loan@debianminatchy:~/Téléchargements$ sudo mv node_exporter-1.3.1.linux-amd64/node_exporter /usr/local/bin/
```

Création d'un utilisateur dédié au service node_exporter :

```
loan@debianminatchy:~/Téléchargements$ sudo useradd -rs /bin/false node_exporter
```

Création du service :

```
sudo nano /etc/systemd/system/node_exporter.service
```

```
GNU nano 3.2 /etc/systemd/system/node_exporter.service Modifié

[Unit]
Description=Node Exporter
After=network.target

[Service]
User=node_exporter
Group=node_exporter
Type=simple
ExecStart=/usr/local/bin/node_exporter

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

Lancement du service :

Rechargement des daemon->Lancement du service ->Check de l'états du service

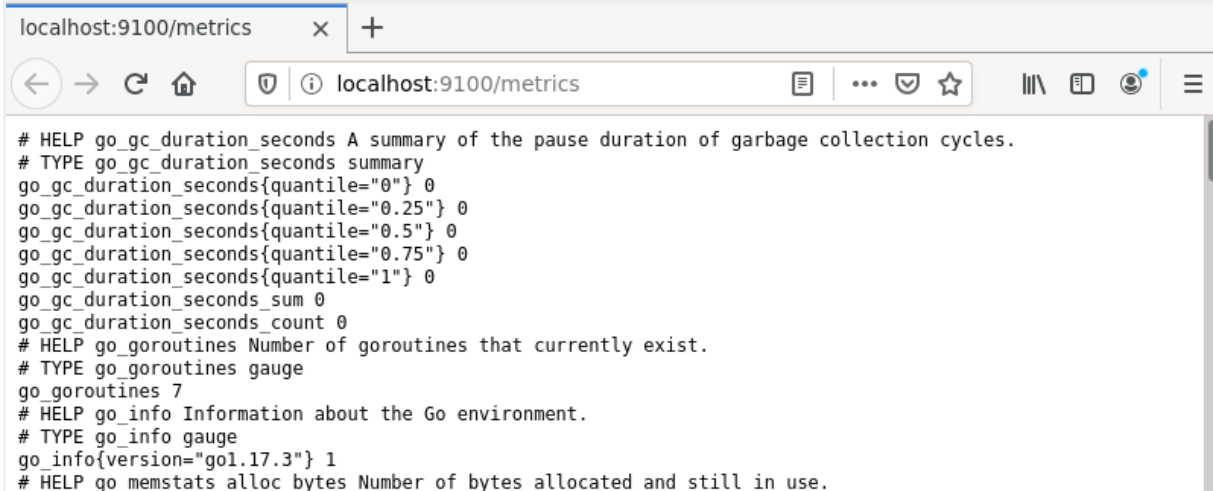
```
loan@debianminatchy:~$ sudo systemctl daemon-reload
loan@debianminatchy:~$ sudo systemctl start node_exporter
loan@debianminatchy:~$ sudo systemctl status node_exporter
● node_exporter.service - Node Exporter
   Loaded: loaded (/etc/systemd/system/node_exporter.service; disabled; vendor
   Active: active (running) since Tue 2022-04-26 11:15:03 CEST; 32s ago
   Main PID: 1226 (node_exporter)
     Tasks: 4 (limit: 4689)
    Memory: 6.8M
    CGroup: /system.slice/node_exporter.service
            └─1226 /usr/local/bin/node_exporter
```

Ajout du service de démarrage (si tous va bien) :

```
loan@debianminatchy:~$ sudo systemctl enable node_exporter
```

Seconde vérification de l'état :

```
loan@debianminatchy:~$ firefox curl http://localhost:9100/metrics
```



```
localhost:9100/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"} 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
go_gc_duration_seconds_sum 0
go_gc_duration_seconds_count 0
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 7
# 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.
```

Configuration de prometheus

!(attention à ne surtout pas modifier la configuration système de prometheus!!)

On va créer un répertoire pour que prometheus ajoute automatiquement à sa configuration tous les fichiers .yml qui y seront déposés.

```
ubuntu@ubuntu2004:~$ sudo mkdir -p /etc/prometheus/nodes
```

```
ubuntu@ubuntu2004:~$ sudo nano /etc/prometheus/nodes/localhost.yml
```

```
GNU nano 4.8 /etc/prometheus/nodes/localhost.yml
- targets:["localhost:9100"]
  labels:
    host: "localhost"
    #Permet de taguer la machine
```

Création d'un reseau interne

Dans le but de faire communiquer les machines ubuntu et debian.

Sous debian !

```
sudo nano /etc/network/interfaces
```

```
Fichier  Édition  Affichage  Terminal  Onglets  Aide
GNU nano 3.2 /etc/network/interfaces Modifié

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

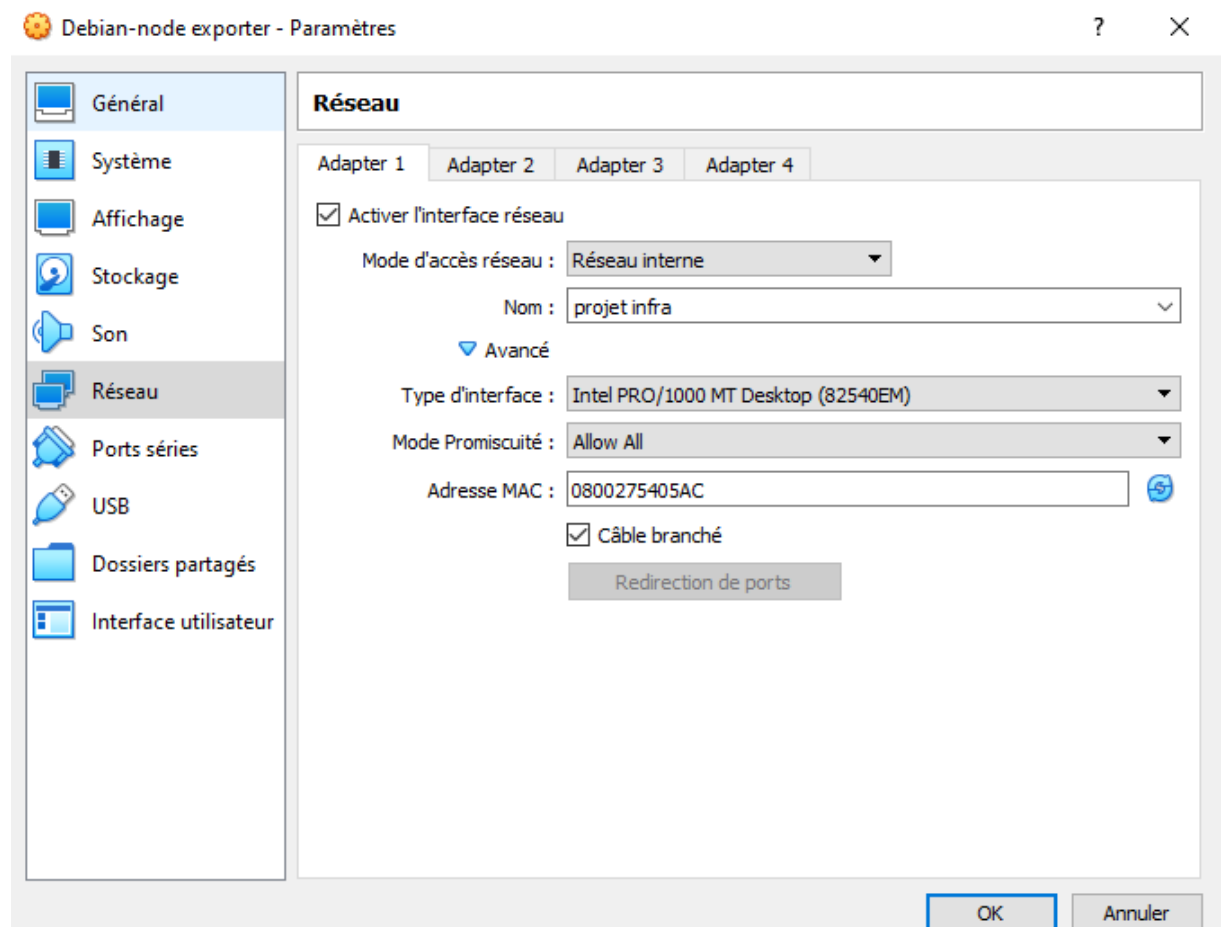
# The loopback network interface
auto lo
iface lo inet loopback

#The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet static
    address 192.168.1.2
    netmask 255.255.255.0
    gateway 192.168.1.1
```

Recharger la configuration ip :

```
/etc/init.d/networking restart
```


Crée également ce réseau sur VirtualBox :



Installation de grafana

Fonctionnement de Grafana:

Grafana est un logiciel Open Source pour la visualisation et la supervision d'une infrastructure

Il propose une liste de dashboards pré-générés pour récupérer les informations en provenance de Prometheus.

Il permet d'afficher graphiquement les métriques récupérées par Prometheus.

- **Rappel:** Les métriques sont un ensemble de données permettant la supervision et la récupération d'information sur un parc informatique à un intervalle de temps (comme par exemple la météo).

Installation de grafana

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

```
loan@loan-ubuntu:~$ sudo wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -
```

#permet d'obtenir une clé a la liste des clé de confiance ce qui permettra de télécharger et faire vérifier si il sagit du bon paquet.

Ajouter au référentiel grafana au source apt :

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

```
loan@loan-ubuntu:~$ sudo wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -
```

Faire la maj des paquets disponibles :

sudo apt update

```
loan@loan-ubuntu:~$ sudo apt update
```

Installer grafana :

sudo apt install grafana

```
loan@loan-ubuntu:~$ sudo apt install grafana
```

Lancer le serveur grafana :

```
sudo systemctl start grafana-server
```

```
loan@loan-ubuntu:~$ sudo systemctl start grafana-server
```

Affiche l'état du service (actif ou non) :

```
sudo systemctl status grafana-server
```

```
loan@loan-ubuntu:~$ 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 Tue 2022-04-26 14:55:20 CEST; 9s ago
     Docs: http://docs.grafana.org
    Main PID: 3938 (grafana-server)
      Tasks: 9 (limit: 2290)
     Memory: 89.1M
    CGroup: /system.slice/grafana-server.service
            └─3938 /usr/sbin/grafana-server --config=/etc/grafana/grafana.ini --pidfile=/run/grafana/grafana-server.pid

avril 26 14:55:27 loan-ubuntu grafana-server[3938]: logger=sqlstore t=2022-04-26T14:55:27.03+0200 lvl=info msg="Created"
avril 26 14:55:27 loan-ubuntu grafana-server[3938]: logger=plugin.manager t=2022-04-26T14:55:27.07+0200 lvl=info msg="Pl
avril 26 14:55:27 loan-ubuntu grafana-server[3938]: logger=plugin.finder t=2022-04-26T14:55:27.07+0200 lvl=warn msg="Sk
avril 26 14:55:27 loan-ubuntu grafana-server[3938]: logger=query_data t=2022-04-26T14:55:27.08+0200 lvl=info msg="Query
avril 26 14:55:27 loan-ubuntu grafana-server[3938]: logger=live.push_http t=2022-04-26T14:55:27.1+0200 lvl=info msg="Liv
avril 26 14:55:27 loan-ubuntu grafana-server[3938]: logger=server t=2022-04-26T14:55:27.3+0200 lvl=info msg="Writing PID
avril 26 14:55:27 loan-ubuntu grafana-server[3938]: logger=http.server t=2022-04-26T14:55:27.31+0200 lvl=info msg="HTTP
```

Lancer grafana automatiquement :

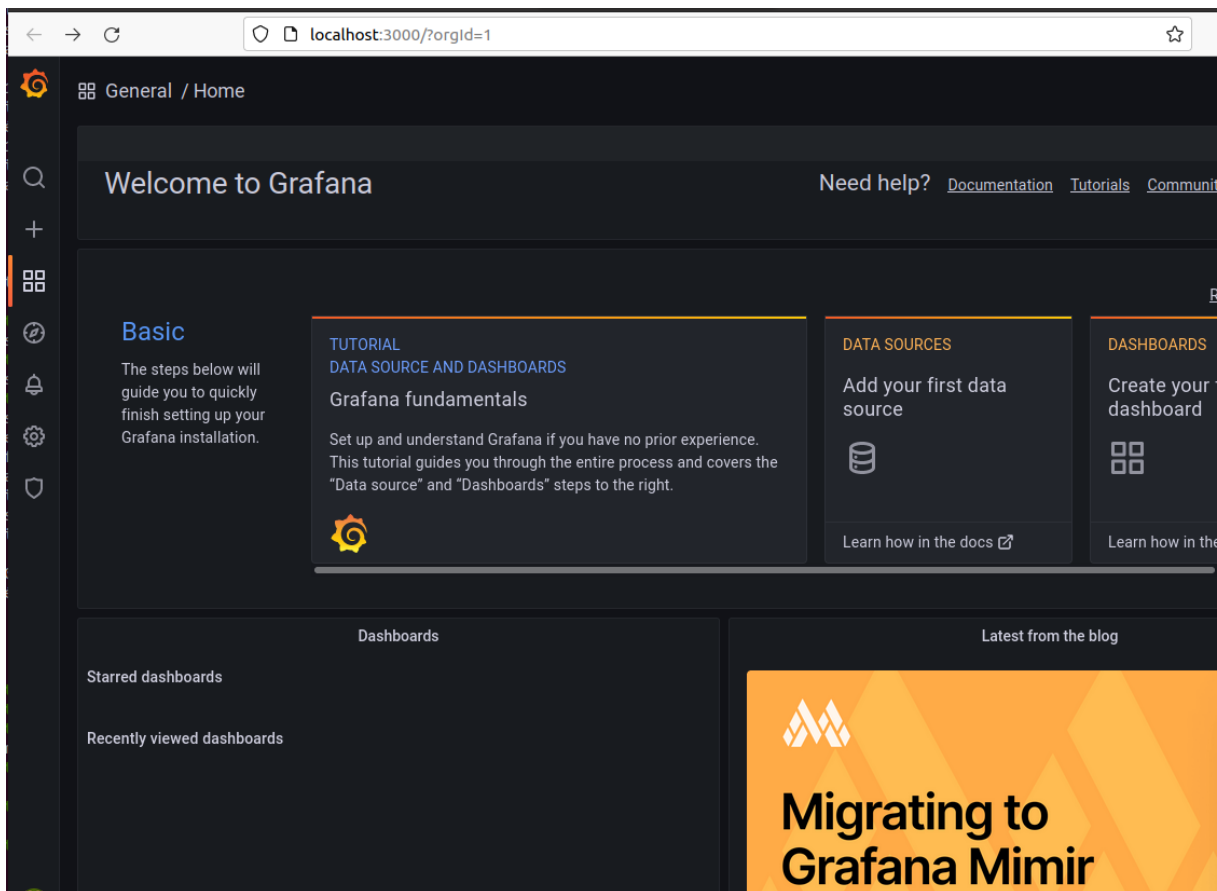
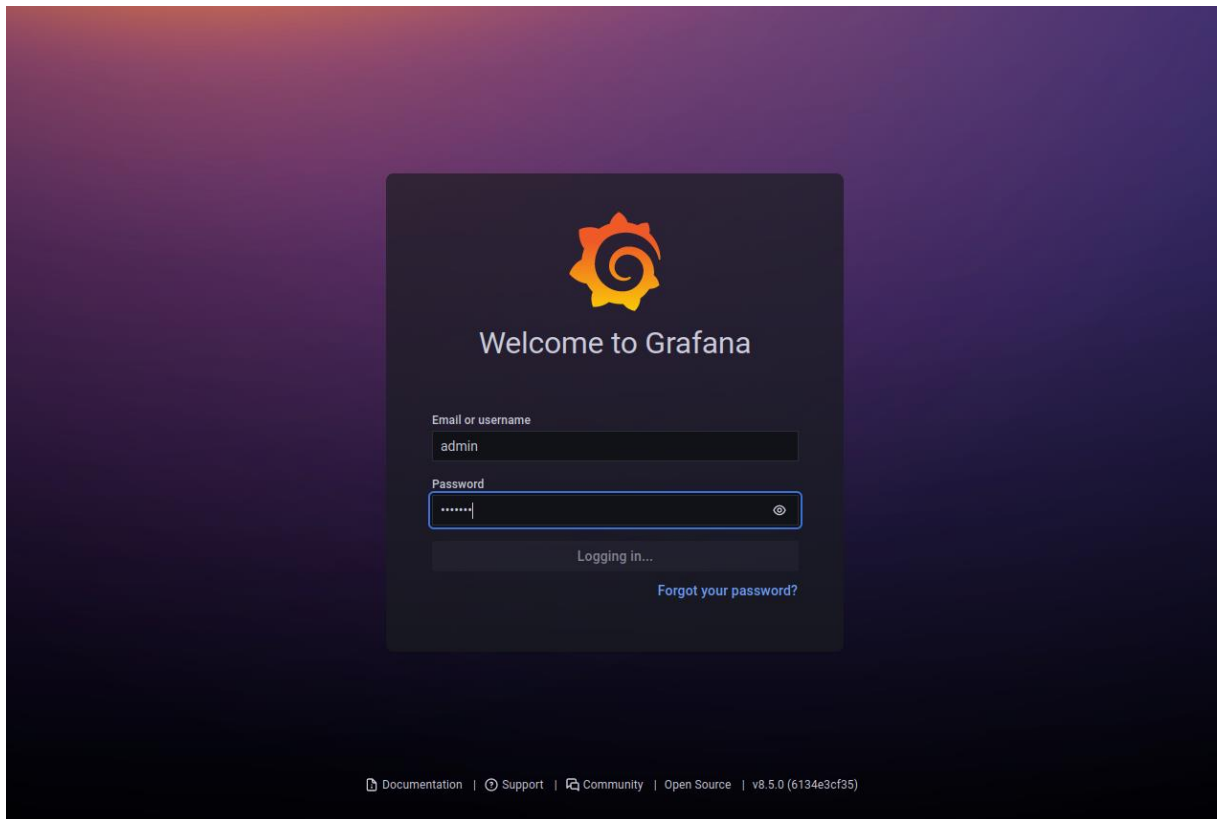
```
sudo systemctl enable grafana-server
```

```
loan@loan-ubuntu:~$ sudo systemctl enable grafana-server
```

PS : Nous n'avons pas besoin de créer un utilisateur dédié et déplacer les fichiers dans le bon répertoire car comme grafana a été installé depuis le cache apt de la machine les configurations ont déjà été renseignées par défaut

Accéder à grafana depuis le navigateur :

```
loan@loan-ubuntu:~/Documents/sauvegarde$ firefox http://localhost:3000/grafana
```



Configuration de prometheus :

Cette configuration permettra à prometheus de faire le lien entre les nodes présent sur les machines distantes et lui-même.

Modification du fichier de configuration de prometheus :

```
loan@loan-ubuntu:/usr/share/prometheus$ sudo nano prometheus_config.yml
```

Configuration par default :

```
global:
  scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
  # scrape_timeout is set to the global default (10s).

# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
      - targets:
        # - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9090"]
```

A ajouter au niveau du scrape_config :

```
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any
  - job_name: "prometheus"
  - job_name: "node"
    file_sd_configs:
      - files: ["/etc/prometheus/nodes/*.yml"]
```

Cette instruction indique que toutes les metrics seront récupérées et stockées dans le répertoire nodes et seront en .yml

Création du répertoire jobs :

```
loan@loan-ubuntu:/usr/share/prometheus$ sudo mkdir -p /etc/prometheus/nodes
```

Création d'un fichier .yaml permettant le lien avec le nodes exporter installé précédemment :

```
loan@loan-ubuntu:/etc/prometheus/nodes$ sudo nano debian_node-exporter.yaml
```

```
GNU nano 4.8 /etc/prometheus/nodes/debian_node-exporter.yaml
- targets: ["192.168.1.2:9100"]
  labels:
    #On peut ajouter le nom que l'on souhaite pour taguer la machine souhaiter
    host: "debian_node-exporter"
```

!!! Le 9100 est très important car cela indique à prometheus d'aller chercher les métriques sur le port de la machine cliente vu que le node laisse c'est métrique sur ce port

Inclure la machine prometheus dans le même réseau que la machine node :

Annuler Filaire Appliquer

Détails Identité **IPv4** IPv6 Sécurité

Méthode IPv4

- ☐ Automatique (DHCP)
- ☒ Manuel
- ☐ Partagée avec d'autres ordinateurs
- ☐ Réseau local seulement
- ☐ Désactiver

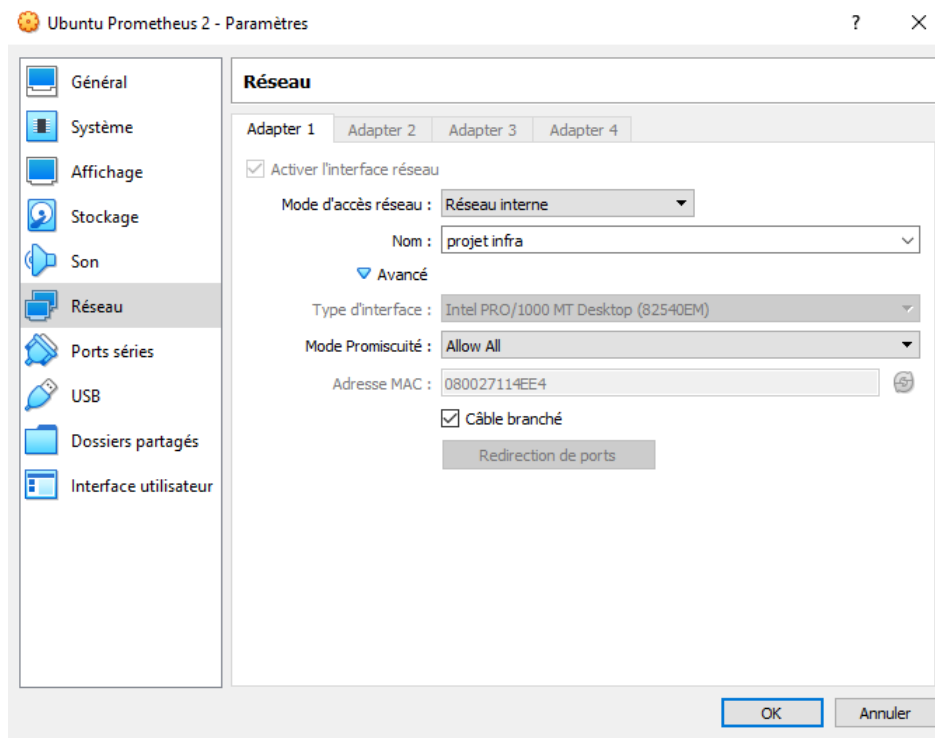
Adresses

Adresse	Masque de réseau	Passerelle	
192.168.1.1	255.255.255.0	192.168.1.1	

DNS Automatique ☒

Séparer les adresses IP avec des virgules

```
loan@loan-ubuntu: ~
loan@loan-ubuntu:~$ 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
        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 08:00:27:11:4e:e4 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.1/24 brd 192.168.1.255 scope global noprefixroute enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::d979:6256:b818:3e22/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
loan@loan-ubuntu:~$
```



Penser à redémarrer le service prometheus après avoir modifié les fichiers de configuration :

```
loan@loan-ubuntu:~$ sudo systemctl restart prometheus
```

Vérifier si prometheus récupère les métriques de la machine ou est présent le node exporter :

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	<code>instance="localhost:9090"</code> <code>job="node"</code>	1.660s ago	9.466ms	
http://192.168.1.2/metrics	DOWN	<code>host="debian_node-exporter"</code> <code>instance="192.168.1.2:80"</code> <code>job="node"</code>	11.9s ago	1.479ms	server returned HTTP status 404 Not Found

Installation d'apache2

Sur la machine node on installe le serveur apache2(debian) qui sera supervisé par prometheus grâce au node présent sur la machine debian :

```
loan@debianminatchy:~$ sudo apt install apache2
```

Redemarage du serveur apache -> lancement

```
loan@debianminatchy:~$ sudo systemctl daemon-reload
loan@debianminatchy:~$ sudo systemctl start apache2
loan@debianminatchy:~$ sudo systemctl status apache2
```

```
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese
   Active: active (running) since Tue 2022-04-26 11:51:22 CEST; 6h ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 516 (apache2)
    Tasks: 55 (limit: 4689)
   Memory: 11.6M
    CGroup: /system.slice/apache2.service
            └─516 /usr/sbin/apache2 -k start
              └─517 /usr/sbin/apache2 -k start
                └─518 /usr/sbin/apache2 -k start

avril 26 11:51:22 debianminatchy systemd[1]: Starting The Apache HTTP Server.
avril 26 11:51:22 debianminatchy systemd[1]: Started The Apache HTTP Server.
lines 1-14/14 (END)
```


Installation de Mysql

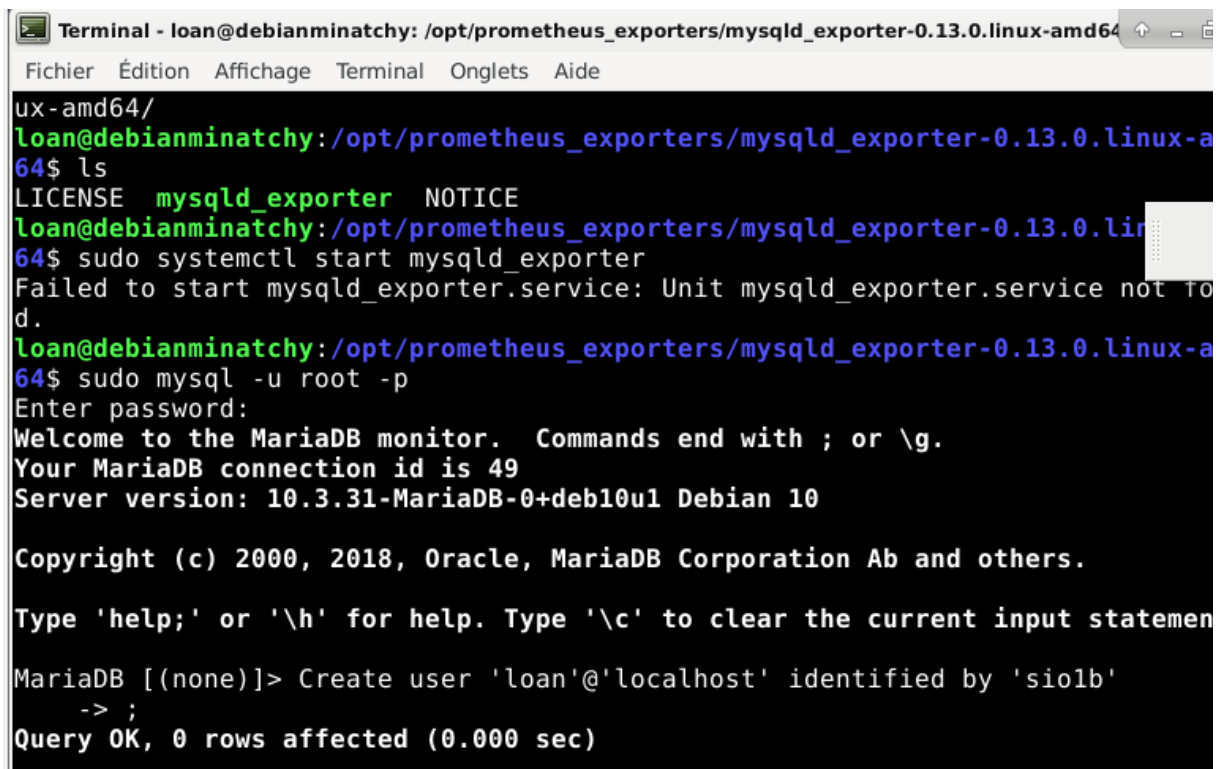
Chercher la présence du paquet :

```
sudo apt-cache mysql-server
```

Installation de mysql :

```
Sudo apt install default-mysql-server
```

Sudo mysql -u root -p permet de lancer mysql



```
Terminal - loan@debianminatchy: /opt/prometheus_exporters/mysql_exporter-0.13.0.linux-amd64
Fichier  Édition  Affichage  Terminal  Onglets  Aide
ux-amd64/
loan@debianminatchy:/opt/prometheus_exporters/mysql_exporter-0.13.0.linux-amd64$ ls
LICENSE  mysql_exporter  NOTICE
loan@debianminatchy:/opt/prometheus_exporters/mysql_exporter-0.13.0.linux-amd64$ sudo systemctl start mysql_exporter
Failed to start mysql_exporter.service: Unit mysql_exporter.service not found.
loan@debianminatchy:/opt/prometheus_exporters/mysql_exporter-0.13.0.linux-amd64$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 49
Server version: 10.3.31-MariaDB-0+deb10u1 Debian 10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> Create user 'loan'@'localhost' identified by 'sio1b'
-> ;
Query OK, 0 rows affected (0.000 sec)
```