

# Télémetrie Linux

Jean-Marc Pouchoulon

Mai 2025



## 1 Métriques systèmes sous Linux

### 1.1 Contexte

Le TP est à réaliser de manière individuelle sur une machine virtuelle Debian. Il a pour ambition de vous montrer comment récupérer des métriques et des logs du système d'exploitation Linux.

Les logiciels suivants seront utilisés:

- *Glances* est un outil de monitoring qui permet de visualiser les ressources système en temps réel. Il peut être utilisé en mode CLI ou via un navigateur web et peut exporter ses métriques vers *Influxdb* ou *Prometheus*.
- *Prometheus Node Exporter* est un agent qui collecte des métriques sur les systèmes d'exploitation et les expose à *Prometheus*. Il est conçu pour être léger et facile à déployer. Il collecte des métriques sur l'utilisation du CPU, de la mémoire, du disque et du réseau ...
- *Prometheus* est un système de surveillance et d'alerte open-source qui collecte des métriques à partir de services et d'applications. Il utilise un modèle de données basé sur des séries temporelles et permet de stocker et d'interroger ces données.
- *InfluxDB* est aussi une base de données qui est conçue pour stocker des données de séries temporelles. Elle est souvent utilisée pour stocker des métriques et des événements dans des applications de surveillance et d'analyse. Dans ce TP, il est important d'utiliser **la version 2 d'InfluxDB !**
- *Grafana* est un outil de visualisation de données qui permet de créer des tableaux de bord à partir de différentes sources de données, y compris *Prometheus* et *Influxdb*.

### 1.2 Installation et exploitation des outils pour cadvisor sous Linux

La version de *Glances* sous forme de "package" via "apt install" est trop ancienne pour réaliser l'export de Métriques. On va donc réaliser l'installation de *Glances* via *pipx*.

1. À l'aide de *pip*, installez *pipx*. Avec *pipx* installez *Glances* et vérifiez que l'installation s'est bien déroulée en lançant l'utilitaire.
2. Vérifiez que *Glances* fonctionne via le Cli ainsi que depuis un navigateur. Trouvez l'option pour que *Glances* exporte ses métriques vers *Prometheus*.
3. Installez *Grafana*, *Prometheus node exporter*, *Prometheus server* et *Influxdb2* sur la VM.
4. Vous "nattrez" les ports via virtualbox afin d'y accéder depuis votre machine physique.:
  - 3000 (*Grafana server*);
  - 3100 (*Loki*);
  - 8080 (*Logs*);
  - 8086 (*Influx*);
  - 8098 (*Cadvisor*);
  - 9090 (*Prometheus-server*);
  - 9100(*Prometheus Node Exporter*);

— 61208 (*Glances*);

Pour l'installation d'*Influxdb* reportez-vous à l'url suivante: <https://docs.influxdata.com/influxdb/v2/install/> et lancez influxsetup.

5. Configurer *Prometheus* afin qu'il puisse remonter les métriques de *Glances* et de *Prometheus Node Exporter*. Pour cela, il faut modifier le fichier de configuration de Prometheus qui est `/etc/prometheus/prometheus.yml`. (voir la section `scrape_configs`).
6. Quel est le sens des flux ? faites un schéma.
7. Vérifiez que les métriques sont bien exposées depuis les url des 3 services (*Prometheus*, *Glances*, *Prometheus Node Exporter*).
8. Afficher les métriques de *Glances* et de *Prometheus Node Exporter* sur *Grafana*.
9. Sur *Influxdb* (port 8086) créer un graphique du "cpu total".<sup>1</sup>
10. Installer le dashboard "*Node Exporter Full*" sur *Grafana*.

## 2 *Loki* n'est pas le dieu du mal mais des logs

On va maintenant installer *Loki* et *Promtail*. *Loki* est un système de gestion de logs développé par *Grafana Labs*. *Promtail* est un agent qui collecte les logs et les envoie à *Loki*.

*Loki* est conçu pour être léger et facile à utiliser, et il s'intègre bien avec *Grafana* pour la visualisation des logs. Il utilise un modèle de données basé sur des séries temporelles, similaire à celui de Prometheus, mais il est optimisé pour le stockage et la recherche de logs.

### 2.1 Avec *Docker* c'est plus rapide...

On va utiliser Docker pour installer *Loki* et *promtail*. On va utiliser docker-compose pour simplifier le processus d'installation. Il vous faudra arrêter donc arrêter le service *Grafana* de la machine hôte avant de lancer les containers pour éviter les conflits de ports.

```
systemctl stop grafana-server.service
systemctl disable grafana-server.service
```

1. Utilisation d'un fichier *docker-compose.yml* pour la gestion des logs d'un container nginx.

Clonez le dépôt git suivant et installez les containers avec "docker compose".

```
git clone https://github.com/pushou/docker-promtail-loki.git
cd docker-promtail-loki
docker-compose up -d
```

Vérifiez que les containers sont bien lancés et que les logs de *nginx* sont bien envoyés à *Loki*. Générer des logs dans *nginx* sur <http://localhost:8080> et vérifiez que les logs sont bien envoyés à *Loki*.

2. Affichez les logs de *nginx* dans *Grafana*:

---

1. Le dashboard *Grafana* "glances 2024" ne fonctionne pas !

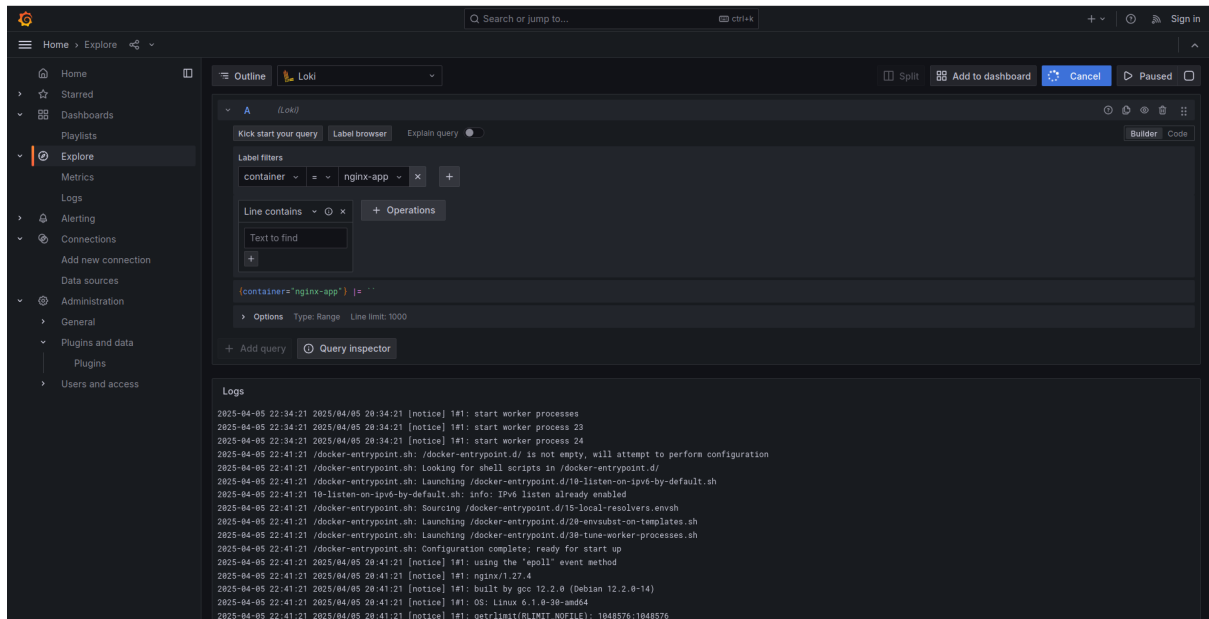


FIGURE 1 – Visualisation des logs de nginx dans Grafana

## 2.2 Affichage des métriques produites par *Cadvisor* dans *Grafana*

On se propose maintenant de visualiser à l'aide de *Grafana* les métriques remontées par *cadvisor* sur les conteneurs *Docker* en cours d'exécution. Pour cela, modifiez le fichier *docker-compose.yml* pour ajouter un service *Cadvisor*.

```
prometheus:
  image: quay.io/prometheus/prometheus:latest
  volumes:collecte
  des métriques
    - ./prometheus/./etc/prometheus/
    - prometheus_data:/prometheus
  command:
    - '--config.file=/etc/prometheus/prometheus.yml'
    - '--storage.tsdb.path=/prometheus'
    - '--web.console.libraries=/usr/share/prometheus/console_libraries'
    - '--web.console.templates=/usr/share/prometheus/consoles'
  ports:
    - 9090:9090
  links:
    - cadvisor:cadvisor
    - alertmanager:alertmanager
#   - pushgateway:pushgateway
  depends_on:
    - cadvisor
#   - pushgateway
  networks:
    - back-tier
  restart: always

cadvisor:
  image: registry.iutbeziers.fr/cadvisor:v0.52.1
  platform: linux/amd64
  container_name: cadvisor
  ports:
```

```

- 8098:8080
expose:
- 8080
volumes:
- /:/rootfs:ro
- /var/run:/var/run:ro
- /sys:/sys:ro
- /var/lib/docker:/var/lib/docker:ro
- /dev/disk:/dev/disk:ro
devices:
- /dev/kmsg
privileged: true
restart: unless-stopped
networks:
- app

```

Rajoutez un "scrapper" pour *cadvisor* dans le fichier de configuration de *Prometheus*.

```

- job_name: 'cadvisor'
  scrape_interval: 5s
  static_configs:
    - targets: ['cadvisor:8080']

- job_name: 'node'
  # If prometheus-node-exporter is installed, grab stats about the local
  # machine by default.
  static_configs:
    - targets: ['prometheus:9100']

```

Dans *Grafana* rajoutez *Prometheus* comme data source et rajoutez un dashboard pour *cadvisor* (numéro 19792).

Enjoy !

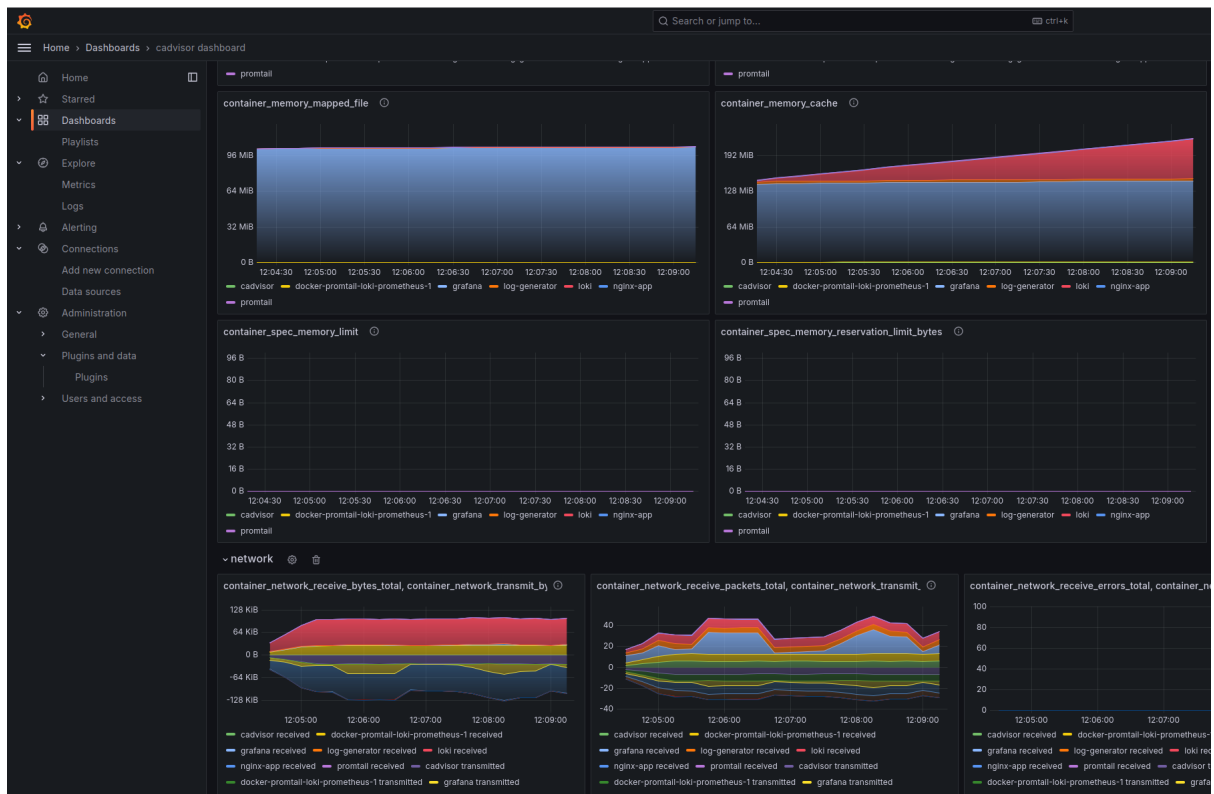


FIGURE 2 – Visualisation des métriques cadvisor dans Grafana