

TP 3 – Premier pas avec le monitoring

Objectif général

Rendre concrets les concepts vus en cours (monitoring) au travers d'exercices graduels. Vous serez capable de lancer votre première stack de monitoring basée sur prometheus, loki, grafana et de créer vos premières alertes.

L'ensemble des exercices suivant doit se faire sur une machine Linux (VM du premier cours, Kali, poste personnel, ...)

Exercice 1 – Installation de Prometheus

Objectifs :

- Avoir accès au dashboard prometheus.

Outils conseillés : docker, docker-compose, prometheus

Durée indicative : 10-20 min

Consignes

1. Vérifier que docker et docker-compose sont bien installés (cf doc officielle : <https://docs.docker.com/engine/install/>)
2. Créez un dossier « monitoring » pour contenir vos différents volumes.
3. Créer un fichier docker-compose-prometheus.yml avec un service Prometheus pour que celui-ci soit accessible sur le port 9090. Vous devrez également monter dans votre conteneur le fichier `./volumes/prometheus.yml`.

Vous devriez avoir cette arborescence :

```
user@debian:~$ tree monitoring/  
monitoring/  
├── docker-compose-prometheus.yml  
└── volumes  
    └── prometheus.yml
```

4. Lancer prometheus et essayez d'y accéder via votre navigateur :

```
docker compose -f docker-compose-prometheus.yml up -d
```

Liens utiles

- <https://mxulises.medium.com/simple-prometheus-setup-on-docker-compose-f702d5f98579>

Exercice 2 – Surveiller son linux avec Prometheus

Objectifs :

- Prometheus collecte les métriques système (CPU, RAM, etc.).

Outils conseillés : prometheus

Durée indicative : 10-20 min

Consignes

1. Ajouter le service node-exporter au docker-compose-prometheus.yml.
2. Configurer Prometheus (prometheus.yml) pour ajouter les métriques du node-exporter.
3. Vérifier dans l'interface Prometheus (onglet "Status > Target health") que la cible est "UP".
4. Vérifiez que la query « node_memory_MemFree_bytes » nous donne des résultats

Liens utiles

- <https://mxulises.medium.com/simple-prometheus-setup-on-docker-compose-f702d5f98579>

Exercice 3 – Installation de grafana et connecter prometheus

Objectifs :

- Grafana est connecté à Prometheus.
- Avoir un dashboard complet affichant les métriques système.

Outils conseillés : prometheus, grafana

Durée indicative : 20-30 min

Consignes

1. Créer un fichier docker-compose-grafana.yml avec un service grafana écoutant sur le port 3000.
2. Lancer Grafana et accéder à son interface web (login: admin/admin).
3. Ajouter Prometheus comme source de données.
4. Importer le dashboard officiel "Node Exporter Full" (ID 1860).
5. Créer un nouveau dashboard.
6. Ajouter un panel avec la requête rate(node_cpu_seconds_total[1m]).
7. Personnaliser le titre, les axes et les couleurs.

Liens utiles

- <https://grafana.com/grafana/dashboards/1860-node-exporter-full/>

Exercice 4 – Centralisation des logs avec Loki

Objectifs :

- Loki est accessible et stocke des logs.
- Affichage et filtrage des logs dans Grafana.
- Avoir un dashboard combinant métriques et logs.

Outils conseillés : loki, grafana

Durée indicative : 30-40 min

Consignes

1. Créer un fichier docker-compose-loki.yml avec les services loki et promtails
2. Lancer les services et vérifier que Loki reçoit des logs.

Liens utiles

- <https://www.shpv.fr/blog/stack-logging-loki/>
- <https://medium.com/@netopschic/implementing-the-log-monitoring-stack-using-promtail-loki-and-grafana-using-docker-compose-bcb07d1a51aa>