

Prérequis :

- Centre d'observabilité fonctionnelle
- Vm debian 11, 12 ou Ubuntu avec service ssh

Configuration recommandée de la VM VirtualBox

Élément	Recommandation	Raison
Nom	vm-observed	Cible supervisée
OS	Debian 11 / 12 ou Ubuntu Server 22.04 LTS	Stable, léger, compatible avec node_exporter, Promtail, etc.
Réseau	Bridged Adapter ou Réseau privé hôte (Host-Only) combiné avec NAT pour avoir internet	Permet à Prometheus/Loki (sur ton hôte) d'accéder à la VM
RAM	2 Go minimum	Suffisant pour les tests observabilité
CPU	2 cœurs	Permet de tester la charge
Disque	10–20 Go	Assez pour logs + services
Utilisateur	admin (sudoer)	Pour les installations
Accès SSH	Oui (port 22 ouvert)	Pour automatisation future via Ansible

Étapes d'installation

1. Créer la VM

- Crée une nouvelle VM dans VirtualBox.
- Monte l'ISO Debian ou Ubuntu.
- Installe le système (SSH activé, user sudoer).

2. Configurer le réseau

- Si on veut que **le Docker Compose (centre observabilité)** sur la machine hôte Windows accède à la VM :
 - Mets l'adaptateur en “**Bridged**” → ta VM aura une IP du même réseau que ton PC.
 - Ou “**Host-only Adapter**” si tu veux un réseau isolé local , ensuite rajoute un adaptateur que tu mettras en NAT .
- Vérifie avec :
- ip a

puis ping depuis le cmd de ta machine hôte :

ping 192.168.x.x

3. Vérifier SSH

- Vérifie que SSH tourne :
sudo systemctl status ssh
- Si besoin, démarre-le :
sudo systemctl enable --now ssh
- Depuis ton PC teste la connexion :
ssh admin@192.168.x.x

Si on peut se connecter alors le SSH est bien activé et la VM est prête à être supervisée.

Installe les outils nécessaires dans la VM

1. Node exporter (métriques système)

```
sudo apt update && sudo apt install wget -y
```

```
wget https://github.com/prometheus/node_exporter/releases/download/v1.7.0/node_exporter-1.7.0.linux-amd64.tar.gz
```

```
tar xvf node_exporter-1.7.0.linux-amd64.tar.gz
```

```
sudo mv node_exporter-1.7.0.linux-amd64/node_exporter /usr/local/bin/
```

```
sudo useradd -rs /bin/false nodeusr
```

- Créer un service :

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

Contenu :

```
[Unit]
```

```
Description=Node Exporter
```

```
After=network.target
```

```
[Service]
```

```
User=nodeusr
```

```
ExecStart=/usr/local/bin/node_exporter
```

```
[Install]
```

```
WantedBy=default.target
```

Active :

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable --now node_exporter
```

Teste :

```
curl http://localhost:9100/metrics
```

2. Promtail (collecte des logs)

Installe Promtail (même version que ton Loki) :

```
wget https://github.com/grafana/loki/releases/download/v2.9.2/promtail-linux-amd64.zip  
unzip promtail-linux-amd64.zip  
sudo mv promtail-linux-amd64 /usr/local/bin/promtail
```

Crée le fichier de config :

```
sudo nano /etc/promtail-config.yaml
```

Contenu minimal :

server:

```
http_listen_port: 9080  
grpc_listen_port: 0
```

positions:

```
filename: /tmp/positions.yaml
```

clients:

```
- url: http://<IP_HOTE>:3100/loki/api/v1/push # IP de ta machine hôte
```

scrape_configs:

```
- job_name: system
```

static_configs:

```
- targets:
```

```
  - localhost
```

labels:

```
  job: vm-observed
```

```
  host: debian-vm
```

```
  __path__: /var/log/*.log
```

Service Promtail :

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

Contenu :

[Unit]

Description=Promtail

After=network.target

[Service]

ExecStart=/usr/local/bin/promtail -config.file=/etc/promtail-config.yml

Restart=always

[Install]

WantedBy=multi-user.target

Active :

sudo systemctl daemon-reload

sudo systemctl enable --now promtail

3. Vérifie la connexion à ton centre observabilité

- Loki sur ton hôte : http://localhost:3100
- Prometheus sur ton hôte : http://localhost:9090
- Grafana sur ton hôte : http://localhost:3000

Teste depuis ta VM :

curl http://<IP_HOTE>:3100/ready

curl http://<IP_HOTE>:9090/metrics

Si ça répond → ta VM communique bien avec le centre.

4. Ajoute ta VM dans Prometheus

Sur ton hôte, dans prometheus.yml :

```
- job_name: 'vms'  
  static_configs:  
    - targets: ['192.168.x.x:9100'] => mettre l'ip de la vm
```

Puis reload Prometheus :

docker-compose restart prometheus

5. Vérifie dans Grafana

- Data sources : Prometheus + Loki.
 - Explore :
 - {job="vm-observed"} pour logs Promtail.
 - node_load1 pour métriques Prometheus.
-

Résultat attendu

Tu auras :

- Une **VM Debian/Ubuntu** supervisée par ton centre d'observabilité.
 - **Métriques système** (CPU, RAM, disque).
 - **Logs système** collectés dans Loki.
 - Un premier **dashboard Grafana** “VM observée”.
-

Ensuite, tu pourras ajouter **d'autres VMs** ou **importer ces configs via Ansible** pour étendre ton périmètre.

6. Vérifie que ta VM apparaît bien

Ouvre ton navigateur sur :

<http://localhost:9090/targets>

On devrait voir un bloc :

```
job_name: vms
target: 192.168.4.xx:9100
last scrape: <quelques secondes>
state: UP
```

Si on voit UP, c'est que Prometheus collecte bien les métriques de la vm.

Si c'est marqué DOWN :

- Vérifie que node_exporter tourne sur la vm :
sudo systemctl status node_exporter
- Vérifie la connectivité :
curl http://192.168.4.xx:9100/metrics
- Si ça renvoie des lignes, le problème est juste dans le prometheus.yml (erreur d'adresse ou de format).