

3MIN

Nous monitorons l'infrastructure avec des outils **open source**, principalement la suite **Grafana + Prometheus**.

Services utilisés

Service	Description
Prometheus	Collecte les métriques via différents exporters.
Grafana	Visualisation des métriques à travers des dashboards personnalisés.
Node Exporter	Suivi des ressources système (CPU, RAM, disque).
Blackbox Exporter	Vérification de la disponibilité des services via HTTP, ICMP ou TCP.
Keepalived Exporter	Suivi des basculements de haute-disponibilité via VRRP.
Nginx / Apache Exporter	Métriques pour les serveurs web.
Mysqld Exporter	Surveillance des performances des bases de données MySQL.

✨ Prérequis

- Renseignez les **adresses IP** de vos services dans les fichiers `targets.*.json` selon les exporters installés.

Exemple :

```
192.168.1.10 = apache_exporter → à mettre dans targets.apache.json
```

- Remplissez les identifiants Grafana (`.env`) :

```
GRAFANA_USER=admin
GRAFANA_PASSWORD=*****
```

- Pour déployer les exporters sur vos serveurs :

```
cd exporter
./install_node_exporter.sh
./install_apache_exporter.sh
# etc.
```

⚠ **N'oubliez pas d'initialiser les variables dans vos scripts**, dans le dossier `exporter`.

- Pour la configuration des alertes, suivez la documentation officielle PagerDuty :



[Intégration Prometheus - PagerDuty](#)

Lancer le projet

Les services sont orchestrés avec **Docker Compose**.

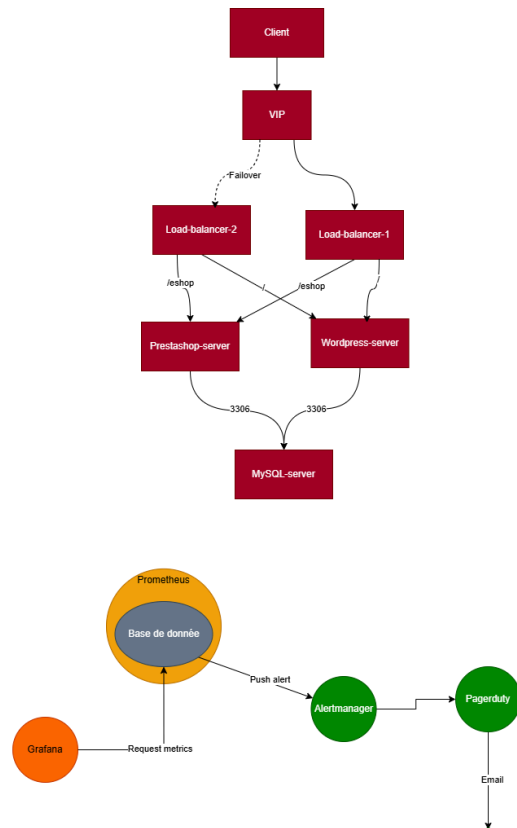
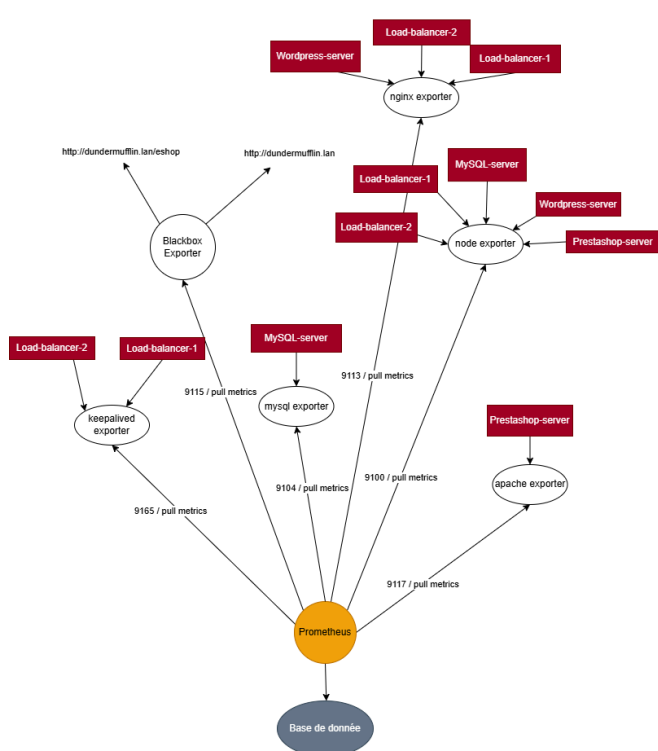
Exécutez à la racine du projet :

```
docker compose up -d
```

Accès aux interfaces :

- **Grafana** : <http://localhost:3000>
- **Prometheus** : <http://localhost:9090>

Schéma d'infrastructure



Ajouter ou modifier des targets

1. Arrêtez les services :

```
docker compose down
```

2. Mettez à jour vos fichiers `targets.*.json` .

3. Redémarrez :

```
docker compose up -d
```