

Contents

Présentation du projet	2
Prérequis	2
Structure du projet	2
Démarrage rapide	3
A. Démarrage automatique (recommandé)	3
B. Démarrage manuel	4
Accès aux interfaces	4
Utilisation des dashboards Grafana	4
Générer du trafic et tester les alertes	5
Test manuel	5
B. Test automatique	5
Alertes et notifications	6
Ajouter/modifier des dashboards Grafana	6
Déploiement sur un autre environnement	7
Bonnes pratiques & sécurité	7
Dépannage	7
Voir les logs d'un service	7
Documentation complémentaire	8
Pour aller plus loin	8
Pour toute question ou contribution, consulte le README ou ouvre une issue sur le dépôt GitHub.	8

Guide d'utilisation professionnel et détaillé monitoring avec Prometheus, Grafana et Flask.

Présentation du projet

Ce projet propose une solution clé-en-main pour superviser une application web (Flask) et son infrastructure Docker grâce à Prometheus (collecte de métriques), Grafana (visualisation), cAdvisor (métriques conteneurs), Node Exporter (métriques système) et Alertmanager (alertes).

Prérequis

Docker et Docker Compose installés ([télécharger ici](https://docs.docker.com/get-docker/))

Python 3.8+ (pour les scripts de test de charge)

Accès aux ports : 3000 (Grafana), 9090 (Prometheus), 9093 (Alertmanager), 5001 (Flask), 9100 (Node Exporter), 8080 (cAdvisor)

Structure du projet



— app.py	Application Flask instrumentée
— docker-compose.yml	Orchestration de la stack
— Dockerfile	Image de l'app Flask
— prometheus.yml	Config Prometheus (scraping, alertes)
— prometheus-rules.yml	Règles d'alerting Prometheus
— alertmanager.yml	Config Alertmanager
— dashboards/	Dashboards Grafana (JSON)
— datasources/	Datasource Grafana (Prometheus)
— requirements.txt	Dépendances Python
— start.sh / start.bat	Scripts de démarrage automatisés
— load-test.py	Script de test de charge
— Document/	(Branche docs) Rapport PDF
— README.md	Documentation rapide

Démarrage rapide

A. Démarrage automatique (recommandé)

Sous Windows Double-clique sur *start.bat* ou Exécute dans un terminal :
start.bat

Sous Linux/Mac Rends le script exécutable puis lance-le :

```
chmod +x start.sh
```

```
./start.sh
```

Le script :

- 🚦 Vérifie Docker et Docker Compose
- 🚦 Génère un *.env* sécurisé si besoin
- 🚦 Lance tous les services
- 🚦 Affiche les liens d'accès

B. Démarrage manuel

Exécute dans un terminal :

docker-compose up -d --build

Accès aux interfaces

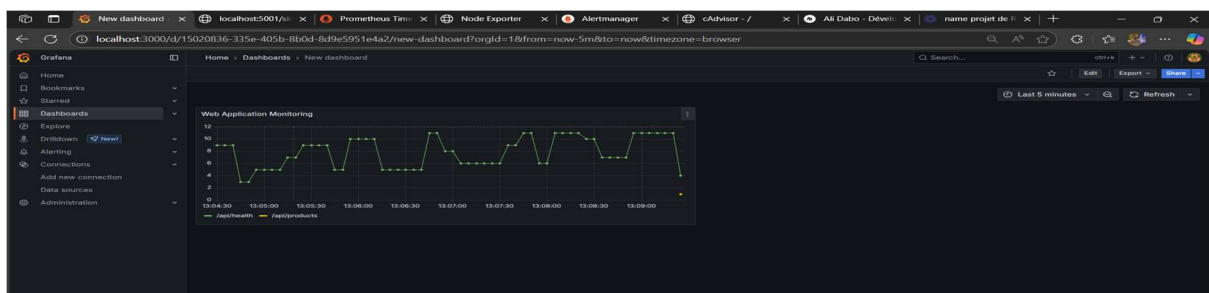
Service	URL	Identifiants par défaut
Grafana	http://localhost:3000	admin / admin
Prometheus	http://localhost:9090	
Alertmanager	http://localhost:9093	
Flask App	http://localhost:5001	
Node Exporter	http://localhost:9100	
cAdvisor	http://localhost:8080	

Utilisation des dashboards Grafana

🚦 Le dashboard principal est **provisionné automatiquement** (pas besoin de le créer à la main).

🚦 Pour y accéder :

1. Connecte-toi à Grafana
2. Clique sur “Dashboards” > “Manage”
3. Ouvre “Web Application Monitoring Dashboard”



Générer du trafic et tester les alertes

Test manuel

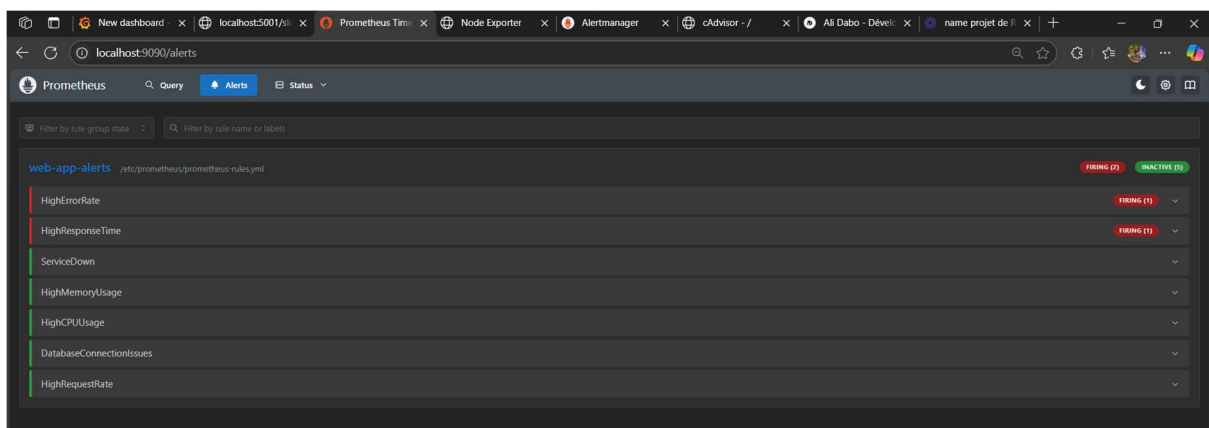
- Navigue sur les endpoints de l'app Flask :

- `/`
- `/api/health`
- `/api/users`
- `/api/products`
- `/api/orders`
- `/metrics`
- `/error` (pour simuler une erreur)
- `/slow` (pour simuler une lenteur)

B. Test automatique

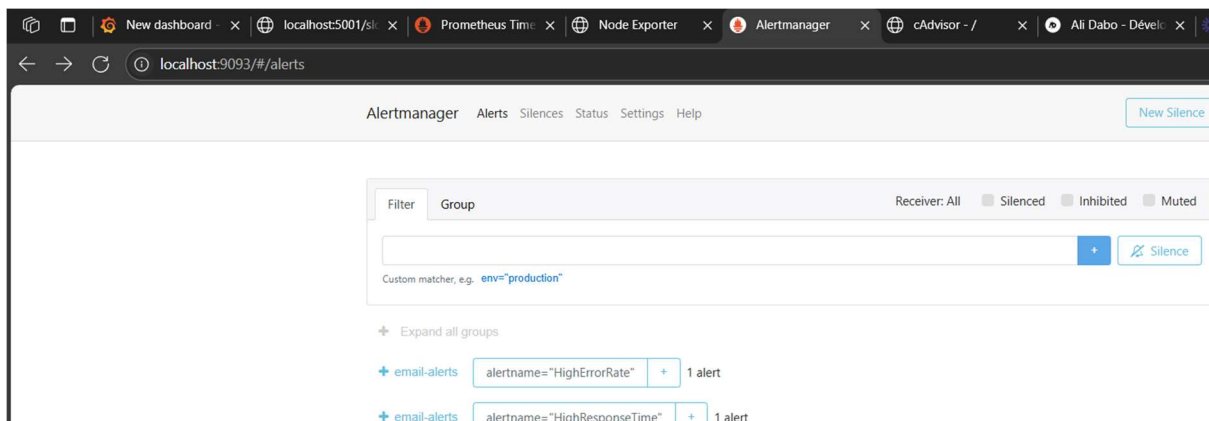
- Lance le script de charge :

```
python load-test.py --duration 60 --rps 10 --threads 4
```



Alertes et notifications

- Les alertes sont configurées dans *prometheus-rules.yml* (erreurs, lenteurs, ressources...)
- Alertmanager peut être relié à Slack, email, webhook, etc.
- Les alertes apparaissent dans l'interface Alertmanager et peuvent être visualisées dans Grafana.



Ajouter/modifier des dashboards Grafana

- Place tes fichiers JSON dans dashboards/
- Modifie `dashboards/dashboard.yml` si besoin
- Redémarre Grafana pour appliquer les changements

Déploiement sur un autre environnement

Clone le repo :

```
git clone https://github.com/alsondab/Prometheus_Grafana_Web_Flask.git
```

```
cd Prometheus_Grafana_Web_Flask
```

- Lance le script de démarrage ou **docker-compose up -d**
- Adapte le fichier `.env` si besoin (clé PROMETHEUS_HEX)

Bonnes pratiques & sécurité

Ne partage jamais ton fichier `.env` en public

- 🛠 Change le mot de passe admin Grafana en production
- 🛠 Utilise HTTPS pour exposer Grafana/Prometheus en production
- 🛠 Mets à jour régulièrement les images Docker

Dépannage

Voir les logs d'un service:

```
docker-compose logs flask-app
```

```
docker-compose logs prometheus
```

```
docker-compose logs grafana
```

Redémarrer un service:

```
docker-compose restart grafana
```

Vérifier l'état des services:

```
docker-compose ps
```

Documentation complémentaire

- ✚ [Documentation Prometheus](https://prometheus.io/docs/)
- ✚ [Documentation Grafana](https://grafana.com/docs/)
- ✚ [Documentation Docker Compose](https://docs.docker.com/compose/)

Pour aller plus loin

- ✚ Intégration CI/CD (GitHub Actions, GitLab CI...)
- ✚ Déploiement sur le cloud (AWS, Azure, GCP)
- ✚ Supervision multi-applications ou microservices
- ✚ Ajout de nouveaux exporters (MySQL, Redis, etc.)

Pour toute question ou contribution, consulte le README ou ouvre une issue sur le dépôt GitHub.