

Rapport Technique - Projet CRV_TME

Autoscaling & Observabilité avec Kubernetes, Prometheus & Grafana

Brahim Halloucha

April 14, 2025

Lien du projet GitHub : https://github.com/brahimhalloucha/crv_tme

1 Objectifs du Projet

L'objectif de ce projet était de mettre en place une infrastructure conteneurisée et scalable pour une application web composée :

- d'un backend **Node.js** avec cache **Redis**,
- d'un frontend **React**,
- d'un système de monitoring avec **Prometheus** et **Grafana**,
- le tout orchestré par **Kubernetes**, en local via **Minikube**.

2 Choix Techniques & Justifications

Composant	Choix	Justification
Orchestration	Kubernetes (Minikube)	Puissance et popularité dans le monde DevOps
Backend	Node.js + Redis	Redis utilisé pour accélérer les lectures de données
Frontend	React	Application légère, SPA moderne
Monitoring	Prometheus + Grafana	Standards de monitoring cloud-native
Exposition publique	LocalTunnel	Alternative simple pour exposer localement vers Internet

3 Conteneurisation

Backend

- Dockerfile Node.js configuré pour exécuter une API avec accès Redis.
- Image nommée : **backend-img**
- Disponible sur Docker Hub : **brahimhalloucha/backend-img**

Frontend

- Dockerfile React servant l'application via un serveur léger.
- Image nommée : **frontend-img** 1
- Disponible sur Docker Hub : **brahimhalloucha/frontend-img**

Décision : Images Docker disponibles en ligne pour simplifier le déploiement sans

5 Monitoring avec Prometheus & Grafana

Prometheus

Déploiement à l'aide de 3 fichiers :

- `prometheus-config.yaml`
- `prometheus-deployment.yaml`
- `prometheus-service.yaml`

Grafana

- Déployé avec `grafana-deployment.yaml`
- Exposé via NodePort

Configuration de Grafana

- Connexion Prometheus comme source de données
- Dashboards importés :
 - Node Exporter Full (ID : 1860)
 - Kubernetes Cluster Monitoring (ID : 6417)

Décision : Monitoring centralisé pour visualiser l'état des pods, containers et performances réseau.

6 Exposition externe via LocalTunnel

Pour permettre l'accès à l'API backend depuis l'extérieur :

- Script `start-localtunnel.sh` écrit pour lancer automatiquement le tunnel avec nom personnalisé : `backend-crv.local.lt`

7 Example

8 Tests & Vérifications

- Accès aux services testés avec `minikube service <name>`
- Visibilité confirmée dans Prometheus et Grafana
- LocalTunnel testé via navigateur externe

New key

key:	<input type="text" value="ee"/>
Value:	<input type="text" value="ee"/>
add	<input type="button" value=""/>

FIGURE 1 – Capture d'écran de l'interface Grafana en fonctionnement