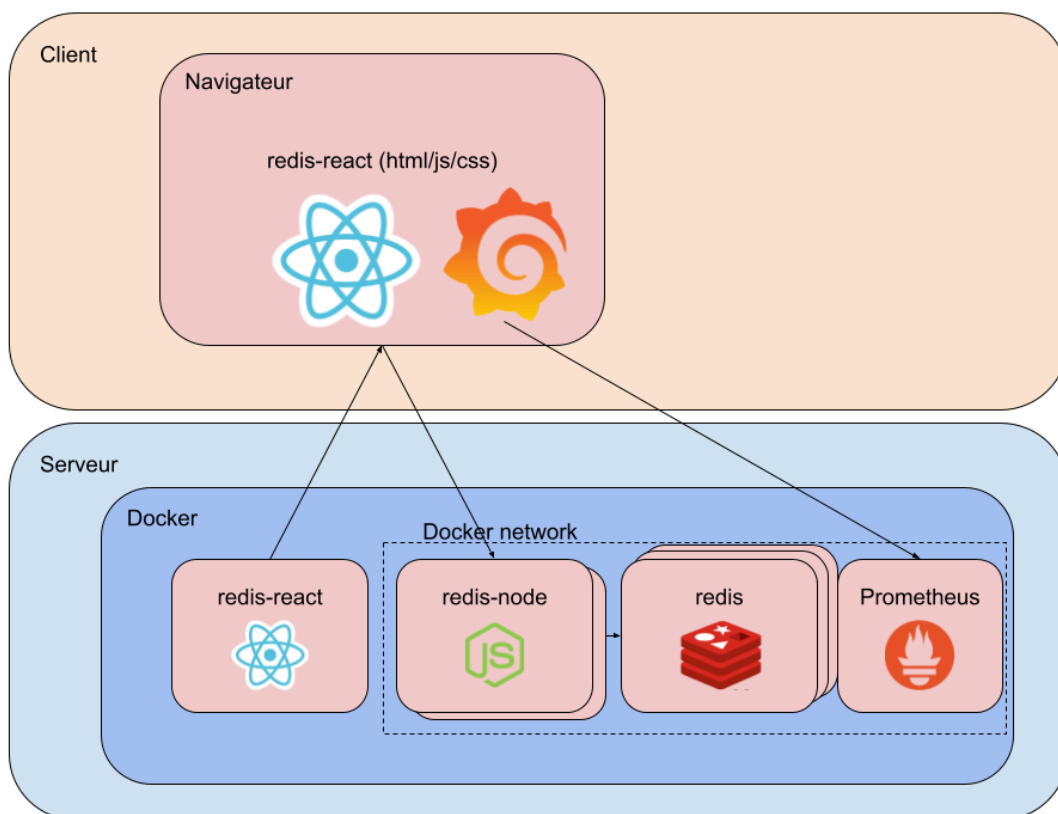


Projet Autoscaling et IaC

L'objectif de ce projet est de déployer une application web sur un cluster Kubernetes. Cette application est composée d'un frontend en React, un serveur backend en nodeJS, et une base de données Redis. De plus, l'outil de monitoring Prometheus/grafana devra être déployé et connecté à l'API /metrics de nodeJS. Enfin, une mise à l'échelle automatique du serveur et de la base de données seront configurés.

Ci-dessous l'infrastructure demandée :

L'infrastructure



Afin de mettre en place cette infrastructure, nous avons dû créer au total 6 fichiers de configuration yaml (redis.yaml, nodeJS.yaml, frontend.yaml, prometheus.yaml, grafana.yaml, et autoScaling.yaml). Ces fichiers sont utilisés pour déployer dans notre cluster minikube l'ensemble des pods et services nécessaires.

Davy TSE - 3672079
Ounas YHIA - 21312539

Le rôle de chaque fichier dans notre infrastructure est le suivant :

- ❖ nodeJS.yaml : Permet de déployer les serveurs nodeJS stateless. Ces derniers sont accessibles via un service d'équilibrage de charge.
- ❖ redis.yaml : Permet de déployer la base de données Redis implémentant le pattern main/replicas, également accessible via un service spécifique.
- ❖ autoScaling.yaml : Permet la mise en place de l'auto-scaling (montée à l'échelle automatique) sur les serveurs nodeJS (de 3 à 10 réplicas) ainsi que sur les réplicas Redis (de 1 à 6 réplicas)
- ❖ prometheus.yaml : Permet de déployer l'outil de monitoring prometheus. Celui-ci est connecté aux serveurs nodeJS via son API /metrics.
- ❖ grafana.yaml : Permet de déployer le tableau de bord grafana personnalisé afin d'afficher les données récoltées par prometheus sous forme de graphs.
- ❖ frontend.yaml : Permet de déployer le front-end de l'application.

Connexion entre les différents composants de notre application au sein de notre Cluster minikube :

- Les serveurs nodeJS sont connectés à la base de donnée Redis main via un service de type ClusterIP.
- Le frontend en React est connecté aux serveurs nodeJS via un tunnel minikube. Ce tunnel permet de relier le pod contenant le frontend au service de type LoadBalancer des serveurs nodeJS.
- Le frontend est accessible à l'extérieur du cluster via un tunnel http Tunnelmole.

Déployer l'application dans un cluster minikube à l'aide du script d'automatisation :

1. Pour déployer l'application de manière totalement automatisée, il suffit d'exécuter le script **script.sh**. Ce dernier permet de déployer l'ensemble des composants de l'application web dans le cluster minikube en respectant l'architecture présentée un peu plus haut. Le tableau de bord grafana est automatiquement lancé et le lien permettant d'accéder au site est affiché sur le terminal.

Ci-dessous le résultat affiché lors de l'exécution du script :

```
etudiant@sirius:~/Téléchargements/PROJET_Ounas-YHIA_Davy-TSE$ ls
autoScaling.yaml frontend.yaml grafana.yaml nodeJS.yaml prometheus.yaml redis.yaml script.sh
etudiant@sirius:~/Téléchargements/PROJET_Ounas-YHIA_Davy-TSE$ ./script.sh
deployment.apps/redis-main created
service/redis-main created
deployment.apps/redis-replica created
service/redis-replica created
deployment.apps/node-server created
service/node-server-loadbalancer created
deployment.apps/frontend created
deployment.apps/prometheus created
service/prometheus created
configmap/prometheus-config created
configmap/grafana-datasources created
configmap/grafana-dashboard-provider created
configmap/grafana-config created
deployment.apps/grafana created
service/grafana created
configmap/grafana-dashboards created
horizontalpodautoscaler.autoscaling/redis-replica-autoscaler created
horizontalpodautoscaler.autoscaling/node-server-autoscaler created
Les pods sont en cours de lancement. Veuillez patientez :)
pod/redis-main-59fccf9c4-ld577 condition met
pod/redis-replica-5b76d5896c-rjvkb condition met
pod/node-server-6584647868-246bj condition met
pod/frontend-655ffdcdbd-5mj9j condition met
pod/prometheus-647678c7bd-k68jx condition met
pod/grafana-5bbf974dc-t9t5x condition met
```

Ici le script lance l'ensemble des pods et services. Cette étape peut prendre quelques minutes.

Davy TSE - 3672079
Ounas YHIA - 21312539

A cette étape un tunnel minikube est créé. Le script ouvrira un terminal où il peut vous être demandé d'entrer votre mot de passe. Ensuite, la page web permettant d'accéder au tableau de bord grafana sera ouvert automatiquement :

```
Lancement du tunnel minikube... Veuillez entrer votre mot de passe si necessaire dans le terminal qui vient de s'ouvrir !
Adresse IP externe du service de load balancing des serveurs en attente de provisionnement.
Adresse IP externe du service de load balancing des serveurs en attente de provisionnement.
Adresse IP externe du service de load balancing des serveurs en attente de provisionnement.
Adresse IP externe du service de load balancing des serveurs en attente de provisionnement.
Adresse IP externe du service de load balancing des serveurs bien recuperee
```

Lancement du tableau de bord grafana relié à grafana :

NAMESPACE	NAME	TARGET PORT	URL
default	grafana	http/3000	http://192.168.49.2:30183

🔥 Ouverture du service default/grafana dans le navigateur par défaut...

E0406 15:29:37.406612 13702 logFile.go:53] failed to close the audit log: invalid argument

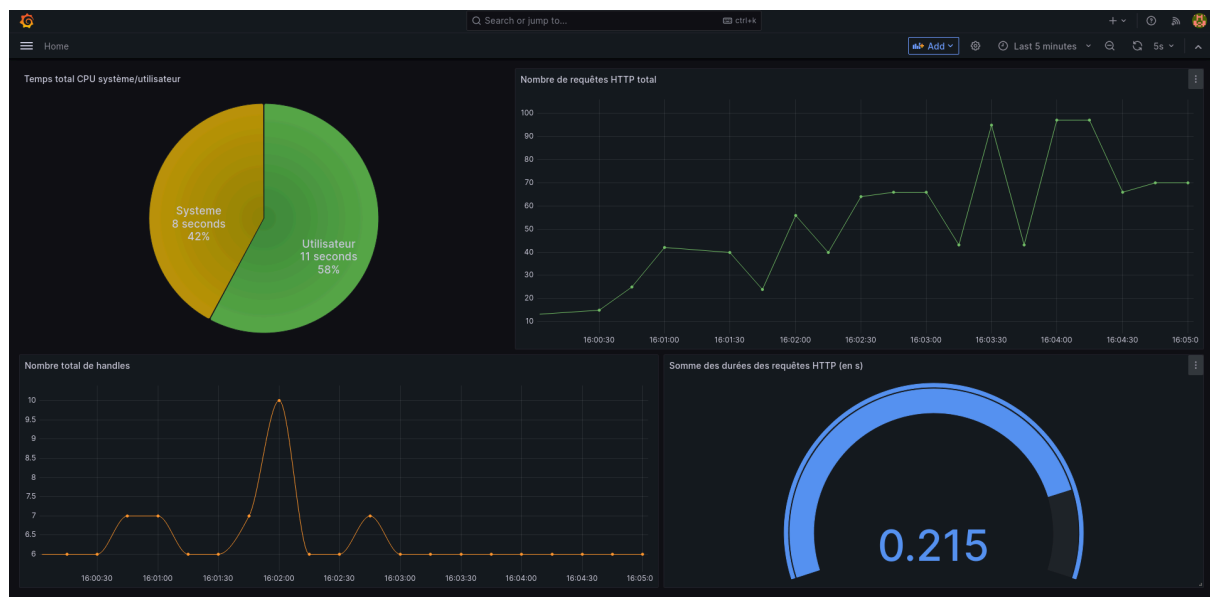


Tableau de bord grafana

Enfin, à cette étape, le script créé un tunnel http tunnelmole permettant d'accéder au site web. **ATTENTION**, il faut choisir le premier lien (le lien HTTP et non pas le lien HTTPS) :

Ci-dessous les url permettant d'accéder au frontend du site. Afin que tout fonctionne sans encombre, veuillez cliquez sur le premier lien http et pas celui https :

```
http://l2ejfp-ip-45-66-108-81.tunnelmole.net is forwarding to localhost:3000
https://l2ejfp-ip-45-66-108-81.tunnelmole.net is forwarding to localhost:3000
```

Hey there 🐼. I'm Robbie, the creator of Tunnelmole. I'm looking to improve Tunnelmole and your feedback could make a huge difference!

Would you be interested in participating in a short interview over video call? As thanks, I'm offering a free 3-month Tunnelmole subscription. Send me an email at robbie-cahill@proton.me and we can set something up.

Thanks for supporting the community!

NB1 → Le tableau de bord Grafana est accessible en se connectant avec les identifiants suivants : username = admin, password = admin. Il vous sera proposé de changer votre mot de passe par la suite.

NB2 → A son lancement, le tableau de bord Grafana propose plusieurs graphes mais les données /metrics étant encore vide, il faut réaliser quelques requêtes sur le site pour que ceci s'affichent.

Davy TSE - 3672079

Ounas YHIA - 21312539

2. Si le script ne fonctionne pas pour une raison quelconque vous pouvez toujours refaire l'ensemble des commandes manuellement :
 - 2.1. `kubectl apply -f redis.yaml`
 - 2.2. `kubectl apply -f nodeJS.yaml`
 - 2.3. `kubectl apply -f frontend.yaml`
 - 2.4. `kubectl apply -f prometheus.yaml`
 - 2.5. `kubectl apply -f grafana.yaml`
 - 2.6. `kubectl apply -f autoScaling.yaml`
 - 2.7. `minikube addons enable metrics-server`
 - 2.8. `minikube tunnel`
 - 2.9. `kubectl get svc` (afin de récupérer l'adresse IP du service `node-server-loadbalancer`)
 - 2.10. Rentrer dans le pod du frontend puis dans le fichier `src/conf.js` écrire l'URL suivante : http://adresseIP_serveurJS:3000
 - 2.11. Toujours à l'intérieur du pod frontend, entrer la commande suivante pour créer le tunnel http : `tmole 3000`
 - 2.12. Dans un autre terminal, pour lancer grafana : `minikube service grafana`