CRV

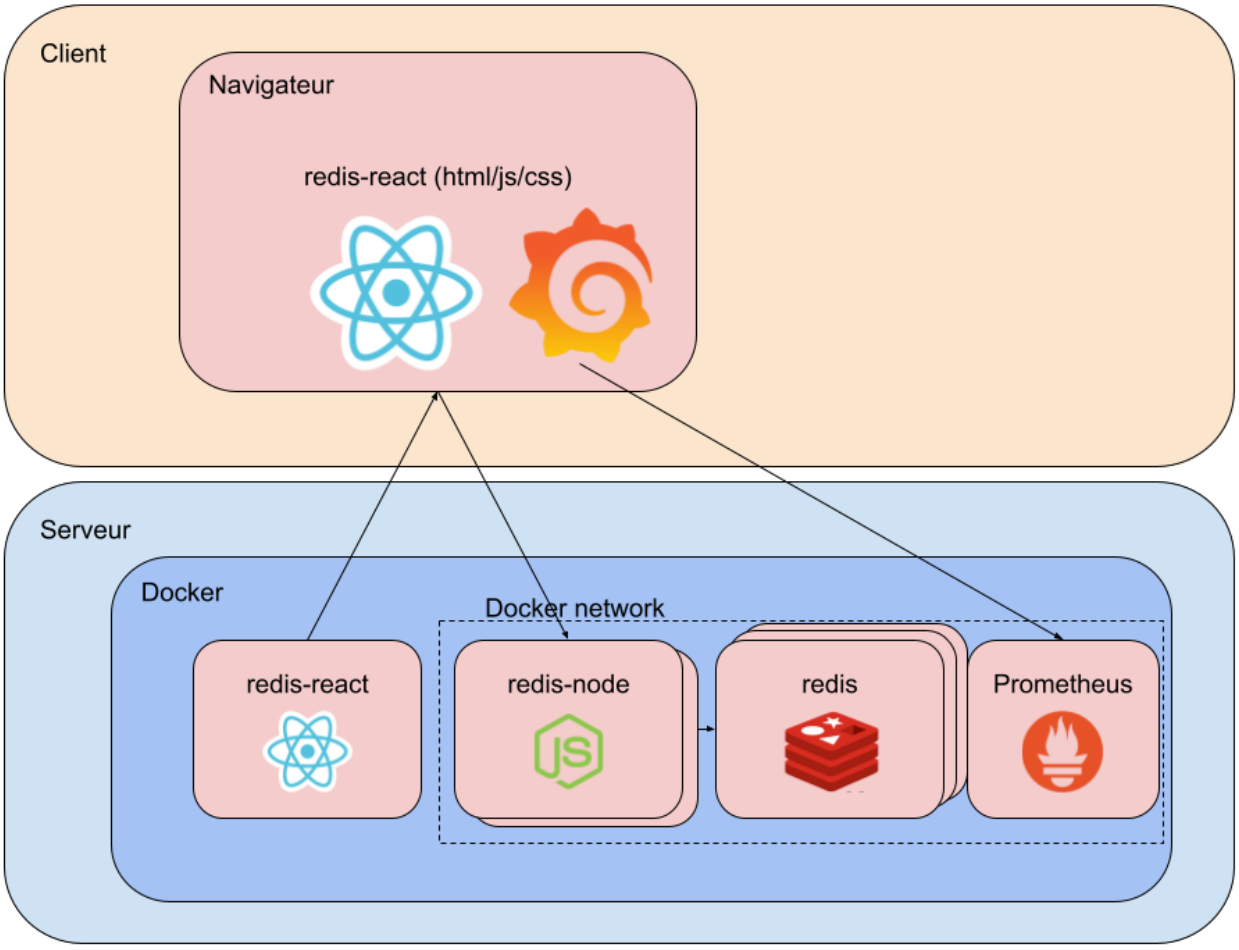
Projet AutoScaling et IaC

L'objectif de ce projet est de créer une infrastructure pour le projet redis-nodejs vu en TME dans un cluster kubernetes.

Cette infrastructure devra pouvoir monter à l'échelle automatiquement pour chacun de ses composants.

De plus pour observer le comportement des différents composants il faudra configurer un outil de monitoring: prometheus/grafana.

L'infrastructure



Redis

Une base de données redis implémentant un pattern main/replicas, une base redis principale

accepte toutes les opérations et des replicas recopient les données de cette base pour accepter plus de lecture en parallèle.

Lorsqu'une écriture survient sur la base principale propage la modifications aux réplicas. Les réplicas n'acceptent pas d'écriture.

Dans un premier temps seul les replicas pourront monter à l'échelle. Mais il est possible d'implémenter d'autres stratégies plus élaborées si le travail initial est terminé.

https://hub.docker.com/_/redis

Nodejs

Un serveur nodejs stateless (vu en TME) qui appel la base redis et ses replicas.

Grâce à la caractéristique stateless de ce serveur (ne possède pas d'état persistant) il est possible de le dupliquer sans modifier le comportement de l'application.

https://github.com/arthurescriou/redis-node

React

Un projet front end fait avec le framework React qui appel le serveur nodejs. Qui doit être build pour être déployé par un serveur de fichier statique.

La raison demandant une montée à l'échelle du client frontend est l'ouverture de beaucoup de clients simultanément. Ce n'est pas un cas que nous souhaitons explorer dans ce projet, le frontend ne sera utilisé seulement pour vérifier que le reste de l'application fonctionne corréctement.

https://github.com/arthurescriou/redis-react

Prometheus/grafana

Un outil de monitoring capable de se brancher sur plusieurs sources de données.

Il est indispensable de connecter prometheus au serveur nodejs via son api `/metrics`.

Cependant il est également possible d'ajouter des sources de données provenant de kubernetes ou redis grâce à des connecteurs supplémentaire.

Il peut être nécessaire de créer des prometheus exporter pour pouvoir récupérer des données d'autres sources. Pour ça il faut trouver ou construire des images docker adéquates.

Il est inutile de réfléchir à comment faire monter cette partie à l'échelle de l'application dans le cadre de ce projet.

https://hub.docker.com/r/prom/prometheus

Prometheus vient avec le tableau de bord Grafana pour afficher les données :

https://hub.docker.com/r/grafana/grafana

Objectifs et rendu

L'objectif de ce projet est de créer l'infrastructure décrite ci-dessus, la déployer sur le cluster kubernetes fourni dans le cadre du cours et de documenter comment la reproduire.

Pour ça il faudra fournir un depôt git contenant les divers fichiers necessaire à sa création :

- configuration yaml de kubernetes et prometheus
- Dockerfile créés
- script bash d'automatisation
- un readme.md décrivant la démarche à suivre
- etc

Des images docker peuvent être aussi hebergé sur un depôt docker distant et fourni via la documentation.

De plus un rapport au format pdf décrivant plus en détails votre démarche et vos décisions lors de cette conception devra être ajouté au rendu.

Un soin particulier devra être apporté tout au long du projet et du rapport sur les notions de montée à l'échelle automatique et dynamique et sur la reproductibilitée du déploiement de l'infrastructure demandée.