

Rapport de Projet de Programmation Distribuée – Microservices

Binôme : ALLAIN Yanis - TADLY Samy

Professeur : CHARROUX Benoit

Sujet : Application de gestion de tournois avec Flask, PostgreSQL et Kubernetes

Préambule

Ce projet n'a pas pu aboutir à cause de facteurs imprévus et indépendants de notre volonté. En effet, un membre de l'équipe a été hospitalisé. Ainsi vous ne trouverez pas sont Google lab dans ce rapport car il n'a pas pu le fournir. De plus, le second membre se retrouvant seul n'a pas pu finir correctement le projet. Plusieurs mails ont été envoyés pour avertir de la situation mais se sont retrouvés sans réponse. Donc nous demandons ici encore une fois un délai supplémentaire pour pouvoir finaliser le projet dans de bonnes conditions. Si vous acceptez, veuillez ne pas tenir compte de ce qui suit dans ce rapport. Merci de votre attention.

Introduction

Ce projet vise à illustrer l'utilisation de Kubernetes pour le déploiement et la gestion d'une application web composée de plusieurs services distincts. On met ici en avant les bénéfices des conteneurs dans le cadre du développement d'une application. En adoptant une architecture microservices, le projet favorise une meilleure modularité ainsi qu'une capacité d'adaptation et de montée en charge plus efficace. L'objectif principal est de proposer une plateforme robuste et souple pour le déploiement d'applications conteneurisées dans un environnement Kubernetes.

L'application

Il s'agit d'un site web développé en Python avec le backend ainsi que le frontend qui utilisent Flask. Nous avons utilisé PostgreSQL pour la base de données.

Pour la structure, le backend est la partie qui va gérer les données et faire les requêtes à la base. Le frontend fait alors des requêtes http au backend qui lui transmet les données voulues pour qu'il puisse les afficher avec des templates html.

Vous trouverez une vidéo du fonctionnement du site dans le dépôt GitHub.

Les images Docker

A partir de notre code nous avons créé des images Docker à partir de dockerfile vous pouvez les retrouver sur Docker Hub.

```
[yanis@cachyos-x8664 yml]$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
kemoury/mkbackend	latest	07ad9e0ead9e	2 hours ago	592MB
kemoury/mkfrontend	latest	cdbca9b07558	2 hours ago	207MB

Kubernetes et déploiement

Nous avons ensuite déployé l'application sur Kubernetes en définissant des déploiements pour chaque service à l'aide de fichiers YAML dédiés. Ces fichiers précisent les caractéristiques propres à chaque service, telles que l'image Docker à utiliser, le nombre de répliques souhaité, ainsi que les ressources nécessaires. Nous avons également mis en place une passerelle Ingress.

```
[yanis@cachyos-x8664 yml]$ kubectl get all -n myproject
```

NAME	READY	STATUS	RESTARTS	AGE
pod/backend-6ff8d8575-46k8b	1/1	Running	0	47m
pod/backend-6ff8d8575-xgzbt	1/1	Running	0	47m
pod/frontend-5fd8bb796d-77v7b	1/1	Running	0	55m
pod/frontend-5fd8bb796d-wlm5s	1/1	Running	0	55m
pod/postgres-fd6d648df-qdddj	1/1	Running	0	61m

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/backend-service	ClusterIP	10.99.198.191	<none>	80/TCP	61m
service/frontend-service	ClusterIP	10.111.7.50	<none>	80/TCP	61m
service/postgres-service	ClusterIP	10.99.250.20	<none>	5432/TCP	61m

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/backend	2/2	2	2	61m
deployment.apps/frontend	2/2	2	2	61m
deployment.apps/postgres	1/1	1	1	61m

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/backend-5b978f6994	0	0	0	61m
replicaset.apps/backend-6ff8d8575	2	2	2	47m
replicaset.apps/frontend-5fd8bb796d	2	2	2	55m
replicaset.apps/frontend-75cc9f7f98	0	0	0	61m
replicaset.apps/postgres-fd6d648df	1	1	1	61m

```
[yanis@cachyos-x8664 yml]$
```

```
[yanis@cachyos-x8664 yml]$ kubectl get ingress -n myproject
```

NAME	CLASS	HOSTS	ADDRESS	PORTS	AGE
project-ingress	<none>	myproject.local	192.168.49.2	80	66m

Conclusion

Avec ce projet nous avons pu voir comment gérer une application avec plusieurs services. Ainsi la mise en place d'une architecture utilisant des microservices offre une bonne structuration et flexibilité pour les applications. Docker a permis de faciliter le déploiement des différents services mais aussi Kubernetes permet de faciliter leur gestion.

Liens

- Image Docker [frontend](#)
- Image Docker [backend](#)
- Dépôt [Github](#)

Google Labs

Google Cloud Tableau de bord Explorer Parcours Abonnements

Google Cloud Skills Boost

[← Back to Dashboard](#)

Progress

Track your learning activities, pick up where you left off, and celebrate what you've completed

Cours Atelier Quiz Jeu Learning path Classroom En cours Terminée

Activité	Type	Date de début	Date de fin	Score	Réussite
Infrastructure as Code avec Terraform	Atelier	il y a 10 jours	il y a 10 jours	Assessment: 100%	✓
Bases de la gestion de réseaux	Atelier	il y a 10 jours	il y a 10 jours	Assessment: 100%	✓
Présentation des ateliers pratiques Google Cloud	Atelier	il y a 10 jours	il y a 10 jours	Assessment: 100%	✓

yanis a
Date d'abonnement : 2025
3 crédits
[Buy credits or subscription](#)

[Dashboard](#)
[Activités](#)
[Paramètres](#)
[Se déconnecter](#)

[Vie privée](#) [Conditions d'utilisation](#)

www.cloudskillsboost.google utilise des cookies Google afin de fournir ses services, d'en améliorer la qualité et d'analyser le trafic. [En savoir plus](#)

[J'ai compris](#)