

Rapport de projet - Kubernetes-Multi-Service-App

Nassim Lattab, Mohamed Azzaoui

Professeur : Benoit Charroux

Abstract—Cet abstract présente le déploiement d'une application multi-services Kubernetes. Grâce à une architecture basée sur des microservices et l'implémentation des autorisations RBAC, l'application bénéficie d'une sécurité et d'une flexibilité renforcée. L'intégration de Docker simplifie le déploiement, tandis que l'ajout d'une gateway locale Ingress permet une accessibilité facile de l'application depuis l'extérieur du cluster Kubernetes."

keywords—Kubernetes, Docker, DockerHub, Conteneurisation, Déploiement d'applications, Gestion de conteneurs, DevOps, Microservices, Base de données, Ingress Gateway, RBAC, Python, Web Application

1. Introduction

Le projet "Kubernetes-Multi-Service-App" explore l'utilisation de Kubernetes pour le déploiement et la gestion d'une application web composée de plusieurs services. Cette application est conçue pour démontrer les avantages de l'orchestration de conteneurs avec Kubernetes dans le contexte du développement d'applications modernes. Nous avons adopté une approche basée sur des microservices, qui permet une meilleure modularité et une évolutivité accrue de l'application. L'objectif principal est de fournir une plateforme flexible et fiable pour le déploiement d'applications conteneurisées dans des environnements Kubernetes.

2. Développement de l'application

Nous avons commencé par coder une mini application en Python, avec un backend utilisant Flask et un frontend également en Flask pour l'interface utilisateur. Le backend est responsable de la gestion des données, tandis que le frontend se charge de l'affichage des données. Les données manipulées sont simples et comprennent des informations telles que le nom, le prénom et le numéro d'étudiant, constituant une table de gestion standard pour les étudiants (nom, prénom, ID). Nous avons ajouté une base de données MySQL en local pour le backend, permettant ainsi la gestion dynamique des données et améliorant la flexibilité de l'application. Lorsqu'une requête est émise par le frontend, le backend transmet les données correspondantes et le frontend se charge de les afficher avec un simple HTML. Vous trouverez ci-dessous des captures d'écran de la base de donnée ainsi que l'application en cours d'exécution.

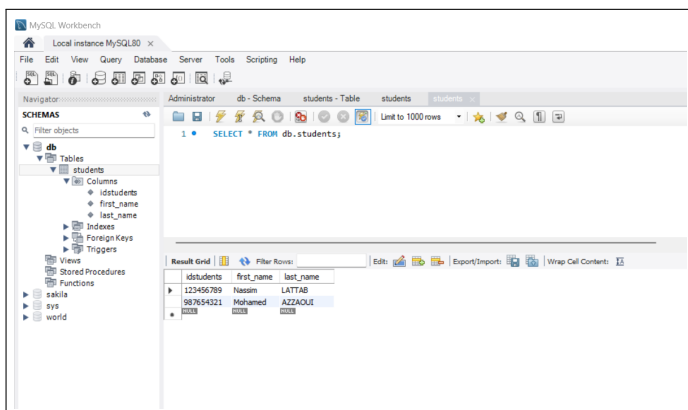


Figure 1. Base de donnée MySQL

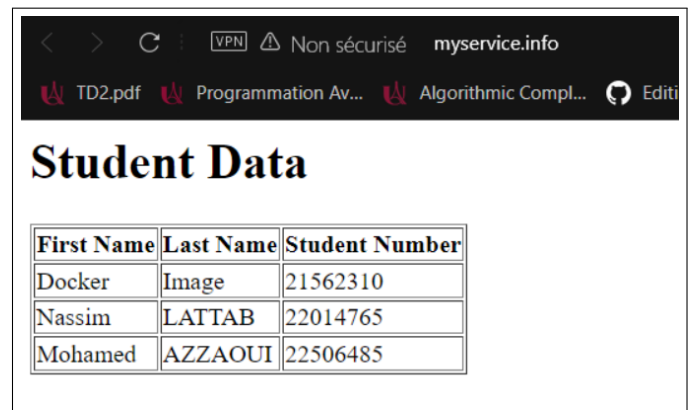


Figure 2. Application en cours d'exécution

3. Création des images Docker

Une fois les services développés, nous avons créé des images Docker pour chacun d'eux en utilisant un Dockerfile. Ces images ont ensuite été publiées sur Docker Hub. Vous pouvez trouver les liens vers ces images ainsi que le lien vers notre répertoire GitHub dans la section [Liens utiles](#).

```

PS C:\Users\Nassim\Desktop\M1\Dev_OPS\Kubernetes-Multi-Service-App> docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
nestallum/back-end   1                   3cbc31028b53       23 hours ago       71.2MB
nestallum/front-end  1                   6337849e435c       27 hours ago       255MB
myservice            latest             e22e626abb73       30 hours ago       251MB
nestallum/myservice  1                   e22e626abb73       30 hours ago       251MB
gcr.io/k8s-minikube/kicbase v0.0.42            dbc648475405       6 months ago       1.2GB
PS C:\Users\Nassim\Desktop\M1\Dev_OPS\Kubernetes-Multi-Service-App>

```

Figure 3. Création des images Docker

4. Déploiement sur Kubernetes

Nous avons ensuite déployé l'application sur Kubernetes en créant des déploiements pour chaque service à l'aide de fichiers YAML spécifiques. Ces déploiements décrivent les caractéristiques et les exigences de chaque service, comme les images Docker à utiliser, le nombre de répliques, les ressources requises, etc. Pour gérer la communication entre les différents composants de l'application, nous avons configuré des services Kubernetes. Ces services permettent aux différents services de communiquer entre eux de manière fiable, indépendamment de leur emplacement dans le cluster. Pour permettre l'accès externe aux services déployés, nous avons ajouté une gateway locale avec Ingress. Cela nous a permis de faciliter l'accès aux services depuis l'extérieur du cluster Kubernetes.

```

PS C:\Users\Nassim\Desktop\M1\Dev_OPS\Kubernetes-Multi-Service-App> kubectl get all
NAME                                READY   STATUS    RESTARTS   AGE
pod/back-end-deployment-7c9dd97577-g48lz  1/1     Running   2 (2m5s ago)  23h
pod/front-end-deployment-6f4d646d5c-5cdj9  1/1     Running   1 (2m36s ago)  23h

NAME                                TYPE               CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
service/back-end-service            ClusterIP          10.97.89.213    <none>        80/TCP           23h
service/front-end-service           NodePort           10.96.98.244    <none>        80:32696/TCP     23h
service/kubernetes                  ClusterIP          10.96.0.1       <none>        443/TCP          23h

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/back-end-deployment  1/1     1             1           23h
deployment.apps/front-end-deployment  1/1     1             1           23h

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/back-end-deployment-7c9dd97577  1         1         1       23h
replicaset.apps/front-end-deployment-6f4d646d5c  1         1         1       23h
PS C:\Users\Nassim\Desktop\M1\Dev_OPS\Kubernetes-Multi-Service-App> kubectl get ingress
NAME          CLASS    HOSTS              ADDRESS          PORTS   AGE
example-ingress  nginx   myservice.info     192.168.49.2    80      24h
PS C:\Users\Nassim\Desktop\M1\Dev_OPS\Kubernetes-Multi-Service-App>

```

Figure 4. Ressources déployées dans l'environnement Kubernetes

5. RBAC Kubernetes

Nous avons mis en place les autorisations RBAC dans Kubernetes. Ce système permet une gestion précise des autorisations en définissant des rôles qui décrivent les actions permises sur les ressources, puis en les associant à des utilisateurs ou des comptes de service via des liaisons de rôles. Ainsi, seuls les utilisateurs autorisés peuvent effectuer des actions spécifiques sur les ressources Kubernetes.

5.1. Role et RoleBinding

Nous avons défini un rôle (Role) nommé "read-only" qui autorise la lecture (get, list, watch) des ressources de type pods dans le namespace par défaut. Ensuite, nous avons créé une liaison de rôle (RoleBinding) nommée "read-only-pods" pour attribuer ce rôle à l'utilisateur "nestallum", permettant ainsi de lire les pods dans ce namespace.

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Nassim\Desktop\M1\Dev_OPS\Kubernetes-Multi-Service-App> kubectl get roles
NAME          CREATED AT
pod-reader    2024-05-11T03:36:36Z
PS C:\Users\Nassim\Desktop\M1\Dev_OPS\Kubernetes-Multi-Service-App> kubectl get rolebindings
NAME          ROLE          AGE
read-pods     Role/pod-reader 16s
PS C:\Users\Nassim\Desktop\M1\Dev_OPS\Kubernetes-Multi-Service-App> kubectl auth can-i get pods --as nestallum
yes
PS C:\Users\Nassim\Desktop\M1\Dev_OPS\Kubernetes-Multi-Service-App> kubectl auth can-i create pods --as nestallum
no
PS C:\Users\Nassim\Desktop\M1\Dev_OPS\Kubernetes-Multi-Service-App> 

```

Figure 5. RBAC : Role et RoleBinding

5.2. ClusterRole et ClusterRoleBinding

Pour une portée plus large, nous avons défini un rôle de cluster (ClusterRole) appelé "secret-reader", permettant cette fois-ci la lecture (get, list, watch) des ressources de type secrets dans tout le cluster. Par la suite, nous avons créé une liaison de rôle de cluster (ClusterRoleBinding) pour attribuer ce rôle à l'utilisateur "nestallum".

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Nassim\Desktop\M1\Dev_OPS\Kubernetes-Multi-Service-App> kubectl get clusterroles secret-reader
NAME          CREATED AT
secret-reader  2024-05-11T02:14:06Z
PS C:\Users\Nassim\Desktop\M1\Dev_OPS\Kubernetes-Multi-Service-App> kubectl get clusterrolebindings secret-reader-global
NAME          ROLE          AGE
secret-reader-global ClusterRole/secret-reader 56m
PS C:\Users\Nassim\Desktop\M1\Dev_OPS\Kubernetes-Multi-Service-App> kubectl auth can-i get secrets --as nestallum
yes
PS C:\Users\Nassim\Desktop\M1\Dev_OPS\Kubernetes-Multi-Service-App> kubectl auth can-i create secrets --as nestallum
no
PS C:\Users\Nassim\Desktop\M1\Dev_OPS\Kubernetes-Multi-Service-App> 

```

Figure 6. RBAC : ClusterRole et ClusterRoleBinding

6. Sécurisation du Registry Docker Hub

Nous avons mis en place des mécanismes d'authentification avancés sur Docker Hub, notamment les access tokens et l'authentification à deux facteurs (2FA), afin de renforcer la sécurité des comptes utilisateur et des registres d'images Docker.

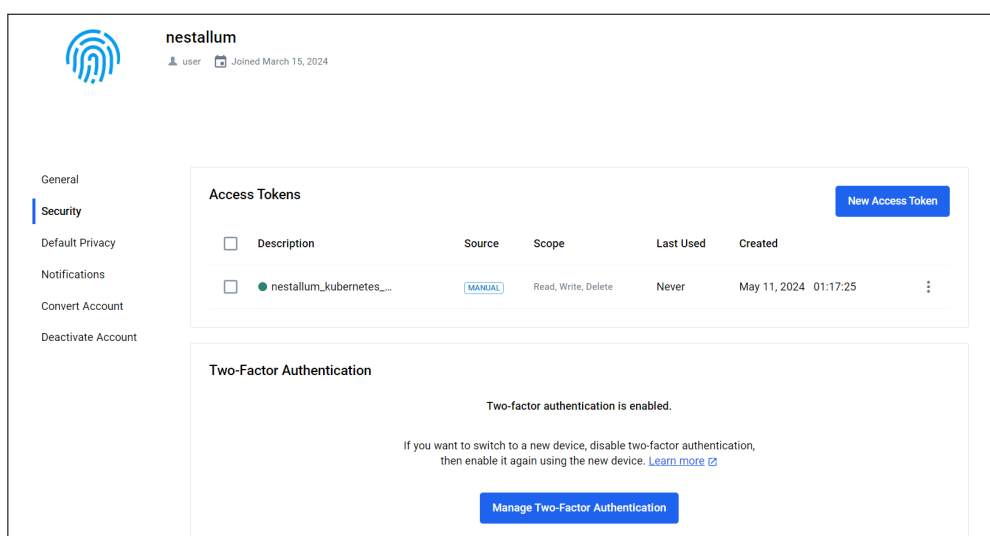


Figure 7. Sécurisation via Access Tokens et 2FA

7. Conclusion

En conclusion, ce projet "Kubernetes-Multi-Service-App" illustre l'utilisation de Kubernetes pour déployer et gérer plusieurs services. L'architecture basée sur des microservices a permis une meilleure organisation et une meilleure flexibilité de l'application. En plus de cela, l'ajout des autorisations RBAC (Contrôle d'Accès Basé sur les Rôles) renforce la sécurité en permettant une gestion fine des autorisations pour les utilisateurs et les comptes de service. Grâce à Docker, le déploiement et la gestion des services ont été facilités. L'ajout d'une gateway locale Ingress a rendu l'application accessible depuis l'extérieur du cluster Kubernetes, offrant ainsi une solution complète pour le déploiement et la gestion d'applications conteneurisées.

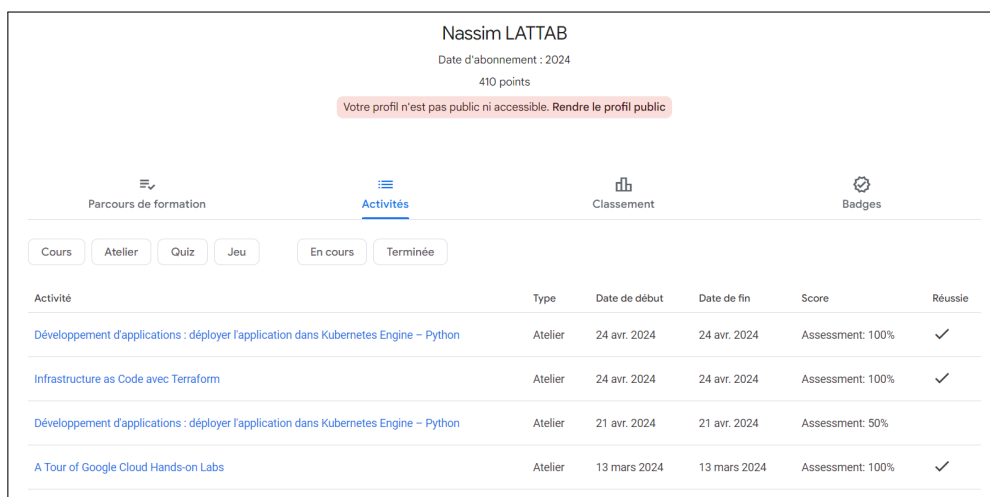
8. Liens utiles

- Repository GitHub : <https://github.com/Nestallum/Kubernetes-Multi-Service-App/tree/main>
- Docker Hub Images :
 - Backend : <https://hub.docker.com/r/nestallum/back-end>
 - Frontend : <https://hub.docker.com/r/nestallum/front-end>

9. Google Labs

Vous trouverez ci-dessous des captures d'écran des activités Google Labs pour nos deux profils.

9.1. Google Labs - Nassim Lattab



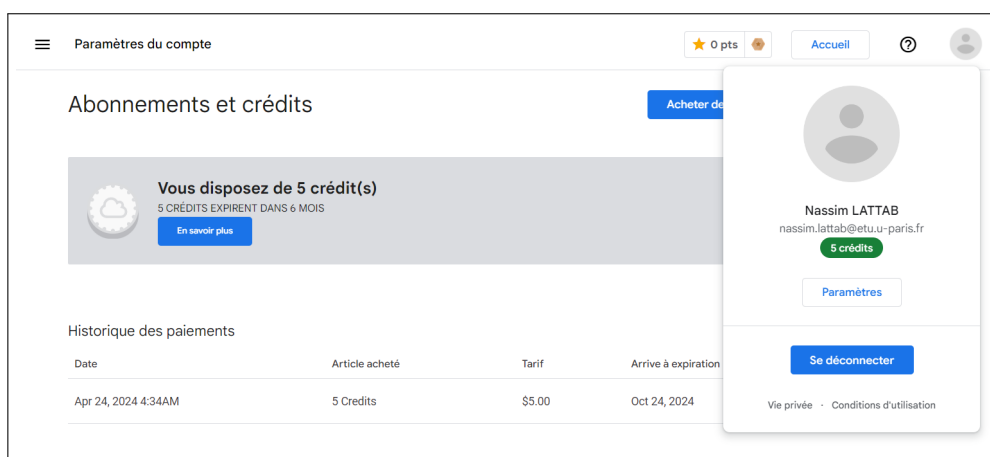
Nassim LATTAB
Date d'abonnement : 2024
410 points
Votre profil n'est pas public ni accessible. [Rendre le profil public](#)

Parcours de formation **Activités** Classement Badges

Cours Atelier Quiz Jeu En cours Terminée

Activité	Type	Date de début	Date de fin	Score	Réussite
Développement d'applications : déployer l'application dans Kubernetes Engine – Python	Atelier	24 avr. 2024	24 avr. 2024	Assessment: 100%	✓
Infrastructure as Code avec Terraform	Atelier	24 avr. 2024	24 avr. 2024	Assessment: 100%	✓
Développement d'applications : déployer l'application dans Kubernetes Engine – Python	Atelier	21 avr. 2024	21 avr. 2024	Assessment: 50%	
A Tour of Google Cloud Hands-on Labs	Atelier	13 mars 2024	13 mars 2024	Assessment: 100%	✓

Figure 8. Activités de profil - Nassim Lattab



Paramètres du compte 0 pts Accueil ?


Abonnements et crédits

Vous disposez de 5 crédit(s)
5 CRÉDITS EXPIRENT DANS 6 MOIS
[En savoir plus](#)

[Acheter des crédits](#)

Historique des paiements

Date	Article acheté	Tarif	Arrive à expiration
Apr 24, 2024 4:34AM	5 Credits	\$5.00	Oct 24, 2024



Nassim LATTAB
nassim.lattab@etu.u-paris.fr
5 crédits

[Paramètres](#)

[Se déconnecter](#)

Vie privée · Conditions d'utilisation

Figure 9. Acquisition de tokens supplémentaires pour effectuer à nouveau le lab n°2.

9.2. Google Labs - Mohamed Azzaoui

Mohamed Azzaoui						
Date d'abonnement : 2024						
290 points						
Votre profil n'est pas public ni accessible. Rendre le profil public						
<div><div>☰</div><div>Parcours de formation</div></div> <div><div>☰</div><div>Activités</div></div> <div><div>☰</div><div>Classement</div></div> <div><div>☑</div><div>Badges</div></div>						
<div><div>Cours</div><div>Atelier</div><div>Quiz</div><div>Jeu</div><div>En cours</div><div>Terminée</div></div>						
Activité	Type	Date de début	Date de fin	Score	Réussie	
Développement d'applications : déployer l'application dans Kubernetes Engine – Python	Atelier	il y a 39 minutes	il y a 0 minute	Assessment: 100%	✓	
Infrastructure as Code avec Terraform	Atelier	il y a 8 heures	il y a 8 heures	Assessment: 100%	✓	
A Tour of Google Cloud Hands-on Labs	Atelier	13 mars 2024	13 mars 2024	Assessment: 100%	✓	

Figure 10. Activités de profil - Mohamed Azzaoui