

Préparation projet

Kubernetes :

- Le phase Dev d'un premier service :
 - programmer un service web qui affiche votre nom dans le langage que vous voulez :
<https://github.com/charroux/kubernetes-minikube/tree/main/MyService>
 - La version de fichier source est indiquée dans :
<https://github.com/charroux/kubernetes-minikube/blob/main/MyService/build.gradle>
 - builder ce programme :
<https://github.com/charroux/kubernetes-minikube/tree/main?tab=readme-ov-file#test-this-project-using-docker>
 - La version du compilateur dépend de votre machine (faire java -version)
- La phase Ops de déploiement d'un premier service :
 - créer une image Docker :
 - un Dockerfile :
<https://github.com/charroux/kubernetes-minikube/blob/main/MyService/Dockerfile>
 - Attention à la version de java
 - <https://github.com/charroux/kubernetes-minikube/tree/main?tab=readme-ov-file#test-this-project-using-docker>
 -
 - publier l'image Docker dans le Docker Hub :
<https://github.com/charroux/kubernetes-minikube/tree/main?tab=readme-ov-file#publish-the-image-to-the-docker-hub>
 - déployer l'image Docker dans un cluster Kubernetes :
 - un fichier yaml de déploiement :
<https://github.com/charroux/kubernetes-minikube/blob/main/myservice-deployment.yml>
 - <https://github.com/charroux/kubernetes-minikube/tree/main?tab=readme-ov-file#create-a-deployment-and-a-service-using-a-yaml-file>
 - ajouter un service Kubernetes devant le déploiement :
 - un fichier yaml :
<https://github.com/charroux/kubernetes-minikube/blob/main/myservice-loadbalancing-service.yml>
 - tester que ça fonctionne dans un navigateur (faire une copie d'écran du fonctionnement) :
<https://github.com/charroux/kubernetes-minikube/tree/main?tab=readme-ov-file#expose-http-and-https-route-using-nodeport>
 - envoyer tout ça dans un dépôt Git (y compris la coupe d'écran sous forme d'un lien vers une image dans le ReadMe)

Une gateway avec le service mesh :

- Installer Istio : <https://github.com/charroux/servicemesh?tab=readme-ov-file#install-istio>
- Configurer la gateway. Voir gateway dans :
<https://github.com/charroux/servicemesh/blob/main/infrastructure.yaml>
- Définir les routes des services dans la gateway. Voir virtual service dans :
<https://github.com/charroux/servicemesh/blob/main/microservices.yaml>

- Récupérer l'adresse de la gateway et afficher dans un navigateur la réponse du service :
<https://github.com/charroux/servicemesh?tab=readme-ov-file#microservices-service-mesh-proxies-and-routing-via-the-gateway>
- Afficher des dashboards du cluster :
<https://github.com/charroux/servicemesh?tab=readme-ov-file#monitoring>

Projet

Sujet libre, mais technologies et patterns imposés :

- Web service (Java, Node...)
- Docker
- Multi conteneurs :
 - o Kubernetes
 - o Base de données (SQL, NoSQL, hors en dans cloud)
- Options :
 - o Micro-services (noté en bonus)
 - o Déploiement dans un cloud
 - o Front end (Angular, React, VueJS), Android..., frameworks

Travail en binôme.

Conseils pour mener à bien le projet

Commencer par un seul service en local (10/20)

- Coder une mini application dans le langage que vous voulez
- Créer une image Docker => faire un Dockerfile
- Publier l'image Docker sur le Docker Hub
- Créer un déploiement Kubernetes
- Créer un service Kubernetes

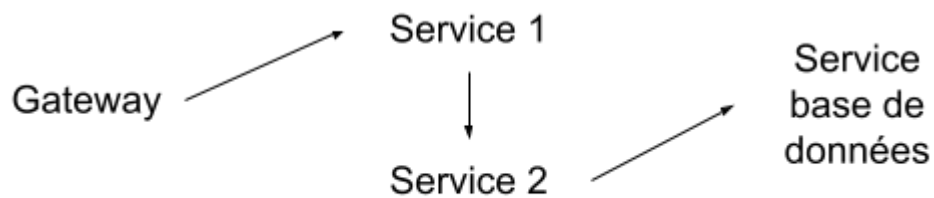
<https://github.com/charroux/kubernetes-minikube>

Ajouter une gateway en local (12/20)

<https://github.com/charroux/kubernetes-minikube#routing-rule-to-a-service-using-ingress>

ou via un service mesh

<https://github.com/charroux/servicemesh>



Service 2

Ajouter un deuxième service en local (14/20)

- Reprendre “Commencer par un seul service”
- Relier les services entre-eux

<https://github.com/charroux/CodingWithKubernetes>

Deux services + Service mesh

<https://github.com/charroux/servicemesh-kubernetes>

Ajouter une base de données en local ou dans un cloud (16/20)

- MySQL : <https://github.com/charroux/noops/tree/main/mysql>
- PostgreSQL : <https://github.com/charroux/noops/tree/main/postgres>

ET/ou accéder à un système de fichier : <https://github.com/charroux/kubernetes-volumes>

Sécuriser le cluster (16/20)

- Ajouter des RBAC Kubernetes : :
<https://kubernetes.io/fr/docs/reference/access-authn-authz/rbac/>
- Sécurité proxy (Service mesh)
 - Cryptage échange :
<https://istio.io/latest/docs/tasks/security/authentication/mtls-migration/>
- Sécurisation image Registry : <https://hub.docker.com/settings/security>
- Outils de contrôle de la sécurité Kubernetes :
- https
- ...

Déployer dans une infrastructure cloud (option) et/ou sécuriser votre cluster (18/20)

Conseil : sauvegarder vos applications dans github, gitlab.

Evaluation du projet

Pour les Master info

- Envoyer par mail à benoit.charroux@gmail.com
- un mini rapport pour que je comprenne ce que vous avez fait avec des copies d'écran de ce à quoi je dois m'attendre et des copies d'écran individuelles des Google labs (voir activité de votre profil)
- le code sur un projet personnel (1 par binôme) sur Github ou Gitlab
- Date butoir de remise du projet 19 mai 2024

Pour les Miage FA

- Envoyer par mail à benoit.charroux@gmail.com
- un mini rapport pour que je comprenne ce que vous avez fait avec des copies d'écran de ce à quoi je dois m'attendre et des copies d'écran individuelles des Google labs (voir activité de votre profil)
- le code sur Github ou Gitlab
- Date butoir de remise du projet 16 juin 2024

Pour les Miage FI

- Envoyer par mail à benoit.charroux@gmail.com
- un mini rapport pour que je comprenne ce que vous avez fait avec des copies d'écran de ce à quoi je dois m'attendre et des copies d'écran individuelles des Google labs (voir activité de votre profil)
- le code sur Github ou Gitlab
- Date butoir de remise du projet fin janvier 2024

Pour les AMSD

- Envoyer par mail à benoit.charroux@gmail.com
- un mini rapport pour que je comprenne ce que vous avez fait avec des copies d'écran de ce à quoi je dois m'attendre et des copies d'écran individuelles des Google labs (voir activité de votre profil)
- le code sur Github ou Gitlab
- Date butoir de remise du projet fin janvier 2024

Pour les Cyber

- Envoyer par mail à benoit.charroux@gmail.com
- un mini rapport pour que je comprenne ce que vous avez fait avec des copies d'écran de ce à quoi je dois m'attendre et des copies d'écran individuelles des Google labs (voir activité de votre profil)
- le code sur Github ou Gitlab
- Date butoir de remise du projet fin avril

Critères d'évaluations (par ordre décroissant d'importance) :

- Intégration complète d'un maximum de technologies (Web Services, Docker, Kubernetes...)
- Codage
- Fonctionnalités
- présentation (front office - css)