STACK WEB VIA KUBERNETES

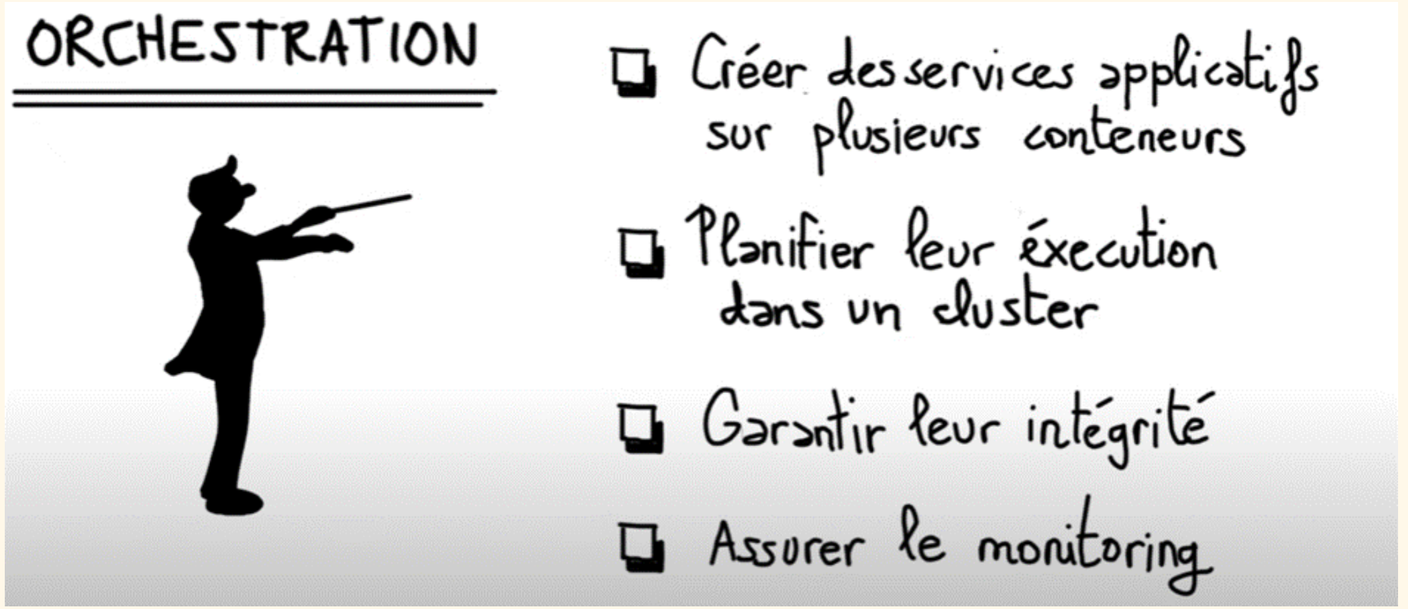
+

Antonio Gernac
Arsène Onambele

Février 2023

Brève description de kubernetes

La virtualisation par conteneur est une méthode de cloisonnement au niveau de l'OS et est basé sur la technologie de virtualisation LXC (Linux Containers). Le principe étant de faire tourner des environnements linux isolés les uns des autres dans des conteneurs tout en se partageant le même noyau Linux. Le conteneur virtualise uniquement l'environnement d'exécution comme la mémoire, le processeur ou le système de fichier...
Comme le conteneur n'a pas conscience de ce qui se passe en dehors du noyau, c'est là que Kurbenetes intervient. Il va apporter l'orchestration et la gestion des conteneurs sur des clusters de serveurs (Un cluster est un ensemble de nœuds reliés entre eux, travaillant ensemble pour exécuter des applications et offrir une haute disponibilité et une tolérance aux pannes.)



Quelques notions qu'apportent kubernetes par rapport à docker

- Nœuds : Qui peuvent être des serveurs physiques ou virtuelles

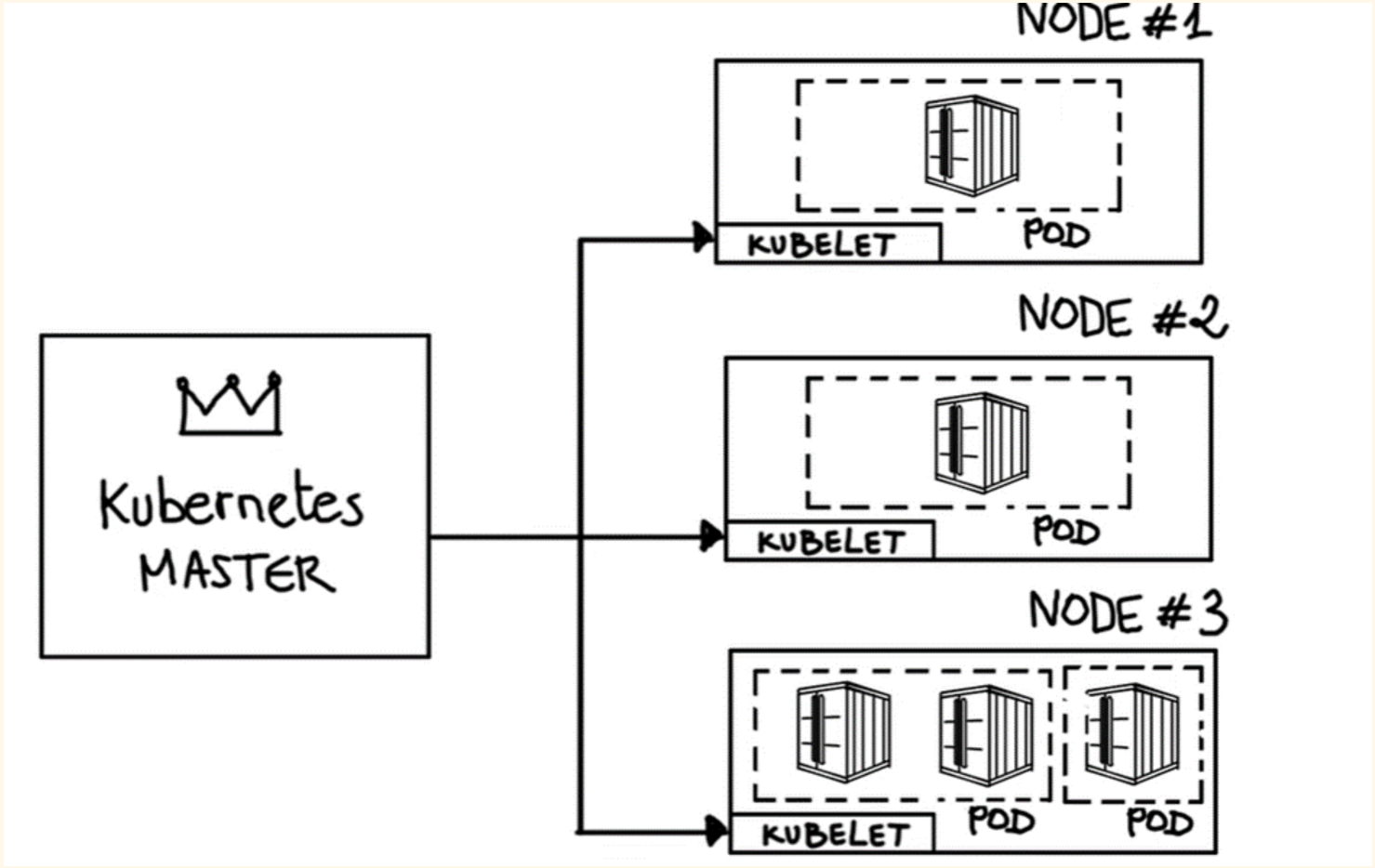
- Pods : Son but est de fournir un ensemble cohérent de conteneurs. En d'autres termes, c'est un environnement d'exécution d'un ou de plusieurs conteneurs

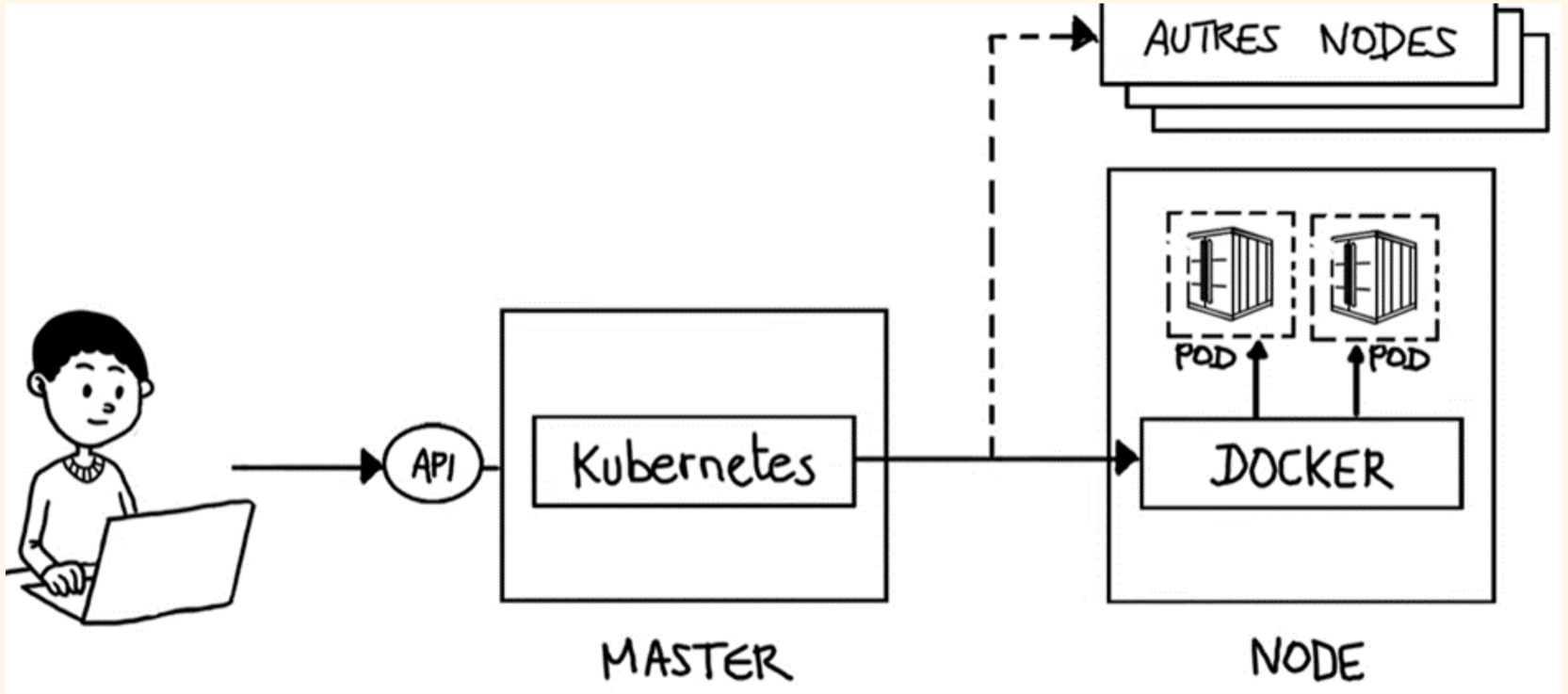
-Services : Un service rend les pods accessibles à d'autres pods ou aux utilisateurs externes au cluster. Sans service, le pod ne peut pas être accessible

-Volumes : Espace de stockage accessible par tous les conteneurs sur un pod. Il répond à deux besoins :
-Préserver les données au-delà du cycle de vie d'un conteneur.
-Nécessaire pour le partage des données entre deux conteneurs.

-Déploiement : Permet de gérer les créations et les suppressions de pods. C'est ici qu'on fournira le descriptif

Fonctionnement basique de kubernetes:





Kubernetes s'exécute au-dessus de l'OS et interagit avec les pods de conteneur qui s'exécutent sur des nodes. Le master kubernetes reçoit les commandes de la part d'un administrateur et va relayer ces instructions aux nodes. Le node le plus adapté pour la tâche va être choisit automatiquement. Il va ensuite allouer les ressources nécessaires au pod choisit dans le node pour effectuer la tâche requise.

Lorsque le master planifie la tâche dans un nœud, le kubelet ordonne à docker de lancer les conteneurs spécifiés et c'est docker qui va démarrer ou arrêter les conteneurs. Le kubelet collecte en continue le statut de ces conteneurs via docker et rassemble ces informations sur le Master.

Objectif du projet

Déployer plusieurs vhost

Déployer plusieurs versions de PHP

-Nous verrons comment modifier facilement la version de php

Déployer plusieurs application :php, python, nodes.js ....

-Nous déploierons nodejs et php

Problème survenue lors de l'installation de minikube/kubectl

Minikube est un outil facilitant l’exécution locale de Kubernetes. Minikube exécute un cluster Kubernetes à nœud unique dans une machine virtuelle (VM) de votre ordinateur portable pour les utilisateurs qui souhaitent essayer Kubernetes ou le développer au quotidien.

Sous ubuntu22.04 l'utilisation de "snap" (non disponibe avec apt) afin d'installer minikube et kubectl nous donner des erreurs , nous avons donc décider d'utiliser le gestionnaire de paquets homebrew

Afin de faciliter l'installation de celui-ci nous avons mis en place un script qui nous prépare un environnement pour kubernetes.

Fichier de configuration YAML

Afin de configurer nos "deployement","services" et "ingress" (différents composant de kubernetes) nous allons utiliser un fichier de configuration YAML (YAML Ain't Markup Language)

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: apache-server
spec:
  replicas: 1
  selector:
    matchLabels:
      app: apache-server
  template:
    metadata:
      labels:
        app: apache-server
    spec:
      containers:
      - name: apache-container
        image: httpd:2.4
        ports:
        - containerPort: 80
```

La configuration de service va permettre a kubernetes de faire communiquer differents pods

ConfigMap

Les ConfigMaps peuvent être créées et gérées directement dans Kubernetes et peuvent être utilisées dans différents contextes, tels que :

Dans les Pod : les ConfigMaps peuvent être utilisées directement dans les Pod en utilisant des variables d'environnement ou en montant un fichier ConfigMap en tant que volume.

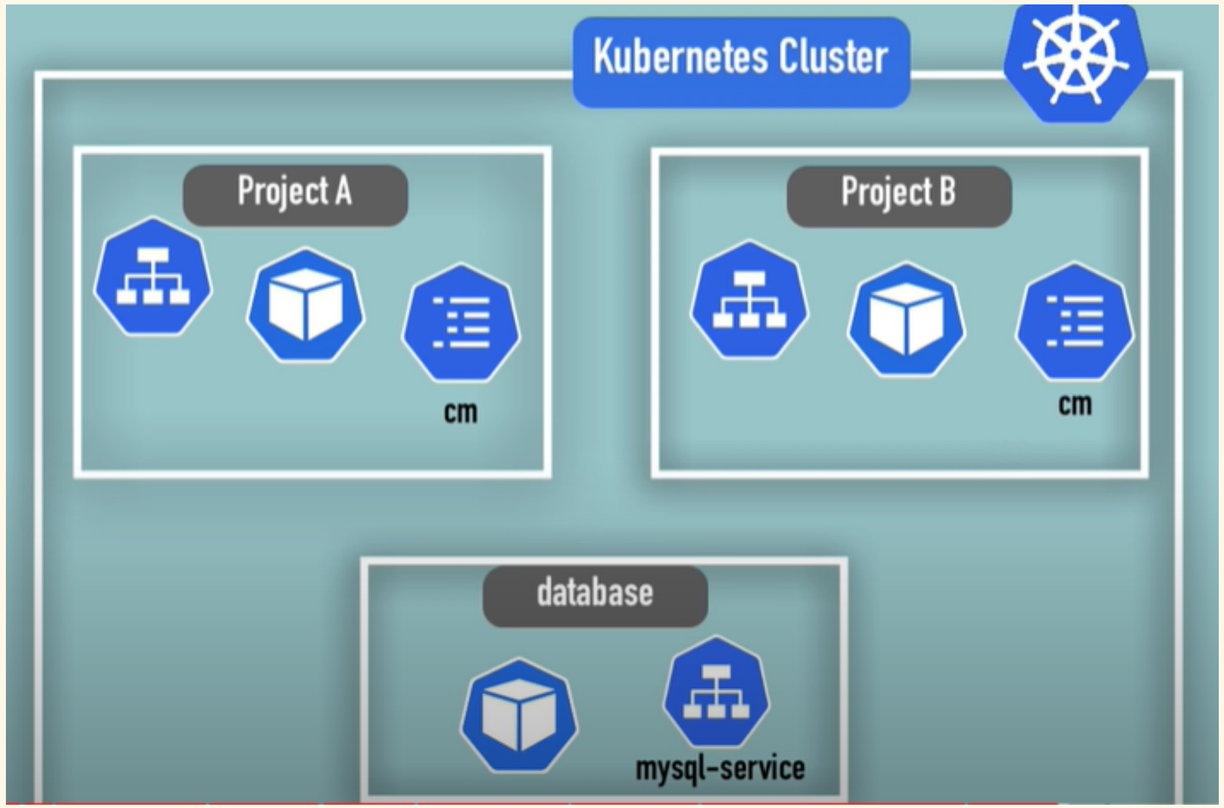
Dans les Deployments : les ConfigMaps peuvent être utilisées pour définir les paramètres d'application lors du déploiement de vos applications.

Dans les Services : les ConfigMaps peuvent être utilisées pour définir des informations de configuration pour les Services, telles que les adresses IP.

En utilisant les ConfigMaps, vous pouvez définir des données de configuration centralisées et partagées entre plusieurs Pod et Services dans votre cluster Kubernetes.

Les namespaces sont utiles pour :

Isoler les ressources de différentes équipes ou projets au sein d'un même cluster, sans avoir à créer plusieurs clusters séparés.
Gérer les autorisations et les contrôles d'accès pour différents utilisateurs ou groupes, sans avoir à affecter des autorisations pour chaque ressource individuelle.
Diviser les ressources en fonction de leur utilisation, par exemple, les ressources de production et de test peuvent être isolées dans des namespaces distincts.

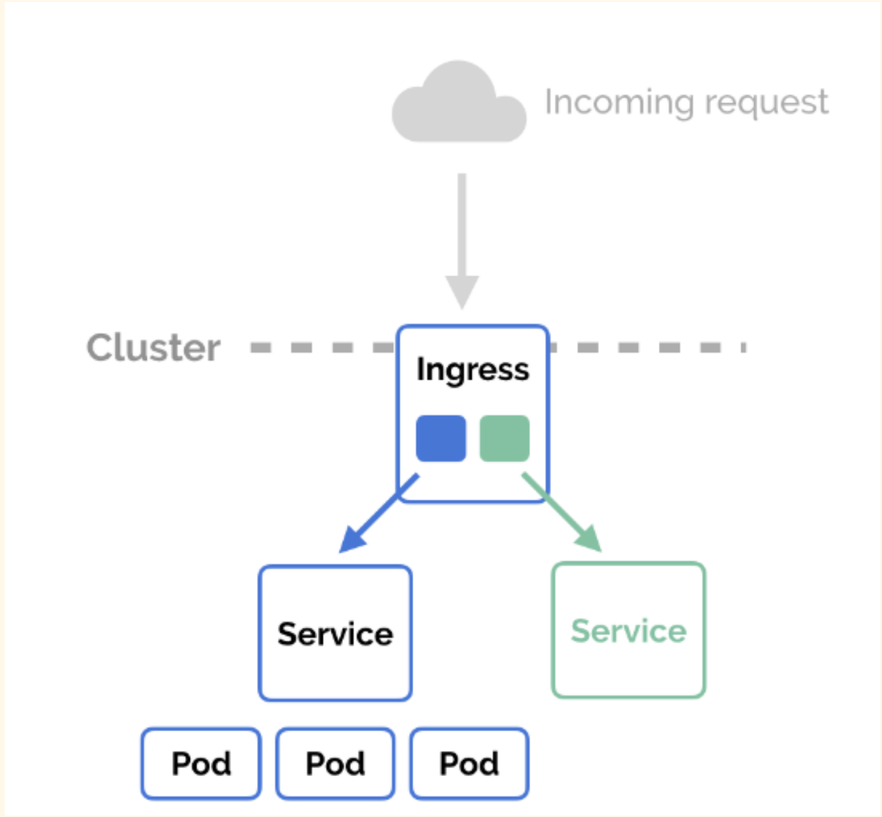


Grace au config Map nous pouvons avoir accès a la base depuis deux nœuds différents

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: mysql-configmap
data:
  db_url: mysql-service.database
```

Ingress

L'ingress est un mécanisme de Kubernetes qui permet de configurer l'accès au services à partir d'Internet. Il définit des règles pour acheminer les requêtes HTTP et HTTPS entrantes vers les services appropriés à l'intérieur de votre cluster. En d'autres termes, l'ingress agit comme une passerelle qui contrôle l'accès à vos applications. Il peut être utilisé pour fournir une adresse IP stable, une URL et une gestion de la qualité de service pour vos applications, ce qui facilite la mise en œuvre d'une architecture microservice.



Avantages de Kubernetes :

-Scalabilité : permet de gérer facilement l'échelle des applications pour répondre à la demande.
-Haute disponibilité : assure la disponibilité des applications en cas de défaillance d'un nœud ou d'un conteneur.
-Portabilité : peut être déployé sur divers environnements, tels que les nuages publics, privés ou hybrides.
-Automatisation : facilite la gestion de la configuration et du déploiement des applications grâce à des outils de déploiement automatisés.
-Interopérabilité : fonctionne avec de nombreuses technologies de conteneurs et de plateformes.

Inconvénients de Kubernetes :

-Complexité : peut être difficile à comprendre et à mettre en œuvre pour les développeurs moins expérimentés.
Ressources requises : peut exiger une grande quantité de ressources pour être déployé et exécuté efficacement.
-Maintenance et coûts supplémentaires : nécessite une maintenance régulière pour garantir son bon fonctionnement, ce qui peut entraîner des coûts supplémentaires.
-Configuration requise : nécessite une configuration spécifique pour être opérationnel, ce qui peut entraîner des problèmes de compatibilité avec certains environnements.

Source:

https://www.youtube.com/watch?v=X48VuDVv0do

https://www.youtube.com/watch?v=8kXz76QpGj8

https://fr.wikipedia.org/wiki/YAML

ChatGPT