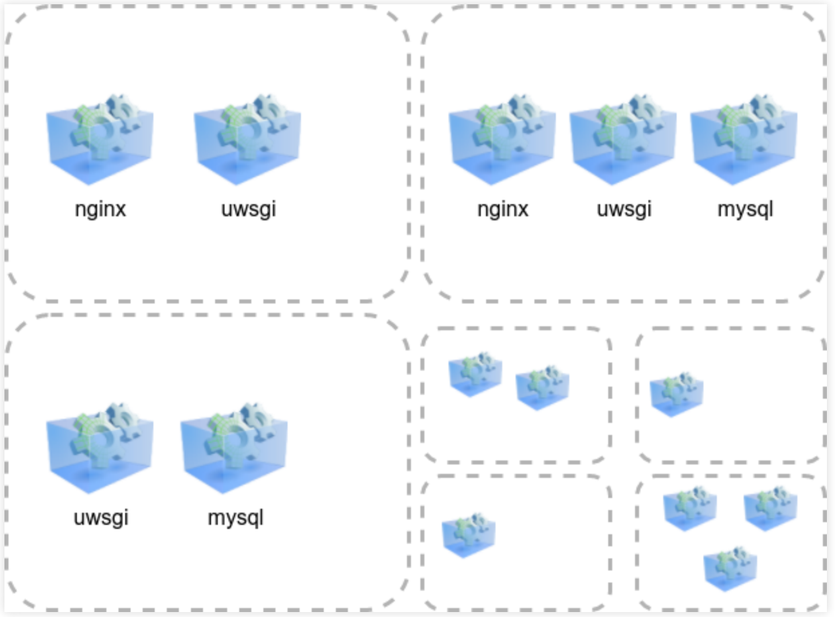
RunDigitalPlateform

Introduction

Manipuler quelques conteneurs sur des environnements de développement est une tâche facile.

Cela se complique en production lorsqu'on cherche de la HA, de la performance et/ou à mutualiser le matériel



Se posent alors les questions suivantes:

- Comment gérer les dysfonctionnements ?
- Comment gérer les déploiements et leurs emplacements ?
- Comment gérer le scaling ?
- Comment gérer les mises à jours ?
- Comment gérer la communication entre les conteneurs ?
- Comment gérer le stockage nécessaire à la persistance des données ?
- Comment gérer les secrets et la configuration ?

Tout gérer de manière manuelle et sans surcouche au système de conteneurs n'est pas viable, maintenable et pérenne.

Il nous faut un orchestrateur de conteneurs.

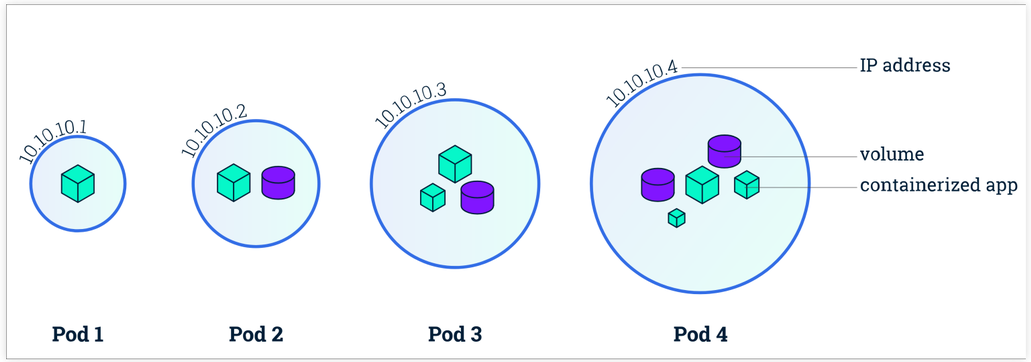
Historique

- Google annonce pour la première fois le projet Kubernetes à la mi-2014
- Kubernetes s'inspire du projet Borg (projet interne à Google pour gérer un cluster de jobs)
- Kubernetes 1.0 sort le 21 juillet 2015
- Un partenariat naît entre Google et la fondation Linux pour créer la Cloud Native Computing Fondation (CNCF)
- En 2017, Docker annonce l'intégration de Kubernetes dans les produits EE et CE.
- Kubernetes devient le premier projet promu de la CNCF en 2018

Concepts

Pods

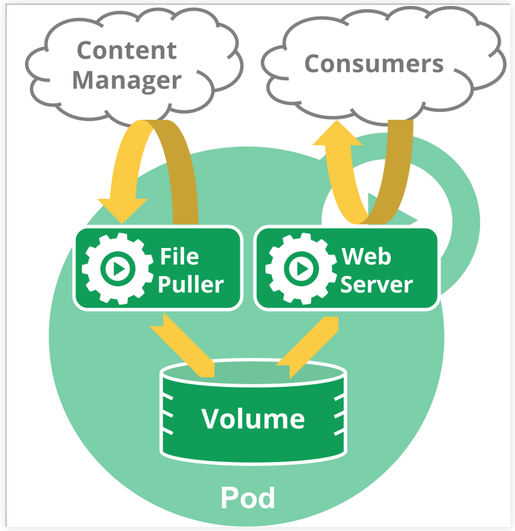
- Un pod est l'élement de base que manipule et ordonnance Kubernetes
- Ensemble de conteneurs colocalisés partageant des ressources



Ref: kubernetes.io

Les pods peuvent etre utilisés pour intégrer des stacks applicatives, mais le but premier du pod est d'ajouter des fonctions de support à un conteneur applicatif principal. On parle de sidecar. Par exemple pour :

- La gestion de contenu ou de cache
- La gestion de métriques, journaux, sauvegrade…
- La gestion d'évenements, de communication par queue…
- La mise en place de proxy



Ref: kubernetes.io

Déploiements

Kubernetes est déclaratif :

Vous dites ce que vous souhaitez… …et Kubernetes se débrouille pour obtenir l'état souhaité

En cas de divergence dans le temps ou de modification voulue de l'état, Kubernetes va converger vers l'état déclaré

Kubernetes utilise des structures de controle qui scrutent l'état du cluster.

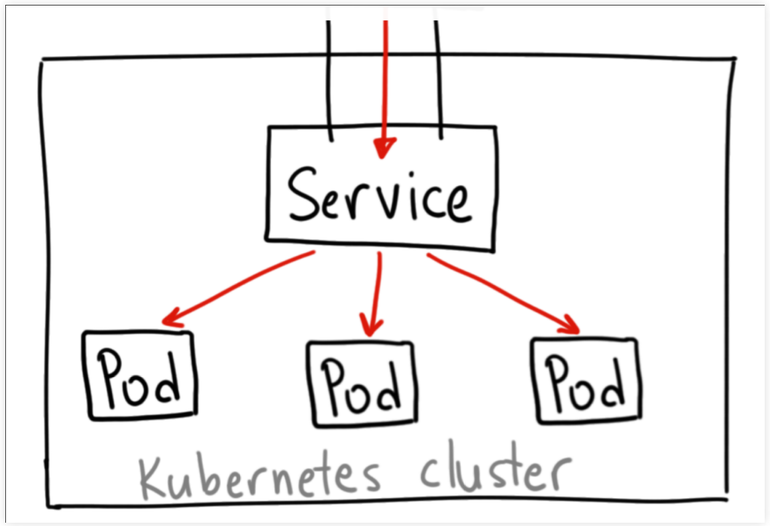
La gestion arative lancement/réplication/scaling) de pods d'une meme application nécéssite ainsi l'introduction de nouveaux objets : les déploiements

Architecture orientée service

L'architecture micro-services laisse place désormais à l'architecture orientée service (SOA)

- Les pods assurent la fonction du service (au travers des déploiements)
- Le service doit pouvoir etre exposé de manière stable aux autres services / à l'exterieur

Le service dans kubernetes permet d'exposer et de load-balancer nos pods via une adresse IP et un nom DNS associé.

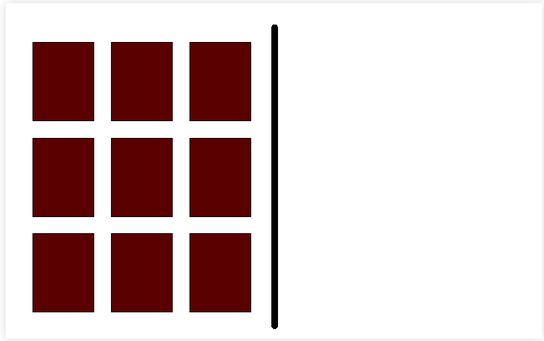


Ref: Ahmet Alp Balkan via medium.com

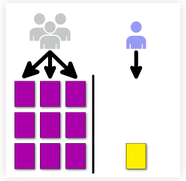
Mise à jour

La notion de service permet de faciliter les mises à jour de vos applications. Vous pouvez choisir parmis trois modes de déploiement:

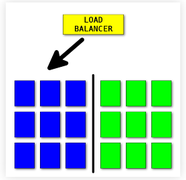
- rolling update (defaut)



- canary



- blue/green



That’s all Folks

Avez-vous des questions ?