

Docker vs Kubernetes

**Que sont les conteneurs
Kubernetes ?**

**Afin de déboguer l'application
qui s'y exécute, il faut d'abord
comprendre ce qu'est
Kubernetes et les conteneurs
Docker. Commençons donc
par une introduction rapide :**

Ce n'est rien d'autre qu'une plateforme d'orchestration de conteneurs ; c'est une plateforme open-source pour la mise à l'échelle, la gestion et le déploiement d'applications. L'automatisation du déploiement, de la mise à l'échelle et de la puissance des applications est l'objectif de cette plateforme.

Google l'a développée en 2014, et a contribué à la fondation de l'informatique en nuage, qui est gérée par elle actuellement. La raison pour laquelle tout le monde veut l'utiliser est qu'il est plus flexible.

Si vous avez de plus en plus de serveurs à ce moment-là, nous devons nous mettre au défi de gérer tous les conteneurs et de savoir comment nous en avons une idée et comment nous savons quelle application est exécutée sur quel serveur pour réduire cette complication, nous l'avons utilisé.

Qu'est-ce qu'un conteneur Docker ?

Il s'agit simplement d'une plateforme permettant de construire, d'expédier et d'exécuter des conteneurs. Il s'agit d'une plateforme open-source d'orchestration de conteneurs pour les conteneurs Docker qui est plus importante que Docker Swarm.

Les microservices se connectent aux deux. Et les deux sont utilisés comme plateforme open-source. Il s'agit d'un outil conçu pour faciliter la création, le déploiement et l'exécution d'applications à l'aide de conteneurs, de sorte que l'on puisse déboguer facilement les applications qui y sont exécutées.

Quelles sont ses principales caractéristiques de k8s?

Les principales caractéristiques de Kubernetes sont énumérées ci-dessous :

- Mise à l'échelle horizontale
- Auto-guérison
- Planification automatisée Équilibrage de la charge
- La distribution de la charge est l'équilibrage de la charge ; au niveau de la répartition, elle est facile à mettre en œuvre.
- Rollback et Rollout automatiques
- Orchestration du stockage L'architecture événementielle et l'architecture sans serveur définissent une nouvelle génération d'applications et de microservices.

**Quels sont ses composants et
son architecture ?**

Elle utilise une architecture client-serveur dans le diagramme où le maître joue le rôle de serveur et le nœud le rôle de client. Il s'agit d'une configuration multi-maître ou serveur possible par laquelle il n'y a qu'un seul serveur maître qui joue le rôle de contrôle du client/nœud. Le serveur et le client sont constitués de divers composants.

Ceux-ci sont décrits ci-dessous :

- Composant maître/serveur
- Un composant primaire et vital du nœud maître est le suivant :
- Ordonnanceur Kubernetes
- Gestionnaire de contrôleur
- Cluster etcd
- Composant client/nœud

**Les composants essentiels du
nœud client/nœud sont les
suivants :**

Pod

Kubelet

**L'architecture orientée
services aide les entreprises à
répondre plus rapidement et
plus efficacement à l'évolution
des conditions du marché.**

Débugger et développer des services localement dans Kubernetes

Le débogage sur Kubernetes consiste en différents services. Chaque service est exécuté dans son conteneur. Développer et déboguer des applications s'exécutant dans ses clusters peut être volumineux et lourd, pour cela il faut avoir un shell sur un conteneur en cours d'exécution puis tous vos outils s'exécutant dans le corps distant. La téléprésence est un outil qui permet de déboguer localement des applications sans aucune difficulté. La téléprésence nous permet d'utiliser des outils personnalisés tels que l'IDE et le débogueur.

Ce document décrit la téléprésence utilisée pour déboguer et développer des services qui s'exécutent localement sur un cluster. Les services de débogage et de développement ont besoin d'installer leur téléprésence en grappe et doivent également être installés.

**Comment développer et
débuguer des services
existants ?**

On fait le programme ou le débogage d'un seul service lorsqu'on développe une application sur celui-ci. Ces services nécessitent d'autres services pour déboguer l'application qui s'y exécute et la tester. Avec la téléprésence, le proxy Kube utilise l'option -swap-deployment pour échanger un déploiement existant. La permutation nous permet de nous connecter au cluster distant et nous permet d'exécuter un service localement en déboguant les applications qui s'y trouvent.

**Le secteur des technologies
de l'information s'oriente
aujourd'hui rapidement vers
l'utilisation de conteneurs
pour le développement de
logiciels.**

Quels sont ses avantages et ses limites ?

Vous trouverez ci-dessous les avantages et les limites mis en évidence :

Avantages du débogage dans Kubernetes

Le meilleur avantage est que les développeurs peuvent maintenant utiliser d'autres outils de sécurité Kubernetes pour le débogage, comme dans le repo Armador, utiliser l'outil de téléprésence ainsi que Ksync et Squash pour déboguer l'application.

Limites du débogage dans Kubernetes

C'est une partie standard du processus et du cycle de vie du développement que chaque développeur débogue localement. Mais lorsqu'il s'agit de le faire, cette approche devient plus difficile.

Il a son mécanisme d'orchestration et ses méthodologies d'optimisation lorsque les développeurs peuvent déboguer des microservices hébergés par des fournisseurs de cloud. Mais cette méthodologie de débogage des applications fonctionnant dans le nuage est excellente. Mais cela rend le débogage des applications dans Kubernetes plus difficile.

Conclusion

Le déploiement de Kubernetes est un concept de pods ; les pods ne sont rien d'autre que des nœuds qui ne sont rien d'autre que des serveurs où différents contenus peuvent être déployés dans un pod. Vous pouvez avoir un seul conteneur ou plusieurs conteneurs. Les pods contiennent plus de conteneurs. Il peut regrouper les conteneurs qui composent une application en unités logiques pour faciliter la gestion et la découverte en identifiant le nombre de nœuds.