

Pourquoi, quand et comment utiliser Kubernetes pour le développement d'applications Web

Résumé:

Qu'ont en commun Tinder, AirBnB, Pokemon Go et Pinterest ? Tous fonctionnent sur la plate-forme Kubernetes. Cet article explique pourquoi les responsables informatiques devraient évaluer l'utilisation de cette technologie, comment elle peut aider les entreprises et les circonstances dans lesquelles elle ne doit pas être utilisée.

Nous examinerons les cas d'utilisation de l'informatique d'entreprise dans le monde réel, ainsi que certains des défis auxquels cette technologie est confrontée, et comment utiliser correctement Kubernetes pour en tirer le meilleur parti.

Qu'est-ce que Kubernetes ?

Pour ceux qui se tiennent au courant des dernières tendances en matière de développement Web, il peut soudainement sembler que Kubernetes apparaisse partout, devenant de plus en plus populaire. Déployé pour la première fois auprès des développeurs en 2015, Kubernetes est progressivement devenu une plate-forme presque universellement appréciée des développeurs.

Kubernetes is in TOP3 platforms, the most loved by developers



Mais Kubernetes convient-il tout aussi parfaitement aux responsables technologiques ? Et si oui, pourquoi ?

[Kubernetes](#) est une plate-forme open source développée par Google, permettant aux utilisateurs de coordonner et d'exécuter des applications conteneurisées sur une série de plusieurs appareils ou machines.

L'objectif de Kubernetes est centré sur le contrôle total de l'ensemble du cycle de vie d'une application conteneurisée, avec des méthodes offrant une disponibilité et une évolutivité améliorées.

Les utilisateurs de Kubernetes sont en mesure de gérer la manière dont leurs applications interagissent avec d'autres applications en dehors de la plate-forme Kubernetes et de déterminer comment et quand ces applications s'exécutent.

Les utilisateurs peuvent facilement réduire ou augmenter selon les besoins, peuvent déployer des mises à jour de manière transparente, et peuvent tester des fonctionnalités et dépanner des déploiements difficiles en basculant le trafic entre plusieurs versions des applications.

QUELLE EST LA DIFFÉRENCE ENTRE KUBERNETES ET DOCKER ?

Fréquemment, les discussions sur Kubernetes et Docker prendront la forme de « Dois-je choisir d'utiliser Kubernetes ou Docker ? » Mais il s'agit d'un cadrage légèrement trompeur, car ces deux plates-formes ne sont pas des alternatives strictes. Il est tout à fait possible d'exécuter Kubernetes sans Docker, et vice versa. Mais Docker peut souvent bénéficier de l'utilisation simultanée de Kubernetes, et il en va de même dans l'autre sens.

[Docker](#) concerne les conteneurs et est fréquemment utilisé pour faciliter la conteneurisation des applications. Notez que Docker peut être installé sur n'importe quelle machine de manière autonome pour exécuter des applications conteneurisées.

Le cœur de la conteneurisation est le concept d'exécution d'une application d'une manière qui l'isole complètement des autres éléments du système. En substance, l'application agit comme si elle avait sa propre instance du système d'exploitation, même si dans la plupart des cas, plusieurs conteneurs se trouvent sur le même système d'exploitation. Docker agit en tant que gestionnaire de ce processus, gérant la création et l'exécution de ces conteneurs dans un système d'exploitation donné.

Maintenant, imaginez que vous utilisez Docker sur plusieurs hôtes. C'est là que Kubernetes peut entrer en jeu. Chaque hôte, ou nœud, peut être soit des machines virtuelles, soit des serveurs physiques. Kubernetes est capable d'automatiser et de gérer des éléments tels que l'équilibrage de charge, la mise à l'échelle et la sécurité, ainsi que l'approvisionnement des conteneurs, en reliant tous ces hôtes via un tableau de bord ou une ligne de commande unique. Nous appelons une collection d'hôtes Docker (ou tout groupe de nœuds) un cluster Kubernetes.

La question logique suivante est : pourquoi serait-il nécessaire de créer plus d'un hôte ou nœud ? Il y a deux raisons principales à cela.

1. Premièrement, rendre vos applications évolutives. Si la charge de travail sur l'application augmente, vous pouvez créer plus de conteneurs, ou bien augmenter le nombre de nœuds au sein du cluster Kubernetes.

2. Deuxièmement, augmenter la robustesse de l'infrastructure. Même si un nœud ou un sous-ensemble individuel de nœuds est déconnecté, votre application reste en ligne.

Kubernetes est capable d'automatiser la suppression, l'ajout, la gestion et la mise à jour des conteneurs. En fait, Kubernetes peut peut-être mieux être considéré comme une plate-forme pour [orchestrer des conteneurs](#). Pendant ce temps, Docker permet la création et la maintenance de conteneurs pour commencer, à un niveau inférieur.

Quand et pourquoi utiliser Kubernetes

En commençant par une série de conteneurs créés par Docker, Kubernetes gère le trafic et alloue des ressources pour ce service.

Ce faisant, il simplifie et simplifie de nombreux aspects de la gestion d'une infrastructure d'applications orientée services. En conjonction avec les derniers outils CI/CD (intégration continue/déploiement continu), Kubernetes peut faire évoluer ces applications sans nécessiter un projet d'ingénierie majeur. Nous pouvons examiner quelques-uns de ces cas de plus près.

ORCHESTRATION DE CONTENEURS

Les conteneurs, dans le vide, sont merveilleux, offrant un moyen simple de regrouper et de déployer vos services avec un processus de création léger. Cela permet des attributs positifs tels que l'utilisation efficace des ressources, l'immuabilité et l'isolation des processus.

Cependant, lorsque vous mettez réellement en œuvre votre projet en production, il est très possible que vous vous retrouviez avec des dizaines à des milliers de conteneurs, en particulier au fil du temps. Si ce travail est effectué

manuellement, il est probable qu'une équipe de gestion de conteneurs dédiée soit nécessaire pour mettre à jour, connecter, gérer et déployer ces conteneurs. La simple exécution des conteneurs n'est pas suffisante. Vous devrez également être capable d'accomplir les tâches suivantes :

- Échelle à la baisse ou à la hausse en fonction de l'évolution de la demande
- Orchestrer et intégrer diverses parties modulaires
- Communiquer entre les clusters
- Assurez-vous que vos conteneurs sont tolérants aux pannes

Il est naturel de penser que les conteneurs devraient pouvoir gérer tout cela sans avoir besoin d'une plate-forme comme Kubernetes. Mais la réalité est que les conteneurs eux-mêmes fonctionnent à un niveau d'organisation inférieur. Afin de bénéficier des avantages significatifs d'un système construit avec des conteneurs, il est nécessaire d'utiliser des outils d'orchestration de conteneurs tels que Kubernetes, des outils qui reposent sur et gèrent les conteneurs.

GESTION DES MICROSERVICES

Le concept de microservices est loin d'être une idée nouvelle. Les architectes logiciels ont travaillé à diviser les applications à grande échelle en composants largement réutilisables pendant des décennies. Les microservices offrent de multiples avantages, allant d'un degré de résilience plus élevé à des modèles de déploiement plus rapides et plus flexibles en passant par des tests automatiques plus simples.

Les microservices permettent également aux décideurs de choisir le meilleur outil pour chaque tâche individuelle. Une partie de votre application peut bénéficier davantage de l'augmentation de la productivité d'un langage de haut niveau comme PHP, tandis qu'une autre partie peut bénéficier davantage d'un langage à grande vitesse comme Go.

Mais comment Kubernetes s'intègre-t-il dans ce concept de microservices ? Il est vrai que décomposer votre application à grande échelle en ces microservices plus petits et moins connectés de manière rigide permettra plus de liberté et d'indépendance d'action. Mais il est toujours nécessaire que votre équipe se coordonne tout en utilisant l'infrastructure que toutes ces pièces indépendantes utilisent pour fonctionner.

Il sera nécessaire de prédire la quantité de ressources dont ces pièces auront besoin, ainsi que la façon dont vos besoins évoluent sous la charge. Vous devrez allouer des partitions au sein de votre infrastructure pour vos différents microservices et restreindre l'utilisation des ressources en conséquence.

C'est là que Kubernetes peut vous aider, en offrant un cadre commun qui donne une description de votre architecture d'infrastructure qui vous permet d'inspecter et de résoudre les problèmes d'utilisation et de partage des ressources. Si vous envisagez de créer une architecture basée sur les microservices, Kubernetes peut être extrêmement bénéfique.

GESTION DE L'ENVIRONNEMENT CLOUD

Il n'est pas étonnant que les conteneurs et les outils qui gèrent les conteneurs soient devenus de plus en plus populaires, alors que de nombreuses entreprises modernes se tournent vers des modèles basés sur des microservices. Ces modèles de microservices permettent de diviser facilement une application en éléments discrets répartis dans des conteneurs exécutés via des environnements cloud séparés. Cela vous permet de choisir un hébergeur qui correspond parfaitement à vos besoins dans chaque cas.

Kubernetes est conçu pour être déployé n'importe où, ce qui signifie que vous pouvez l'utiliser sur un cloud privé, un cloud public ou un cloud hybride. Cela vous permet de vous connecter avec vos utilisateurs, peu importe où ils se trouvent, avec une sécurité accrue en plus. Kubernetes vous permet d'éviter le problème éventuel du «verrouillage du fournisseur ».

Que signifie « verrouillage du fournisseur » ?

Le verrouillage du fournisseur fait référence à une situation dans laquelle le coût du passage à un autre fournisseur est si élevé que le client est essentiellement coincé avec le fournisseur d'origine. En raison de pressions financières, d'une main-d'œuvre insuffisante ou de la nécessité d'éviter les interruptions des opérations commerciales, le client est « enfermé » dans ce qui peut être un produit ou un service de qualité inférieure.

Imaginez qu'un bureau fasse apporter du café par un vendeur de café et que ce vendeur ait besoin de machines à café spécifiques dans le bureau que seul le vendeur vend. Imaginez maintenant qu'il y a une forte baisse de la qualité du café que ce vendeur livre. Le passage à un nouveau fournisseur de café signifierait que les anciennes machines qu'ils ont achetées deviendraient inutiles, car le changement nécessite probablement l'achat d'un nouvel équipement de fabrication de café. Compte tenu des tracas et des dépenses supplémentaires liés au remplacement de chaque machine à café, les employés du bureau seraient effectivement enfermés dans leur accord avec leur ancien fournisseur et contraints de boire un café de qualité inférieure.

Un exemple concret de verrouillage du fournisseur est la façon dont Apple a contraint les consommateurs à utiliser iTunes au début du service, car la musique achetée via iTunes ne pouvait être lue que dans l'application iTunes ou sur un iPod.

Qu'est-ce que le verrouillage du fournisseur dans le cloud computing ?

Dans le cloud computing, une partie du logiciel ou de l'infrastructure informatique est sous-traitée à un fournisseur de cloud computing, qui le propose en tant que service et le fournit via Internet. Par exemple, les serveurs hébergés dans le cloud sont Infrastructure-as-a-Service (IaaS) et les applications hébergées dans le cloud sont Software-as-a-Service (SaaS).

Parfois, une entreprise peut se retrouver enfermée dans un certain fournisseur de cloud. Le verrouillage du fournisseur peut devenir un problème dans le cloud computing car il est très difficile de déplacer les bases de données une fois qu'elles sont configurées, en particulier dans une migration vers le cloud, qui implique le déplacement des données vers un type d'environnement totalement différent et peut impliquer le reformatage des données. De plus, une fois qu'un logiciel tiers est intégré aux processus d'une entreprise, l'entreprise peut devenir dépendante de ce logiciel.

Pourquoi le verrouillage du fournisseur est-il un problème ?

Un certain nombre de circonstances peuvent avoir un impact négatif sur une entreprise si elle est liée à un certain fournisseur de cloud :

- Si la qualité de service d'un fournisseur décline, ou n'atteint jamais un seuil souhaité pour commencer, le client sera coincé avec elle
- Le fournisseur peut également modifier radicalement ses offres de produits de telle sorte qu'elles ne répondent plus aux besoins d'une entreprise
- Un vendeur peut fermer complètement ses portes
- Enfin, un vendeur peut imposer des augmentations de prix massives pour le service, sachant que ses clients sont bloqués

Dans l'ensemble, transférer une technologie fondamentale et critique à un fournisseur externe n'est pas facile pour une entreprise, et cela nécessite un grand degré de confiance dans le fournisseur.

Comment les entreprises peuvent-elles éviter les risques de blocage des fournisseurs ?

- Évaluez soigneusement les services cloud : les entreprises doivent rechercher en profondeur un fournisseur de cloud avant de s'engager, idéalement avec un déploiement de preuve de concept pour s'assurer que leur niveau de service est suffisant.

- Assurez-vous que les données peuvent être déplacées facilement : les entreprises utilisant le cloud computing doivent s'efforcer de garder leurs données portables ou faciles à déplacer d'un environnement à un autre. Ils peuvent le faire en partie en définissant clairement leurs modèles de données et en conservant les données dans des formats utilisables sur diverses plates-formes, plutôt que dans des formats spécifiques à un fournisseur donné.
- Sauvegardes : conserver des sauvegardes internes de toutes les données aide une entreprise à rester prête à héberger les données ailleurs s'il est trop difficile ou trop long de les extraire du service cloud (ainsi qu'à fournir une certaine protection contre les ransomwares).
- Stratégie multi-cloud ou cloud hybride : une approche multi-cloud intègre plusieurs fournisseurs de cloud, réduisant la dépendance vis-à-vis d'un seul fournisseur. Dans un cloud hybride, certaines données resteront sous le contrôle direct d'une organisation, soit dans un cloud privé, soit stockées sur site.

Quand ne pas utiliser Kubernetes

Bien que Kubernetes soit puissant et flexible, ce n'est pas le bon outil pour chaque cas d'utilisation ou projet. Gardez à l'esprit que Kubernetes a été créé pour résoudre un certain nombre de problèmes et de défis potentiels. Si vous n'êtes pas réellement confronté à ces défis, il est très possible que Kubernetes soit plus difficile à manier qu'utile.

Si votre projet semble atteindre un point où la mise à l'échelle et le déploiement nécessitent sa propre ressource dédiée, l'orchestration commence à devenir un choix viable.

Les concepts d'orchestration et d'automatisation ont tendance à aller de pair. L'automatisation profite aux entreprises en augmentant l'efficacité,

permettant une diminution des interactions humaines avec les systèmes numériques. Avec un logiciel gérant ces interactions à la place, nous constatons une réduction des erreurs et des coûts.

En règle générale, l'automatisation signifie l'automatisation d'une tâche discrète. En revanche, l'orchestration fait référence à l'automatisation d'un flux de travail ou d'un processus complet composé de plusieurs étapes et souvent de plusieurs systèmes. Tout d'abord, vous intégrez l'automatisation dans un flux de travail. Ensuite, si la complexité augmente jusqu'à un certain point, l'orchestration prend le relais pour les gouverner et les faire fonctionner de concert.

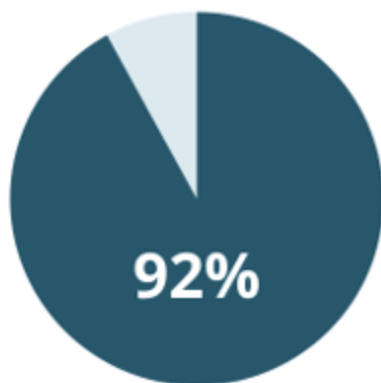
Kubernetes n'est **pas le meilleur choix** pour :

1. Projets simples ou à petite échelle, car c'est assez cher et trop compliqué.
2. Des projets qui comportent une petite base d'utilisateurs, une faible charge, une architecture simple et aucun plan pour augmenter tout cela.
3. Développer une version MVP, car il vaut mieux commencer par quelque chose de plus petit et de moins complexe, comme Docker Swarm. Ensuite, après avoir solidifié une vision du fonctionnement de votre application, vous pouvez envisager de passer à Kubernetes.

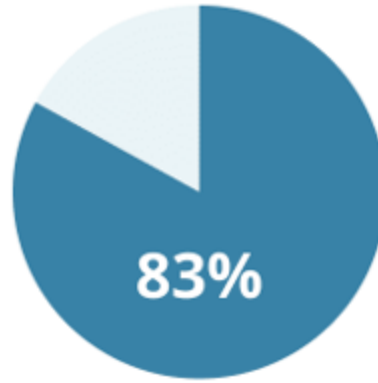
Comment Kubernetes impacte l'informatique d'entreprise en 2021

L'enquête Cloud Native, qui interroge les organisations technologiques/logicielles, [rapporte](#) que l'utilisation de conteneurs dans les productions a augmenté de 84 % par rapport à l'année précédente, jusqu'à 92 %. L'utilisation de Kubernetes est en hausse de 78 % par rapport à l'année précédente, atteignant 83 %.

Kubernetes adoption in 2020



of organizations used
containers in production



of organizations used
Kubernetes for containers orchestration

Comment les grandes entreprises utilisent-elles Kubernetes ?

Pokemon Go a [travaillé](#) sur GKE (Google Container Engine) à l'aide de Kubernetes, ce qui lui a permis d'orchestrer un cluster de conteneurs couvrant le monde entier. Cela a permis à son équipe de se concentrer plus efficacement sur la mise en place de changements en direct pour sa base de joueurs actifs de plus de 20 millions d'utilisateurs quotidiens.

Tinder a [effectué la transition](#) de ses services hérités vers Kubernetes, un cluster composé de plus de 1 000 nœuds et 200 services, avec 15 000 pods et 48 000 conteneurs. Cela a réduit le temps d'attente pour les instances EC2 de plusieurs minutes à quelques secondes, offrant des réductions significatives des coûts.

L'avenir de Kubernetes

L'architecture distribuée et l'évolutivité de Kubernetes se marient bien avec l'apprentissage automatique et l'intelligence artificielle. Alors que ces technologies continuent de mûrir, 2021 est l'année à surveiller pour la croissance dans cet espace.

Les entreprises doivent garder à l'esprit que les outils soutiennent les objectifs commerciaux et ne sont pas une fin en soi. 2020 a été une année au cours de laquelle presque toutes les entreprises ont dû faire face à des changements inattendus. Kubernetes a la capacité d'accélérer les services de développement d'applications grâce à des solutions construites avec un écosystème natif du cloud, tout en permettant une utilisation malléable des applications et des données, afin qu'une entreprise puisse réussir grâce à la modernisation de ses plates-formes et applications.

Références :

<https://mobidev.biz/blog/>