

FAVORISER LA TRANSFORMATION DIGITALE GRÂCE AUX CONTENEURS ET À KUBERNETES

Comment Kubernetes gère les applications conteneurisées de façon à générer des avantages stratégiques

Sommaire

Introduction.....	3
La transformation digitale et l'évolution vers les applications conteneurisées	3
Applications Cloud et applications à 12 facteurs	4
Méthodologie de déploiement de logiciels SaaS (Software as a Service).....	5
Présentation concise de Kubernetes.....	6
Modèle objet Kubernetes.....	7
Maintien de l'état souhaité.....	7
Avantages stratégiques de Kubernetes	8
Kubernetes au service des applications Cloud et des applications à 12 facteurs	8
Exemple de cas d'usage.....	10
Solutions VMware en matière de technologie de conteneurs.....	11
vSphere Integrated Containers	11
Wavefront by VMware.....	13
Pivotal Container Service.....	13
Conclusion	14

QU'EST-CE QU'UN CONTENEUR DOCKER ?

Avec les conteneurs, Docker a défini un format standard de packaging et de portage de logiciels, tout comme les conteneurs ISO définissent une norme pour le fret. Un conteneur est une instance d'exécution d'une image Docker, et réunit trois composants :

- Une image Docker
- Un environnement d'exécution pour cette image
- Un ensemble d'instructions relatives à l'exécution de cette image

- Adapté du [Glossaire Docker](#)

Introduction

Kubernetes assure la gestion des conteneurs. Un conteneur regroupe une application et ses dépendances dans une image distribuable pouvant être exécutée pratiquement partout, de façon à simplifier le développement et le déploiement de logiciels. Les entreprises qui adoptent les conteneurs franchissent une étape décisive dans leur transformation en entreprises numériques agiles axées sur la fourniture accélérée de produits, de services et d'expériences clients innovants. Au lieu de subir la perturbation, ces entreprises se donnent les moyens de devenir elles-mêmes des forces perturbatrices.

Les conteneurs engendrent cependant des problèmes spécifiques de gestion des technologies, en particulier lorsque des applications conteneurisées doivent être déployées et gérées à grande échelle : c'est là que Kubernetes intervient. Kubernetes orchestre les applications conteneurisées de façon à gérer et automatiser l'utilisation des ressources, le traitement des défaillances, la disponibilité, la configuration, l'évolutivité et la gestion de l'état souhaité.

Ce livre blanc décrit la solution Kubernetes et présente ses avantages stratégiques, puis examine l'un de ses cas d'usage et vous montre comment cette solution peut accélérer la transformation digitale de votre entreprise.

La transformation digitale et l'évolution vers les applications conteneurisées

Selon le New York Times, l'innovation technologique est actuellement en phase d'accélération et d'expansion¹. C'est pourquoi la transformation digitale fait désormais partie des objectifs de nombreuses entreprises, tandis que l'adoption d'initiatives numériques tend à se généraliser.²

Pour les entreprises qui ont engagé une transformation digitale, les motivations sont claires :

- Créer de nouvelles applications permettant d'interagir avec les clients de manière innovante et captivante
- Améliorer les opérations pour permettre à l'entreprise de fournir plus efficacement de meilleurs produits et des services à coût réduit
- Générer de nouveaux flux de revenus en s'adaptant rapidement aux évolutions du marché et des préférences des consommateurs

Gartner Research estime que « l'avenir appartiendra aux sociétés capables de créer les solutions logicielles les plus efficaces, les plus intelligentes et les plus autonomes »³. Les ingrédients à mettre en œuvre pour générer des applications efficaces et autonomes sont cependant moins clairement définis que les résultats poursuivis.

Pour garantir une pleine efficacité en ce domaine, il importe d'utiliser une architecture d'applications qui favorise la fluidité, la rapidité et la réactivité du développement et du déploiement, sans toutefois remettre en question les niveaux de sécurité, de performances et de rentabilité assurés par les modèles établis. Les conteneurs fournissent les bases d'une nouvelle architecture d'applications qui soutient la transformation digitale et crée une fondation solide pour l'innovation.

1 « [Digital Transformation Going Mainstream in 2016, IDC Predicts](#) », Steve Lohr, The New York Times, 4 novembre 2015.

2 « [New Research Finds Investment from Outside IT Is Key to Digital Transformation Success](#) », source : Business Wire, The New York Times, 11 mai 2017.

3 « [Digital Transformation Going Mainstream in 2016, IDC Predicts](#) », Steve Lohr, The New York Times, 4 novembre 2015.

Les entreprises adoptent de plus en plus la technologie de conteneurs. Une étude récente de 451 Research a ainsi révélé un profil de croissance de l'adoption réellement impressionnant compte tenu du caractère émergent de cet écosystème⁴. Les entreprises qui adoptent les conteneurs considèrent cette technologie comme une voie rapide vers la génération et le déploiement d'applications Cloud et d'applications à 12 facteurs.

Applications Cloud et applications à 12 facteurs

Selon la Cloud Native Computing Foundation (un projet de la Fondation Linux), les propriétés qui définissent les applications Cloud sont les suivantes⁵ :

1. **Conteneurisation** : chaque élément (application, processus, etc.) est packagé dans son propre conteneur. Cette propriété favorise la reproductibilité et la transparence, ainsi que l'isolation des ressources.
2. **Orchestration dynamique** : les conteneurs sont planifiés et gérés activement de façon à optimiser l'utilisation des ressources.
3. **Orientation microservices** : les applications sont segmentées en microservices. Cette segmentation améliore sensiblement l'agilité et la facilité de maintenance globales des applications.

Kubernetes couvre la seconde partie de cette définition en assurant la planification et la gestion des conteneurs. En ce qui concerne la première partie, Kubernetes et Docker contribuent conjointement à la mise en œuvre des microservices.

L'élément clé réside toutefois dans le conteneur, un processus qui s'exécute sur un ordinateur ou une machine virtuelle avec ses propres application, système de fichiers et réseau autodéfinis et isolés. Un conteneur permet de packager une application sous une forme reproductible : sa distribution et sa réutilisation ne nécessitent qu'un effort minime.

Les conteneurs Docker sont ceux qui sont les plus largement déployés. Un manifeste, appelé fichier Dockerfile, décrit les modalités selon lesquelles l'image et ses composantes doivent être exécutées au sein d'un conteneur hébergé sur un hôte. Afin d'illustrer concrètement ces notions de fichier Dockerfile et d'image, voici un exemple de fichier Dockerfile installant MongoDB sur une machine Ubuntu exécutée dans un conteneur. Les lignes commençant par un signe dièse sont des commentaires décrivant les commandes qui suivent.

```
# MongoDB Dockerfile depuis https://github.com/dockerfile/mongodb
# Extraire l'image de base.
FROM dockerfile/ubuntu
# Installer MongoDB.
RUN \
  apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv
  7F0CEB10 && \
  echo 'deb http://downloads-distro.mongodb.org/repo/ubuntu-
  upstart dist 10gen' > /etc/apt/sources.list.d/mongodb.list && \
  apt-get update && \
  apt-get install -y mongodb-org && \
  rm -rf /var/lib/apt/lists/*
# Définir les répertoires montables.
```

⁴ « Application containers will be a \$2.7bn market by 2020, representing a small but high-growth segment of the CloudEnabling Technologies market », 451 Research, 10 janvier 2017.

⁵ Cette définition est extraite de la FAQ de la Cloud Native Computing Foundation, <https://www.cncf.io/about/faq/>.

```
VOLUME ["/data/db"]
# Définir le répertoire de travail.

WORKDIR /data
# Définir la commande par défaut.
CMD ["mongod"]
# Exposer le port 27017 pour le processus et le port 28017
pour http
EXPOSE 27017
EXPOSE 28017
```

Méthodologie de déploiement de logiciels SaaS (Software as a Service)

En revanche, les applications à 12 facteurs se définissent autant par leurs processus que par leurs propriétés. Il s'agit d'une méthodologie consistant à développer une application SaaS (Software as a Service), autrement dit une application Web, pour la déployer le plus souvent sur une plate-forme sous forme de service (PaaS), telle que Pivotal Cloud Foundry. Voici la liste des 12 facteurs avec une brève explication pour chacun d'eux⁶ :

1. **Déployer l'application un grand nombre de fois à partir d'une seule base de code.** La base de code est stockée dans un référentiel, où elle est gérée via un système de contrôle des versions tel que Git au fur et à mesure de ses modifications. De nombreuses instances d'exécution de l'application sont ensuite déployées à partir de cette même base de code. Il s'ensuit que le déploiement se poursuit fréquemment dans trois environnements : l'environnement local de chaque développeur, un environnement de préproduction et l'environnement de production.
2. **Déclarer et isoler les dépendances.** Plutôt que de s'appuyer implicitement sur des packages définis à l'échelle du système, l'application déclare les dépendances via un manifeste de déclaration. Cette déclaration explicite des dépendances permet aux développeurs de configurer plus facilement leur environnement de développement.
3. **Stocker la configuration dans l'environnement et non dans le code.** Les informations de configuration qui varient d'un déploiement sur l'autre sont stockées par l'application dans des variables d'environnement. Ces variables d'environnement sont des paramètres de contrôle granulaire gérés de manière indépendante à chaque déploiement, ce qui permet de faire évoluer facilement l'application en ajoutant progressivement des déploiements.
4. **Connecter l'application à des services de support.** tels que base de données ou système de stockage, plutôt que d'incorporer ces services dans le code. L'application traite ces services comme des ressources pouvant être associées ou dissociées d'un déploiement par simple modification de la configuration.
5. **Traiter la génération et l'exécution comme des étapes distinctes.** Chaque déploiement de la base de code s'effectue en trois étapes : génération, release et environnement d'exécution. L'étape « génération » convertit la base de code en exécutable (ou génération), tandis que la release combine l'exécutable généré avec la configuration pour produire une release prête à être exécutée dans l'environnement d'exécution.
6. **Exécuter l'application comme un ensemble de processus sans état.** Les processus ne partagent rien entre eux, et les données qui doivent persister sont stockées dans une base de données exécutée sous la forme d'un service de support sans état.

⁶ La présentation des douze facteurs est librement adaptée des descriptions proposées sur le site Web [The Twelve-Factor App](#).

GÉRER DES APPLICATIONS CONTENEURISÉES AVEC KUBERNETES

Kubernetes orchestre les applications conteneurisées distribuées de manière à :

- Optimiser l'utilisation des ressources de calcul
- Fournir des règles pour la planification
- Maintenir l'état souhaité
- Automatiser le traitement des défaillances et des pannes
- Assurer la haute disponibilité
- Surveiller les travaux en temps réel
- Gérer la configuration d'une application
- Assurer une évolutivité dynamique pour répondre aux fluctuations de la demande

7. **Exposer les services par liaison de port.** Par exemple, dans le cas du protocole HTTP, l'application exporte le traitement HTTP sous forme de service en établissant une liaison avec un port, puis en écoutant sur ce port pour détecter les requêtes entrantes.
8. **Assurer l'évolutivité horizontale par ajout de processus simultanés.** L'application traite les charges de travail en affectant chaque type de travail à un type de processus. Ainsi, les processus Web traitent des requêtes HTTP, tandis que les processus Worker gèrent des tâches d'arrière-plan.
9. **Garantir la durabilité au moyen de processus « jetables ».** Les processus présentent un caractère jetable dans la mesure où il est possible de les démarrer ou de les arrêter rapidement pour effectuer facilement les modifications ou évolutions de l'application.
10. **Homogénéiser le développement et la production.** L'application est conçue pour le déploiement continu dans la mesure où les développeurs peuvent intégrer rapidement le nouveau code et déployer eux-mêmes l'application dans un environnement de production. Il importe donc que les environnements de développement et de production soient aussi similaires que possible.
11. **Traiter les journaux et les flux d'événements.** Au lieu d'assurer l'acheminement ou le stockage du flux de sortie de ses journaux, l'application consigne celui-ci sous la forme d'un flux de données dirigé vers une sortie standard, où il est recueilli par l'environnement d'exécution puis acheminé vers un outil ou système spécialisé comme Hadoop pour stockage ou analyse.
12. **Exécuter les scripts et les tâches de gestion à caractère ponctuel.** tels que les migrations de base de données, dans un environnement identique à celui des processus longue durée de l'application.

L'utilisation des conteneurs et de Kubernetes permet de répondre à certains aspects de ces impératifs. Par exemple, les conteneurs jouent un rôle clé dans le cadre des applications à 12 facteurs dans la mesure où ils vous permettent de déclarer et d'isoler les dépendances. Les conteneurs contribuent également à la durabilité fondée sur les processus jetables, notamment par leur capacité à démarrer rapidement et à s'arrêter en douceur. Nombre des autres facteurs sont pris en charge par Kubernetes.

Présentation concise de Kubernetes

Kubernetes a été développé à l'origine par Google. Google utilise le prédécesseur de Kubernetes, nommé Borg, pour le lancement, la planification, le redémarrage et la surveillance de ses applications publiques, telles que Gmail et Google Docs, ainsi que pour certains frameworks applicatifs internes comme MapReduce⁷. Fondé sur le système initial de Google amélioré grâce aux enseignements retirés de Borg, Kubernetes est un système d'orchestration Open Source pour conteneurs capable de fonctionner aussi bien dans votre Data Center que sur les différents types de Cloud ou au sein d'un Data Center hybride. Kubernetes assure automatiquement le placement de charges de travail, le redémarrage d'applications et l'ajout de ressources en fonction de la demande.

Voici une description succincte de son principe de fonctionnement. Un cluster Kubernetes contient un nœud maître et plusieurs nœuds de travail. Lorsque vous déployez une application sur ce cluster, les composants de cette application s'exécutent au sein des nœuds de travail. Le nœud maître gère le déploiement.

Kubernetes comprend les composants suivants :

- L'API Kubernetes
- L'interface de ligne de commande Kubernetes, nommée kubectl
- Le plan de contrôle Kubernetes

⁷ [Large-Scale Cluster Management at Google with Borg](#), étude Google, 2015.

AVANTAGES LIÉS À L'UTILISATION DE KUBERNETES

- Consolider les serveurs et réduire les coûts via une utilisation efficace des ressources
- Gérer efficacement les défaillances liées aux machines virtuelles via des fonctions d'autoréparation et de haute disponibilité
- Faciliter et accélérer le déploiement, la journalisation et la surveillance des applications
- Automatiser l'évolutivité des conteneurs et des applications conteneurisées
- Dissocier les applications des machines afin de garantir la portabilité et la flexibilité
- Faciliter la modification, la mise à jour, l'extension ou le redéploiement de certaines applications sans incidence sur les autres charges de travail

Ce plan de contrôle englobe les processus exécutés sur le nœud maître et sur les nœuds de travail de Kubernetes. Par exemple, Kubernetes exécute plusieurs processus sur le nœud maître : le serveur d'API, le contrôleur, le planificateur et le processus etcd. Les nœuds de travail exécutent le processus kubelet pour communiquer avec le nœud maître et le processus proxy afin d'assurer la gestion réseau.

Modèle objet Kubernetes

L'une des caractéristiques essentielles du système Kubernetes réside dans son mode de représentation de l'état des applications et des charges de travail conteneurisées qui ont été déployées. Kubernetes représente l'état au moyen de différents « objets », tels que service, espace de nommage ou volume. Ces objets sont généralement définis par une spécification d'objets (ou « spec »), que vous créez à l'échelle de votre cluster.

Dans le mode objet de Kubernetes, le concept de pod correspond à la forme la plus élémentaire de composant déployable. Un pod représente une instance d'application exécutée sous forme de processus sur un cluster Kubernetes. C'est là que l'environnement d'exécution Docker entre à nouveau en ligne de compte : en effet, Docker est généralement utilisé pour constituer l'environnement d'exécution d'un pod Kubernetes.

Kubernetes comprend également un ensemble de contrôleurs qui mettent en œuvre l'essentiel de sa logique fonctionnelle. Ce sont les contrôleurs qui fournissent des fonctionnalités comme les API ReplicaSet et StatefulSet.

Maintien de l'état souhaité

Le plan de contrôle Kubernetes gère l'état de tous ces objets afin de maintenir ceux-ci dans l'état souhaité que vous avez spécifié. Pour spécifier un état souhaité, il vous suffit de créer une spécification d'objets pour un service au moyen d'un fichier YAML. En voici un exemple :

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-demo-service
  labels:
    app: nginx-demo
spec:
  type: NodePort
  ports:
    - port: 80
      protocol: TCP
      name: http
  selector:
    app: nginx-demo
---
apiVersion: v1
kind: ReplicationController
metadata:
  name: nginx-demo
spec:
  replicas: 3
```

```
template:  
metadata:  
  labels:  
    app: nginx-demo  
spec:  
  containers:  
  - name: nginx-demo  
    image: myrepo/nginx  
ports:  
  - containerPort: 80
```

Lorsque vous soumettez un tel fichier au nœud maître Kubernetes via l'interface de ligne de commande kubectl, le plan de contrôle Kubernetes implémente les instructions contenues dans ce fichier en démarrant et en planifiant les applications de telle sorte que l'état du cluster corresponde à l'état souhaité que vous avez spécifié. Le nœud maître et le plan de contrôle Kubernetes conservent alors l'état souhaité via l'orchestration des nœuds du cluster, qui peuvent correspondre à des serveurs réels ou à des machines virtuelles.

La pièce centrale de l'architecture est un serveur d'API qui gère l'état des objets du système. Ce serveur d'API coopère avec différents sous-composants Kubernetes (ou clients Kubernetes) générés sous forme de microservices composables, tels que le contrôleur de réplication spécifié dans le fichier YAML. Ce contrôleur de réplication assure la régulation de l'état souhaité des répliques des pods en cas de défaillance.

Avantages stratégiques de Kubernetes

Pour revenir à la transformation digitale, Kubernetes s'appuie sur cette architecture pour gérer les applications conteneurisées déployées dans un cluster distribué. Il contribue ainsi à concrétiser les avantages stratégiques promis par la transformation digitale.

- Avec Kubernetes, il devient plus simple et plus économique d'exécuter des applications dans des Clouds publics, privés ou hybrides.
- Kubernetes accélère le développement et le déploiement des applications.
- Kubernetes améliore l'agilité, la flexibilité et la capacité d'adaptation au changement.

Kubernetes au service des applications Cloud et des applications à 12 facteurs

Kubernetes assure un fonctionnement gérable des applications conteneurisées à grande échelle. Rappelez-vous la seconde partie de la définition des applications Cloud : celles-ci sont orchestrées dynamiquement de telle sorte que les conteneurs soient planifiés et gérés activement en vue d'optimiser l'utilisation des ressources. C'est exactement ce que fait le système Kubernetes. Il orchestre les conteneurs et leurs charges de travail de façon à optimiser l'utilisation des machines virtuelles et des serveurs physiques qui constituent les nœuds d'un cluster.

Il suffit de reprendre un à un les 12 facteurs pour comprendre encore mieux à quel point Kubernetes simplifie la gestion des applications. De façon générale, Kubernetes est capable de déployer et d'exécuter les applications à 12 facteurs.

COMMENT KUBERNETES ET LES CONTENEURS SIMPLIFIENT LA GESTION DES APPLICATIONS	
Facteur	Avantage
1. Déployer l'application un grand nombre de fois à partir d'une seule base de code.	Kubernetes permet de déployer de nombreuses fois une application à partir d'une base de code unique : il suffit pour cela d'attribuer à un pod une spécification contenant la référence d'une image de conteneur.
2. Déclarer et isoler les dépendances.	Les conteneurs sont capables d'exprimer les dépendances.
3. Stocker la configuration dans l'environnement et non dans le code.	Vous avez la possibilité de stocker certains aspects de la configuration d'une application dans l'environnement Kubernetes. Par exemple, la structure ConfigMaps permet de séparer les objets de configuration des instructions relatives à une image.
4. Connecter l'application à des services de support, tels qu'une base de données, plutôt que d'incorporer ces services dans le code.	Kubernetes vous permet de déployer des services de support, tels qu'une base de données, au sein de conteneurs distincts, puis gère conjointement l'ensemble des composants conteneurisés de façon à garantir la disponibilité et les performances.
5. Traiter la génération et l'exécution comme des étapes distinctes.	Par exemple, vous pouvez générer l'application via Jenkins (un serveur d'automatisation de pipeline distinct de Kubernetes), puis exécuter les images Docker à l'aide de Kubernetes.
6. Exécuter l'application comme un ensemble de processus sans état.	Kubernetes facilite l'exécution d'applications sans état. Kubernetes permet de conserver les états de manière indépendante (par exemple, dans une banque de données etcd) durant l'exécution de l'application. Kubernetes permet également la connexion de stockage persistant. Par exemple, le fichier de spécification qui définit un pod peut requérir un volume persistant, afin que le pod de remplacement puisse se connecter à ce même volume en cas de défaillance du pod initial.
7. Exposer les services par liaison de port.	Kubernetes fournit des options de configuration permettant d'exposer les services sur des ports. Dans l'exemple de fichier YAML relatif à l'application nginx que nous avons présentée plus haut, le serveur Web nginx était lié au port 80 et exposé sous forme de service.
8. Assurer l'évolutivité horizontale par ajout de processus simultanés.	Pour faire évoluer une application, Kubernetes ajoute des pods supplémentaires. Kubernetes peut notamment utiliser le contrôleur de réplication pour ajouter simultanément plusieurs pods.
9. Garantir la durabilité au moyen de processus « jetables ».	Les conteneurs exécutés dans Kubernetes sont considérés comme mutables dans la mesure où vous pouvez les arrêter et les remplacer à la demande ou selon un calendrier déterminé.
10. Homogénéiser le développement et la production.	L'environnement Kubernetes permet de tester le code de développement et le code de test de manière rigoureuse et homogène. Par exemple, vous pouvez exploiter un déploiement Kubernetes à deux pods contenant, respectivement, l'environnement de production et l'environnement de préproduction, ce qui revient à créer des environnements homologues pour la préproduction et la production. De plus, l'environnement spécifié dans un conteneur est uniforme dans les environnements de développement et production.
11. Traiter les journaux et les flux d'événements.	Kubernetes vous permet d'accéder à la sortie standard d'un conteneur, que vous pouvez ainsi traiter sous forme de flux de données à l'aide de l'outil de votre choix, tel que VMware vRealize® Log Insight™.
12. Exécuter les tâches de gestion comme des processus ponctuels.	Vous pouvez planifier un pod constitué par le conteneur de l'application en utilisant un point d'entrée distinct pour l'exécution d'un processus différent, tel que le script de migration d'une base de données.

AVANTAGES POUR LES DÉVELOPPEURS

Au-delà des avantages stratégiques que les conteneurs et Kubernetes apportent à l'entreprise dans son ensemble et au bureau du directeur informatique, les développeurs apprécient les conteneurs car ils leur simplifient la vie, rendent le processus de développement plus intéressant et stimulent leur productivité.

- **Portabilité** : Les conteneurs permettent aux développeurs de choisir la méthode appropriée et le moment opportun pour déployer une application.
- **Rapidité** : Les conteneurs accélèrent certains workflows, tels que les tests, ainsi que les itérations.
- **Pipeline d'intégration et de déploiement continus** : Kubernetes et les conteneurs prennent en charge l'intégration et le déploiement continus.
- **Flexibilité** : Les développeurs peuvent élaborer le code sur leur ordinateur portable où et quand ils le souhaitent, en utilisant leurs outils de prédilection.
- **Le 13e facteur** : Les conteneurs et Kubernetes sont perçus comme des technologies porteuses, ce qui constitue une forte motivation pour les développeurs.

Exemple de cas d'usage

Voici une brève étude de cas illustrant dans ses grandes lignes un cas d'usage de gestion de conteneurs via Kubernetes.

Une société de taxi opérant dans une grande zone urbaine subit une érosion de sa clientèle au profit des services d'autopartage, dont la concurrence menace sa position autrefois solide sur le marché local. Il lui faut donc se transformer en une entreprise numérique capable de concurrencer les sociétés d'autopartage. Pour ce faire, la société doit développer sa propre application mobile, exécuter celle-ci de manière économique dans son Data Center d'envergure modeste et tenter de proposer des services innovants.

Cette société de taxi conserve à son actif un certain nombre d'avantages : une marque bien connue, établie de longue date à l'échelon local, et réputée pour ses chauffeurs ponctuels, courtois et expérimentés.

Tandis que des développeurs récemment recrutés travaillent sur l'application mobile, la société de taxi modernise son Data Center en misant sur du matériel standard et sur la virtualisation. Soucieuse de maximiser l'utilisation des ressources au sein de son petit Data Center, mais aussi de minimiser ses coûts, la société choisit d'exécuter sa nouvelle application dans des conteneurs Docker déployés sur des machines virtuelles. Elle prévoit de recourir à Kubernetes pour orchestrer l'application conteneurisée.

Après un lancement appuyé par une campagne de publicité sur et dans les véhicules, l'application connaît un succès immédiat. Afin de répondre aux fluctuations d'utilisation de l'application, la société a recours à Kubernetes pour moduler dynamiquement le nombre de conteneurs exécutant celle-ci. Par exemple, lorsque les mesures relatives à l'application atteignent le seuil indiquant un taux d'utilisation élevé, comme cela se produit typiquement aux heures de pointe, l'équipe DevOps fait appel à la fonctionnalité Kubernetes d'évolution horizontale automatique des pods pour maximiser automatiquement le nombre de conteneurs et permettre au système d'absorber la demande. À 4 heures du matin, en revanche, le nombre de conteneurs s'adapte de manière élastique à la baisse de la demande afin d'économiser les ressources.

L'application mobile établit une corrélation entre les demandes de course et l'emplacement. En conjuguant l'analyse de ces données avec sa connaissance historique approfondie des schémas propres à l'agglomération, la société de taxi peut positionner ses véhicules là où ils ont les meilleures chances d'être hélés par les clients. Elle parvient ainsi à préempter une partie des demandes de véhicule face à la concurrence. De plus, la société traite les journaux de l'application sous forme de flux d'événements, ce qui lui permet d'opérer dynamiquement de tels ajustements de jour comme de nuit, de façon à transférer ses véhicules vers les zones les plus actives.

Comme la société a implémenté son application via des conteneurs, les développeurs ont la possibilité de déployer les nouvelles modifications au jour le jour. La société peut s'appuyer sur les données collectées par l'application pour identifier de nouvelles fonctionnalités et innover rapidement de façon à tirer parti de ses points forts. Elle a notamment la possibilité de repérer les clients récurrents et de déployer un programme de fidélisation à leur intention.

Les avantages stratégiques découlant de l'agilité technique de la société, de ses applications conteneurisées et des capacités d'orchestration de Kubernetes se traduisent par un véritable avantage concurrentiel :

- Les règles de planification intégrées dans Kubernetes confèrent à la société la souplesse requise pour répondre à la demande de manière dynamique et rentable à partir d'un Data Center modeste mais désormais modernisé.
- Les défaillances et les pannes sont traitées automatiquement par Kubernetes, ce qui limite le nombre de demandes d'intervention adressées à sa petite équipe DevOps.
- La possibilité de modifier l'application et ses fonctionnalités de manière transparente permet à la société de supplanter certains rivaux plus puissants mais moins bien implantés à l'échelon local, dans la mesure où elle est plus agile et mieux à même d'appliquer sa connaissance des schémas locaux.

- Avec les conteneurs et Kubernetes, il devient plus simple et plus économique d'exécuter l'application.
- La facilité avec laquelle l'équipe DevOps team peut porter les conteneurs de l'environnement de test à l'environnement de production a pour effet d'accélérer le développement et le déploiement des nouvelles fonctionnalités.

Solutions VMware en matière de technologie de conteneurs

Dans un rapport récent intitulé « Closing the Digital Transformation Confidence Gap in 2017 », The Hackett Group rend compte d'une enquête menée auprès des cadres dirigeants de plus de 180 grandes entreprises. Ce rapport met en lumière un large écart de confiance entre les attentes élevées des personnes interrogées quant à l'impact stratégique de la transformation digitale et la piètre opinion qu'elles se font de la capacité de l'entreprise à mener à bien cette transformation digitale. Selon The Hackett group, ces résultats démontrent la nécessité pour les départements informatiques d'investir dans les outils requis et d'adopter des techniques de développement accéléré telles que les processus agiles⁸.

En vous aidant à accélérer et rentabiliser la mise en production des conteneurs, les solutions natives Cloud de VMware renforcent votre capacité à réaliser la transformation digitale.

L'exécution de conteneurs sur des machines virtuelles présente également l'intérêt d'offrir un niveau de sécurité supplémentaire pour les applications conteneurisées, en particulier dans le contexte de leur troisième pilier : les microservices. Selon un livre blanc Docker consacré à la sécurité, « le déploiement de conteneurs Docker en conjonction avec des machines virtuelles permet d'isoler un ensemble complet de services les uns des autres, puis de les regrouper sur un hôte de machine virtuelle »⁹.

Le fait de déployer les conteneurs sur des VM permet de protéger l'application par une double couche d'isolation, une approche qui convient parfaitement aux environnements de type Cloud conjuguant mutualisation et charges de travail multiples. « Les conteneurs Docker vont de pair avec les technologies de virtualisation dans la mesure où ils protègent la machine virtuelle elle-même et garantissent une protection approfondie pour l'hôte », conclut le livre blanc Docker sur la sécurité.

vSphere Integrated Containers

Solution de conteneurisation complète basée sur VMware vSphere, VMware vSphere Integrated Containers permet d'exécuter conjointement des charges de travail modernes ou traditionnelles au sein de votre Software-Defined Data Center VMware, en garantissant des fonctionnalités de réseau, de stockage et de sécurité, ainsi que des performances et une visibilité de niveau entreprise.

Compatible avec les conteneurs Docker, vSphere Integrated Containers vous donne les moyens d'utiliser immédiatement la technologie de conteneurisation pour améliorer la productivité des développeurs et l'agilité de l'entreprise. Cette solution vous aide à transformer votre organisation en entreprise numérique et à moderniser votre Data Center en déployant des applications conteneurisées.

⁸ Despite High Expectations for Digital Transformation Led by Cloud, Analytics, Robotic Process Automation, Cognitive & Mobile, IT & Other Business Services Areas See Low Capability to Execute, The Hackett Group, 16 mars 2017. Une version de cette étude est disponible en téléchargement sur inscription à l'adresse suivante : <http://www.thehackettgroup.com/research/2017/social-media/key17it/>.

⁹ Introduction to Container Security, livre blanc Docker, Docker.com.

Présentation de l'architecture

Le diagramme qui suit illustre l'architecture générale de vSphere Integrated Containers.

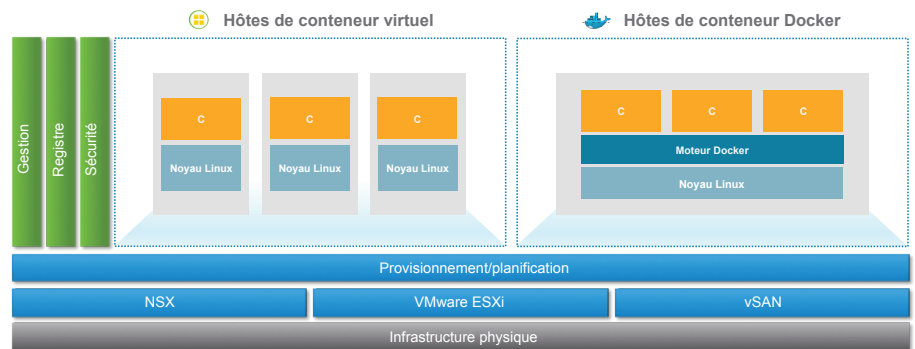


Figure 1 : Architecture générale de vSphere Integrated Containers

Cette architecture met à votre disposition deux modèles de déploiement de conteneurs :

- **Les hôtes de conteneur virtuel :** vSphere Integrated Containers s'appuie sur les structures natives de vSphere pour provisionner les conteneurs. En déployant chaque image de conteneur sous forme de machine virtuelle, vSphere Integrated Containers étend aux charges de travail conteneurisées les fonctions de disponibilité et de performances de vSphere, dont VMware HA, vMotion et Distributed Resource Scheduler (DRS). Les développeurs ont de surcroît la possibilité de consommer une API Docker.
- **Les hôtes de conteneur Docker :** Les développeurs peuvent provisionner les hôtes de conteneur Docker par eux-mêmes à la demande, et exécuter ceux-ci sur vSphere. L'environnement sans ticket fourni par vSphere Integrated Containers permet aux développeurs d'utiliser les outils Docker, et donne aux équipes informatiques les moyens d'assurer la gouvernance et le contrôle de l'infrastructure.

Composants

Chacun des composants de vSphere Integrated Containers est un projet Open Source :

- **Moteur vSphere Integrated Containers :** Ce moteur est un environnement d'exécution de conteneur pour vSphere, qui permet aux ingénieurs logiciels de développer des conteneurs et de déployer des applications conteneurisées aux côtés des charges de travail traditionnelles sur VM déployées sur les clusters vSphere.
- **Harbor :** Harbor est un registre de conteneurs privé de classe d'entreprise conçu pour stocker et distribuer les images de conteneur, qui ajoute à la distribution Open Source de Docker des fonctionnalités d'entreprise telles que la gestion des identités, le contrôle d'accès basé sur les rôles et l'audit.
- **Admiral :** Le portail de gestion de conteneurs Admiral fournit aux équipes DevOps et autres une interface utilisateur pour le provisionnement et la gestion des conteneurs. Admiral peut notamment afficher des mesures concernant les instances de conteneur. Les administrateurs de Cloud peuvent gérer les hôtes de conteneur et assurer la gouvernance de leur utilisation, notamment en appliquant des quotas de capacités.

LES AVANTAGES DES MICROSERVICES

Associés aux conteneurs, les microservices représentent de plus en plus souvent le modèle d'architecture privilégié pour le développement d'une nouvelle application. Cette architecture divise les fonctions d'une application en un ensemble de petits processus distincts, décentralisés et orientés sur l'objectif, chacun d'entre eux pouvant être développé, testé, déployé, remplacé et modulé de manière indépendante.

- Modularité accrue
- Développement et test d'applications facilités
- Développement parallélisé : chaque équipe peut développer et déployer un service indépendamment des équipes travaillant sur les autres services
- Prise en charge du remaniement continu du code, qui renforce progressivement les avantages des microservices
- Évolution vers un modèle d'intégration et de déploiement continus
- Évolutivité améliorée
- Mise à niveau des composants simplifiée

Fonctionnalités

vSphere Integrated Containers comprend un portail de gestion unifiée qui s'intègre avec la gestion des identités pour assurer un provisionnement sécurisé des conteneurs. Les développeurs et les équipes DevOps peuvent satisfaire leurs propres exigences en créant des hôtes de conteneur Docker à la demande.

Résultat : les équipes de développement d'applications ont désormais le moyen de générer, tester et déployer des applications conteneurisées. Cette solution prend en charge les pratiques de développement agile et les méthodologies DevOps telles que l'intégration et le déploiement continus.

Wavefront by VMware

Wavefront® by VMware assure une surveillance efficace des conteneurs à grande échelle. La plate-forme Wavefront comprend des tableaux de bord offrant aux équipes DevOps une visibilité en temps réel sur les opérations et les performances des applications conteneurisées et des clusters Kubernetes.

Le service Wavefront est capable de mesurer, de mettre en corrélation et d'analyser les données à l'échelle des conteneurs et des clusters Kubernetes. Le tableau de bord affiche les données relatives aux performances des microservices et à l'utilisation des ressources pour vous aider à identifier les problèmes et optimiser les applications. Ces données peuvent notamment vous permettre de décider de la méthode appropriée et du moment opportun pour faire évoluer un environnement de conteneurs. Pour plus d'informations, voir la page [VMware et Wavefront](#).

Pivotal Container Service

VMware® Pivotal Container Service offre aux entreprises un environnement Kubernetes de production fiable pour le déploiement, l'exécution et la mise en œuvre d'applications modernes ou traditionnelles sur les Clouds privés et publics. Basé sur le projet Open Source Kubo, Pivotal Container Service garantit la haute disponibilité, la sécurité avancée et l'efficacité opérationnelle dont les entreprises ont besoin pour accélérer leur processus de mise sur le marché, améliorer la productivité des développeurs et réduire leurs coûts d'exploitation.

Afin d'offrir une voie rapide vers la mise en production des microservices et des charges de travail conteneurisées, Pivotal Container Service établit une infrastructure de virtualisation et de conteneurs unifiée sur VMware vSphere ou au sein d'un Software-Defined Data Center VMware.

Composants

Pivotal Container Service comprend les composants suivants :

- **Système Kubernetes de niveau production.**
- **BOSH.** Il s'agit d'un système open source qui unifie l'ingénierie, le déploiement et la gestion du cycle de vie des releases pour les logiciels Cloud à petite ou grande échelle. Parfaitement adapté aux grands systèmes distribués, BOSH assure la surveillance, la récupération après défaillance et les mises à jour logicielles avec une interruption de service nulle ou minime. BOSH est compatible avec de nombreux fournisseurs d'infrastructure sous forme de service (IaaS), dont VMware vSphere, Google Cloud Platform, Amazon Elastic Compute Cloud (EC2) et OpenStack.

EN SAVOIR PLUS SUR...

Pour découvrir comment les solutions VMware peuvent vous aider à générer, exécuter et gérer des applications Cloud, rendez-vous sur : cloud.vmware.com/cloud-native-apps

- **VMware ESXi™**. Hyperviseur bare metal dédié leader du marché, VMware ESXi s'installe directement sur votre serveur physique, qu'il permet ainsi de partitionner en machines virtuelles.
- **VMware NSX®**. Cette technologie de virtualisation de réseau destinée aux architectures applicatives modernes fournit des fonctionnalités réseau aux clusters Kubernetes.

Conclusion

La génération et le déploiement d'applications conteneurisées sur une infrastructure VMware permettent de bénéficier des avantages stratégiques de la transformation digitale. Les solutions VMware améliorent la productivité des développeurs, l'agilité de l'entreprise, la flexibilité informatique et l'évolutivité des applications. Elles facilitent ainsi votre adaptation à l'évolution du marché, et vous aident à réduire le délai de mise sur le marché de vos applications.



VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com

VMware Global Inc. Tour Franklin 100-101 Terrasse Boieldieu 92042 Paris La Défense 8 Cedex France Tél. +33 1 47 62 79 00 www.vmware.com/fr

Copyright © 2017-2018 VMware, Inc. Tous droits réservés. Ce produit est protégé par les lois des États-Unis et internationales sur le copyright et la propriété intellectuelle. Les produits VMware et ceux de ses filiales sont couverts par un ou plusieurs brevets répertoriés à l'adresse <http://www.vmware.com/go/patents>. VMware est une marque déposée ou une marque commerciale de VMware, Inc. ou de ses filiales, aux États-Unis et/ou dans d'autres juridictions. Les autres marques et noms mentionnés sont des marques de leurs propriétaires respectifs. Référence : VMW_17Q3_WP_Driving-Digital-Transformation-with Kubernetes_FINAL2_081617

08/17