



Créer des applications modernes avec les conteneurs Linux

Le conteneur : une stratégie cloud-native légère qui redéfinit le développement d'applications traditionnel

Découvrez ses secrets

03 INTRODUCTION

04 RÉTROSPECTIVE

05 CHAPITRE 1: les bases

- 05 Un conteneur Linux, qu'est-ce que c'est ?
- 05 Fiabilité, rapidité et économies : les trois raisons d'adopter les conteneurs
- 06 Comparaison entre virtualisation et conteneurs
- 07 Conteneurs + Microservices = le duo gagnant
- 07 Réduisez, réutilisez et recyclez avec les images de conteneur

08 CHAPITRE 2 : améliorez votre productivité

- 08 Les conteneurs et vous
- 08 Soyez flexible
- 08 Voyez plus grand
- 08 Travaillez de manière plus intelligente
- 09 Faites de la standardisation une réalité
- 09 « Rédiger une fois, exécuter partout »
- 09 Fournissez des applications de qualité exceptionnelle
- 10 Utilisez vos outils et vos langages préférés
- 10 Devenez la perle rare

11 CHAPITRE 3 : des conteneurs dans la nature

- 11 Approche lift-and-shift
- 11 Réusinage de code
- 12 Développement de nouvelles applications
- 12 Microservices
- 12 Applications hybrides
- 12 Fonctions et tâches répétitives
- 12 Intelligence artificielle et apprentissage automatique

13 CHAPITRE 4 : réflexions et défis

- 13 Avant de se lancer...
- 13 Définissez votre stratégie de données
- 13 Établissez la communication entre vos conteneurs
- 13 Synchronisez et standardisez
- 14 Collectez tous les journaux
- 14 Renforcez la sécurité
- 14 Défis
- 14 Prenez une longueur d'avance sur la technologie à venir
- 14 Adoptez la culture DevOps
- 14 Restez au top de la sécurité
- 14 Gestion et surveillance

15 CONCLUSIONS

- 15 En savoir plus
- 15 Ressources supplémentaires

Introduction

Que contient un conteneur ?

Le temps du développement d'applications monolithiques est révolu, et, bien que l'expression « transformation numérique » sonne comme un terme à la mode, à l'instar de blockchain, agile et cloud, ses avantages sont bien réels. Cette transformation s'accompagne de niveaux de vitesse, de cohérence et d'efficacité sans précédent, qui modifient fondamentalement la manière de travailler des développeurs.

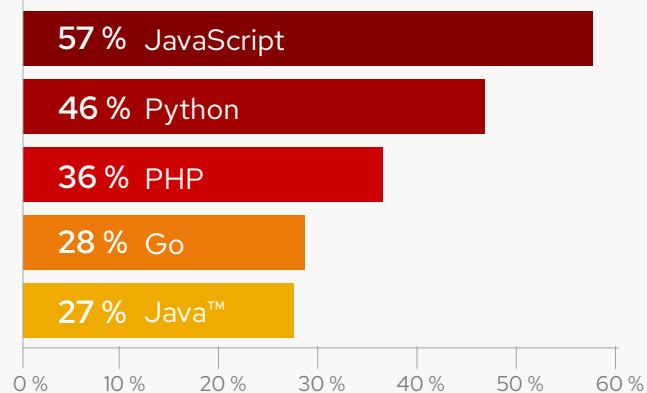
Oui, cela vous concerne aussi.

Lorsque le service commercial fait des promesses, c'est bien souvent aux développeurs de les assumer et de les concrétiser. Dans un monde où les utilisateurs demandent de nouvelles applications, fonctionnalités et mises à jour sur tous leurs appareils en temps réel, la technologie des conteneurs est votre arme secrète. Les conteneurs vous permettent de travailler plus intelligemment en créant des environnements de développement cohérents dans lesquels vous pouvez développer et fournir rapidement des applications qui s'exécutent partout. Avec les conteneurs, vous pouvez également fournir des microservices qui éliminent les cycles de test de régression laborieux, permettent des déploiements sans perturbation et fournissent un mécanisme pour l'application de correctifs ou la restauration du code, fonctionnalité par fonctionnalité.

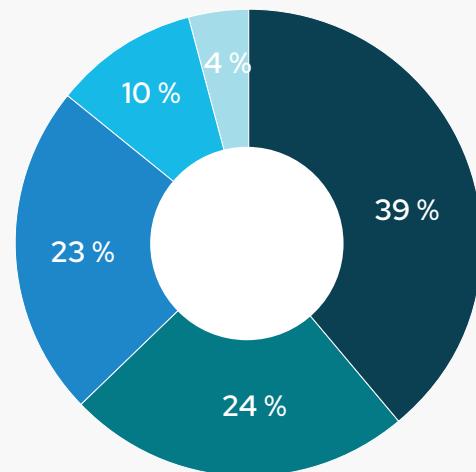
Apprenez à déployer des applications conteneurisées. S'inscrire au cours de formation ➤

Bien que les conteneurs représentent une technologie relativement nouvelle, des développeurs du monde entier les adoptent et les considèrent comme des outils précieux et essentiels au développement moderne. La maturation rapide des conteneurs et les progrès en matière d'orchestration et d'outils de gestion ont permis d'accélérer leur adoption, selon un rapport de Digital Ocean en 2018 qui stipule que 49 % des développeurs les utilisent.¹

LANGAGES UTILISÉS POUR DÉVELOPPER AVEC LES CONTENEURS¹



AVANTAGES DES CONTENEURS¹



- Évolutivité en toute simplicité
- Tests simplifiés
- Tests accélérés
- Aucune dépendance vis-à-vis d'un fournisseur
- Autres

Rétrospective

Un progrès n'arrive jamais seul

« *L'invention de la presse à imprimer de Johannes Gutenberg a fait bondir la demande pour des paires de lunettes, puisque la lecture, étant devenue une pratique courante, a fait réaliser à de nombreux Européens qu'ils voyaient mal. Cette demande a encouragé un nombre croissant de personnes à produire et expérimenter de nouveaux types de verres, ce qui a conduit à l'invention du microscope, qui a son tour nous a permis d'apprendre que le corps humain est composé de cellules microscopiques.* ».

Steven Johnson

How We Got to Now: Six Innovations That Made the Modern World

À l'instar de nombreux autres progrès technologiques, les conteneurs représentent l'aboutissement de plusieurs concepts et technologies qui ont évolué avec le temps. Dans les années 1970-80, nous avons commencé à diviser le code en objets en jouant avec les idées d'abstraction et d'isolation. Nous avons rapidement appris que le fait de sécuriser des parties de notre code et d'en exposer d'autres nous offrait un meilleur contrôle sur le traitement et la manipulation des données, tout en nous donnant la flexibilité d'intégrer des systèmes adjacents. Ces avancées ont permis de séparer et de diviser davantage les processus et les composants. Nous avons alors connu les environnements multicouches et les architectures orientées services (SOA), qui permettaient d'isoler davantage les couches de données des codes métier et des interfaces utilisateur. Pendant ce temps, nos méthodologies ont également évolué, passant du développement monolithique en cascade au cycle de vie de développement logiciel (SDLC), puis aux méthodes de développement agile et scrum, et enfin, à la méthode DevOps et à la distribution continue.

Du point de vue de l'entreprise, toutes ces évolutions ont permis d'alléger le processus de développement et ainsi de produire plus rapidement du code plus économique et plus fiable. Pour les développeurs, chaque nouvelle itération d'innovation a permis de réduire le cycle de développement, mais aussi de garantir un respect plus rigoureux des schémas et des méthodologies. Si chaque évolution représente un pas dans la bonne direction, l'apparition des conteneurs a permis de tout centraliser et de fournir une solution pour un code véritablement flexible, interopérable et portable.

En se penchant sur l'évolution des conteneurs Linux®, il est facile de voir l'impact qu'ils ont eu au cours de ces 15-20 dernières années.

2000

La technologie des conteneurs est apparue sous la forme des FreeBSD Jails, qui permettaient de partitionner les serveurs en sous-systèmes, où un développeur pouvait travailler sans compromettre le système entier.

2001

Le concept de conteneur est apparu sous Linux à travers le projet Linux-VServer avec l'objectif d'exécuter plusieurs serveurs Linux sur un seul boîtier.

2007

Des technologies supplémentaires sont combinées pour faire de la conteneurisation une réalité. Plus particulièrement, cgroups, systemd et les espaces de nom kernel/user ont ajouté des capacités de contrôle global et de virtualisation qui ont servi de structure pour la séparation des environnements.

2008

Docker entre en scène avec sa technologie de conteneurs, ajoutant encore plus de concepts et d'outils pour permettre aux utilisateurs de créer rapidement de nouveaux conteneurs à couches et de les partager.

2012

L'architecture de microservices a évolué en tant que spécialisation et amélioration de la SOA, utilisée pour créer des logiciels flexibles, indépendants et déployables.⁴

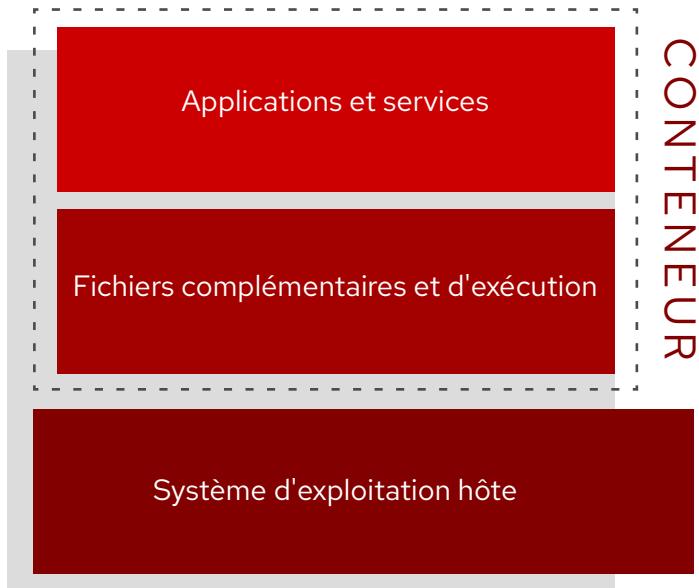
2015

Kubernetes, un système d'orchestration de conteneurs Open Source, a été lancé pour automatiser le déploiement, l'évolution et la gestion des applications.

Chapitre 1 : les bases

Un conteneur Linux, qu'est-ce que c'est ?

Un conteneur désigne simplement une nouvelle manière d'extraire un ou plusieurs processus du reste d'un système. Pour ainsi dire, les conteneurs réduisent la charge et vous permettent de travailler sur de petits sous-ensembles de code sans incidence sur l'environnement d'exécution. Ils fournissent également une méthode standard pour mettre en paquet et isoler le code, les configurations et les dépendances d'une application dans un seul objet.



Fiabilité, rapidité et économies : les trois raisons d'adopter les conteneurs

La véritable valeur des conteneurs réside dans leur portabilité. En rassemblant tous les fichiers nécessaires pour exécuter une application, une fonctionnalité ou une composante dans une seule image distincte, les conteneurs Linux fournissent de la cohérence et de la prédictibilité lorsqu'ils passent du développement aux tests et enfin en production. Cela rend le développement des conteneurs plus rapide, plus fiable et moins onéreux que les pipelines de développement monolithique, qui reposent sur des environnements de développement, de test et de production répliqués. De plus, une fois que vous avez rédigé et développé votre code conteneurisé, vous pouvez le déployer sur de nombreux environnements d'exploitation sans temps de développement supplémentaire, cycles de test chronophages ou processus de déploiement spécifiques.

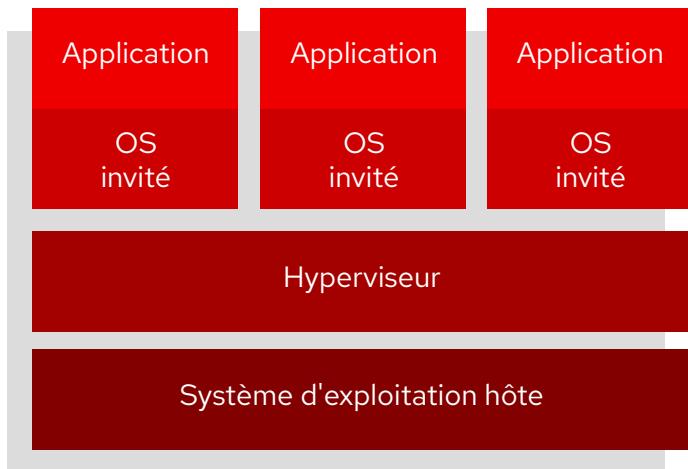
Les conteneurs partagent un système d'exploitation (OS) installé sur le serveur et s'exécutent en tant que processus séparés des ressources, ce qui permet des déploiements rapides, fiables et cohérents, quel que soit l'environnement.

La véritable valeur des conteneurs réside dans leur **portabilité**.

Comparaison entre virtualisation et conteneurs

Bien que les conteneurs et la virtualisation semblent similaires, ils sont en réalité très différents.

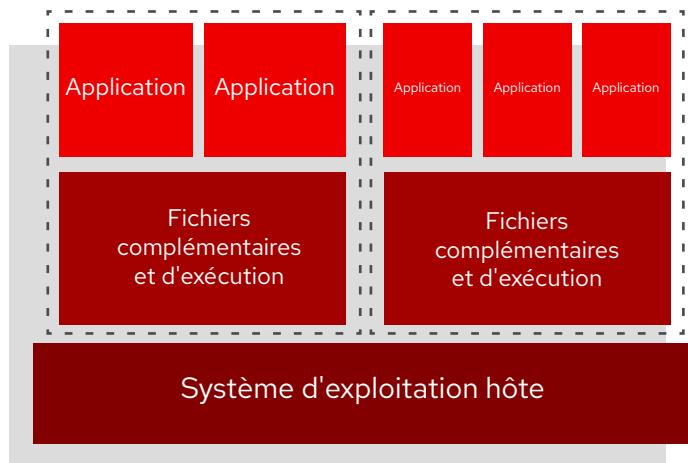
VIRTUALISATION



La virtualisation vous permet d'exécuter plusieurs ordinateurs séparés sur un seul équipement. Les systèmes d'exploitation et leurs applications partagent des ressources matérielles à partir d'un unique serveur hôte. Chaque machine virtuelle nécessite son propre système d'exploitation, et c'est un hyperviseur qui crée et exécute les machines virtuelles.

OU

CONTENEURS



Un conteneur isole les processus de l'application du reste du système et n'inclut que ce qui est nécessaire à l'exécution d'une application spécifique, y compris certains fichiers de système d'exploitation, les programmes et bibliothèques requises, ainsi que les ressources système. Les conteneurs sont légers, démarrent bien plus rapidement que les machines virtuelles et utilisent une fraction seulement de leur mémoire.

Conteneurs + Microservices = le duo gagnant

Maintenant que vous avez une idée générale de ce qu'est un conteneur, découvrons **sa fonction la plus importante pour les développeurs : le développement de microservices.**

Les microservices sont de petites applications, autonomes et à fonction unique, qui communiquent à travers des interfaces de programmations d'application (API). L'un des principes fondamentaux d'une architecture de microservice est que chaque microservice gère une seule et unique fonction et fournit une API bien définie qui permet la communication vers et à partir du code. Les microservices représentent le mécanisme d'encapsulation ultime. Puisque les microservices sont entièrement autonomes, une modification sur un microservice comprend moins de risque pour l'application qu'une modification du code dans une structure monolithique. De plus, les microservices sont plus rapides et agiles que les applications traditionnelles, en raison de leur nature autonome et de leur utilisation indépendante des ressources système.

Les conteneurs et les microservices peuvent exister indépendamment, c'est d'ailleurs souvent le cas. Individuellement, leur fonction est différente. Lorsqu'ils sont implémentés ensemble, ils forment un puissant outil de création d'applications portables et cloud-native.

Les conteneurs sont en quelque sorte la technologie sur laquelle reposent les microservices. Les conteneurs sont extraits du système d'exploitation hôte et contiennent tous les fichiers complémentaires et d'exécution nécessaires pour exécuter le code qu'ils contiennent. Lorsque vous déployez un conteneur, ce dernier pourra s'exécuter quel que soit le système d'exploitation sous-jacent. En outre, les conteneurs sont portables, vous pouvez donc les déployer sur de nombreux clouds et appareils, sans reconstruction ni tests supplémentaires.

Le développement de microservices avec déploiement conteneurisé devient la norme pour le développement en entreprise. Cette architecture offre des niveaux sans précédent d'agilité, de vitesse et d'efficacité en matière de ressources pour de nombreuses tâches sur lesquelles les développeurs comme vous travaillent quotidiennement. Dans un environnement DevOps, les conteneurs et microservices permettent aux développeurs de déployer chaque service de manière indépendante. Cette pratique élimine le besoin de fusionner les modifications du code, améliore grandement les tests et aide à localiser les failles en phase de test et de production. De même, plusieurs équipes de développeurs peuvent travailler en parallèle sur des applications faiblement couplées et choisir la pile technologique la plus adaptée à leurs exigences, sans les imposer aux autres équipes.

Réduisez, réutilisez et recyclez avec les images de conteneur

Les développeurs apprécieront toujours de pouvoir réutiliser leur code. Avec les conteneurs, vous pouvez créer des images de conteneurs de base, les ajouter à un référentiel et les récupérer dès que vous êtes prêt à démarrer un nouveau projet. Une image de conteneur de base est un fichier non modifiable et statique, qui n'inclut aucun exécutable. Il est à la fois cohérent et portable et peut exécuter un processus isolé sur n'importe quelle infrastructure. Une image est composée de bibliothèques système, d'outils système et d'autres paramètres de plateforme nécessaires à l'exécution de vos applications.

Vous pouvez créer vos propres images de conteneur ou vous pouvez les choisir à partir de référentiels publics. De nombreux éditeurs de logiciels, y compris Red Hat et Microsoft, créent des images de leurs produits qu'ils mettent à disposition du public.

0001100101000110110001001001010
010010001111001001001010101000
I0101100
0010001
0001000
0100001
1100100
0100100
1101000
I0101100
0010001
0001000
0100001
1100100

Les microservices et les conteneurs forment une combinaison puissante, particulièrement lors de leur intégration au sein d'un environnement DevOps.

IMAGES DES 5 PRINCIPAUX AVANTAGES DES CONTENUEURS

1

Automatisation des processus de création et de déploiement

2

Des balises pour faciliter la location et le téléchargement

3

Simplification de l'analyse des vulnérabilités

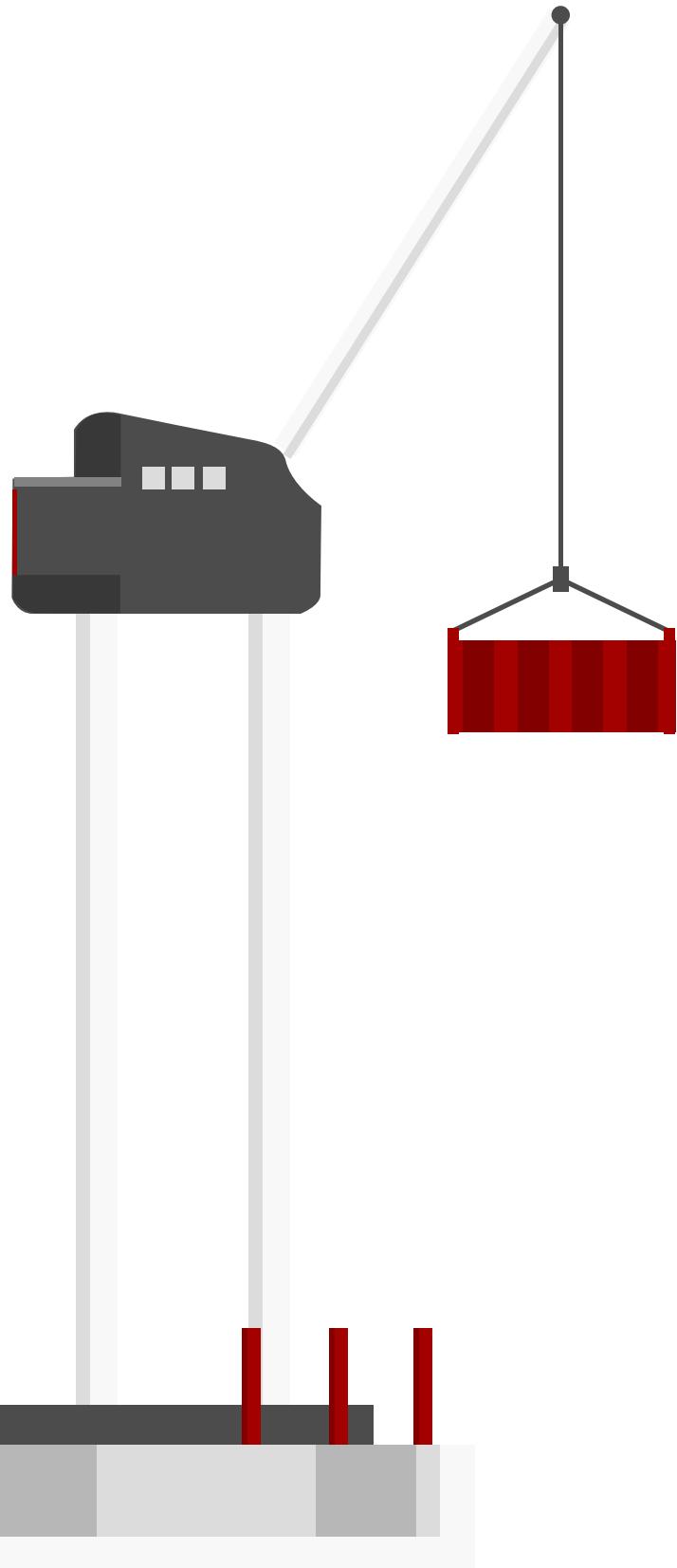
4

Renforcement des normes et politiques de rédaction du code

5

Gain de temps et réutilisation

Chapitre 2 : améliorez votre productivité



Les conteneurs et vous

Les conteneurs fournissent une approche simplifiée pour créer, tester, déployer et redéployer les applications, ce qui en fait un excellent choix aussi bien pour les projets simples que pour les applications critiques. Les conteneurs offrent de multiples avantages dans tous les domaines de l'informatique et de l'entreprise, mais prenons un moment pour découvrir comment ils peuvent améliorer votre réussite en tant que développeur, au quotidien.

Soyez flexible

Les cycles de développement sont toujours trop courts. Le service commercial suit sa feuille de route, votre équipe de test a une liste de problèmes en attente, les équipes d'exploitation luttent pour garder les systèmes à jour et sécurisés, et vos clients ont leur propre idée de ce qui est important. Malheureusement, une journée n'est pas extensible et vous n'avez qu'un nombre limité de développeurs pour travailler sur un projet. Lorsque vos applications fonctionnent sur une base de conteneurs, cela vous permet à vous, aux commerciaux et aux équipes d'exploitation d'agir en temps réel sur les journaux de modifications, les problèmes, les questions de sécurité et les niveaux de correctifs tout en fournissant aux clients les nouvelles fonctionnalités qu'ils demandent. En outre, puisque vous pouvez déployer un seul conteneur sans incidence sur le reste de votre architecture d'application, vous pouvez effectuer des modifications à la volée, sans attendre que les autres équipes finissent leur cycle de développement.

Voyez plus grand

Les conteneurs sont légers et peuvent souvent démarrer en quelques millisecondes. Ils ne nécessitent pas de démarrer un système d'exploitation et ne chargent que les dépendances nécessaires. De plus, il est possible de créer, répliquer et détruire des conteneurs en quelques secondes. Lorsque vos clients créent des charges saisonnières ou que l'entreprise décide d'ajouter de nouvelles équipes d'utilisateurs, votre équipe d'exploitation peut réagir rapidement en ajoutant des ressources que vos conteneurs utiliseront naturellement si nécessaire. De plus, vous pouvez étendre les applications conteneurisées à de nouveaux utilisateurs, même à l'international, en déployant simplement de nouveaux clouds.

Travaillez de manière plus intelligente

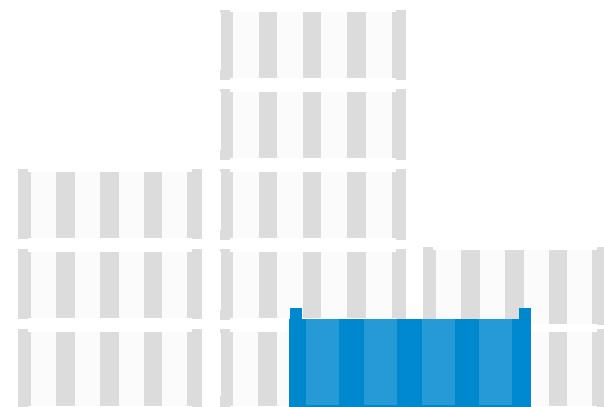
Les conteneurs vous permettent de vous concentrer sur la logique de vos applications sans vous soucier des différentes versions du système d'exploitation et configurations d'application. Après tout, c'est à cela que servent vos équipes d'exploitation. De plus, les conteneurs vous permettent de mettre votre code, vos dépendances et vos configurations en paquet, dans une capsule unique, facile à contrôler, tester et déployer sur différentes versions. En associant les conteneurs à une architecture basée sur les services, vous pouvez également simplifier la prise en charge, les tests et l'amélioration de vos applications.

Faites de la standardisation une réalité

Vous êtes nombreux à penser que la standardisation des environnements du développement à la production n'est qu'une chimère. Alors préparez-vous à être surpris. L'un des principaux avantages des conteneurs est qu'ils permettent de standardiser les environnements locaux, de développement, de test, de contrôle qualité et de production. Avec un tel niveau de prévisibilité, vous serez capable de mettre en place des environnements isolés et passerez moins de temps sur le débogage et le diagnostic des problèmes causés par les différences dans les niveaux de correctifs, les systèmes d'exploitation et les applications. Ainsi, vous pourrez consacrer plus de temps au développement et à la distribution de nouvelles fonctionnalités. De plus, si votre équipe accueille de nouveaux développeurs, ceux-ci pourront commencer à travailler plus rapidement sans perdre de temps à installer et configurer leur propre environnement de développement local. Ils pourront extraire une image de conteneur à partir d'un référentiel et se mettre immédiatement à rédiger du code.

« Rédiger une fois, exécuter partout »

L'expression « rédiger une fois, exécuter partout » s'accompagne historiquement de quelques réserves, la principale étant le système d'exploitation sur lequel vous avez développé votre application. Vous pouvez exécuter votre application partout, à condition que le système cible soit doté du même système d'exploitation. Avec les conteneurs, votre code peut s'exécuter dans quasiment n'importe quel environnement, peu importe où il a été développé. Lorsque vous développez dans un conteneur, vous pouvez déployer votre application sous Linux, Windows ou Mac, sur un système nu, une machine virtuelle, un cloud public ou privé et même un environnement hybride. De même, l'adoption généralisée du projet Open Source Docker est une façon stable d'automatiser le déploiement d'applications dans les conteneurs, et ainsi de libérer du temps pour passer à votre prochain cycle de développement.



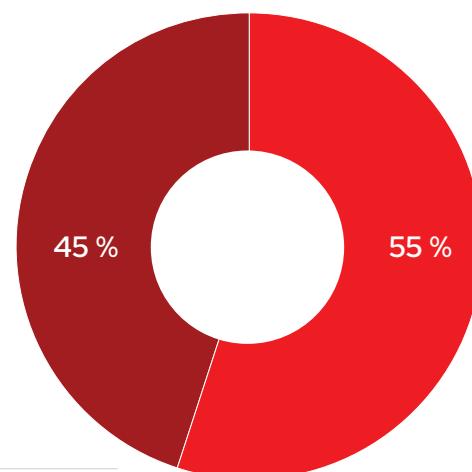
Fournissez des applications de qualité exceptionnelle

Les conteneurs permettent d'accélérer et de simplifier les tests et la résolution des problèmes pour tout le monde. Pour vous, la capacité d'isoler le développement de chaque fonction applicative permet de réduire les risques d'introduire des erreurs dans le reste du code. Une fois que votre conteneur passe en phase de contrôle de version, les équipes de contrôle qualité peuvent effectuer les tests directement sur votre image de conteneur, au lieu d'extraire et de développer l'application entière à partir d'un serveur d'intégration continue (CI), ce qui simplifie et accélère les tests.

Les conteneurs peuvent également aider les équipes de contrôle qualité et d'assistance à identifier plus rapidement l'origine d'un problème dans l'application. Puisque le code est faiblement couplé et basé sur des API, il y a moins de dépendances à vérifier lorsqu'un évènement inattendu survient. Les ingénieurs peuvent rapidement identifier la source du problème et cibler le conteneur à l'origine de l'erreur. Une fois le problème identifié, les conteneurs permettent d'annuler les changements facilement. Chaque conteneur peut faire l'objet d'une restauration sans incidence sur le reste de l'application. Ainsi, lorsque vous devez retravailler sur un conteneur, vous passez moins de temps à chercher et corriger le problème et davantage à développer les fonctionnalités que vous réclamez vos responsables métier et vos clients. Grâce aux conteneurs, vous pouvez corriger les problèmes le jour même de leur apparition, ce qui ne manquera pas de plaire à vos clients et à votre responsable.

Selon une récente étude d'IDC,

55 % des responsables informatiques ont déployé des conteneurs sur site, alors que 45 % les ont déployés dans un cloud public.⁵



Utilisez vos outils et vos langages préférés

En tant que développeur, vous savez à quel point il est agréable de pouvoir choisir les outils et les langages que vous voulez utiliser sur un projet. Les conteneurs vous offrent cette flexibilité. Certains environnements d'exécution d'application sont mieux adaptés à certaines charges de travail et architectures. Par exemple, vert.x améliore la réactivité des architectures distribuées, ce qui représente un avantage pour les applications réactives, en temps réel, telles que celles sur lesquelles s'appuient les appareils IoT (Internet des objets). Vous pourriez parfaitement développer ce type d'application dans un autre langage, mais vous devriez alors réinventer ce que vert.x fournit nativement. Puisque les conteneurs sont véritablement indépendants, ils vous offrent la possibilité et la flexibilité de choisir l'outil le plus adapté à une application, sans forcer toute votre équipe à adopter les mêmes outils ou langages pour leurs propres projets.

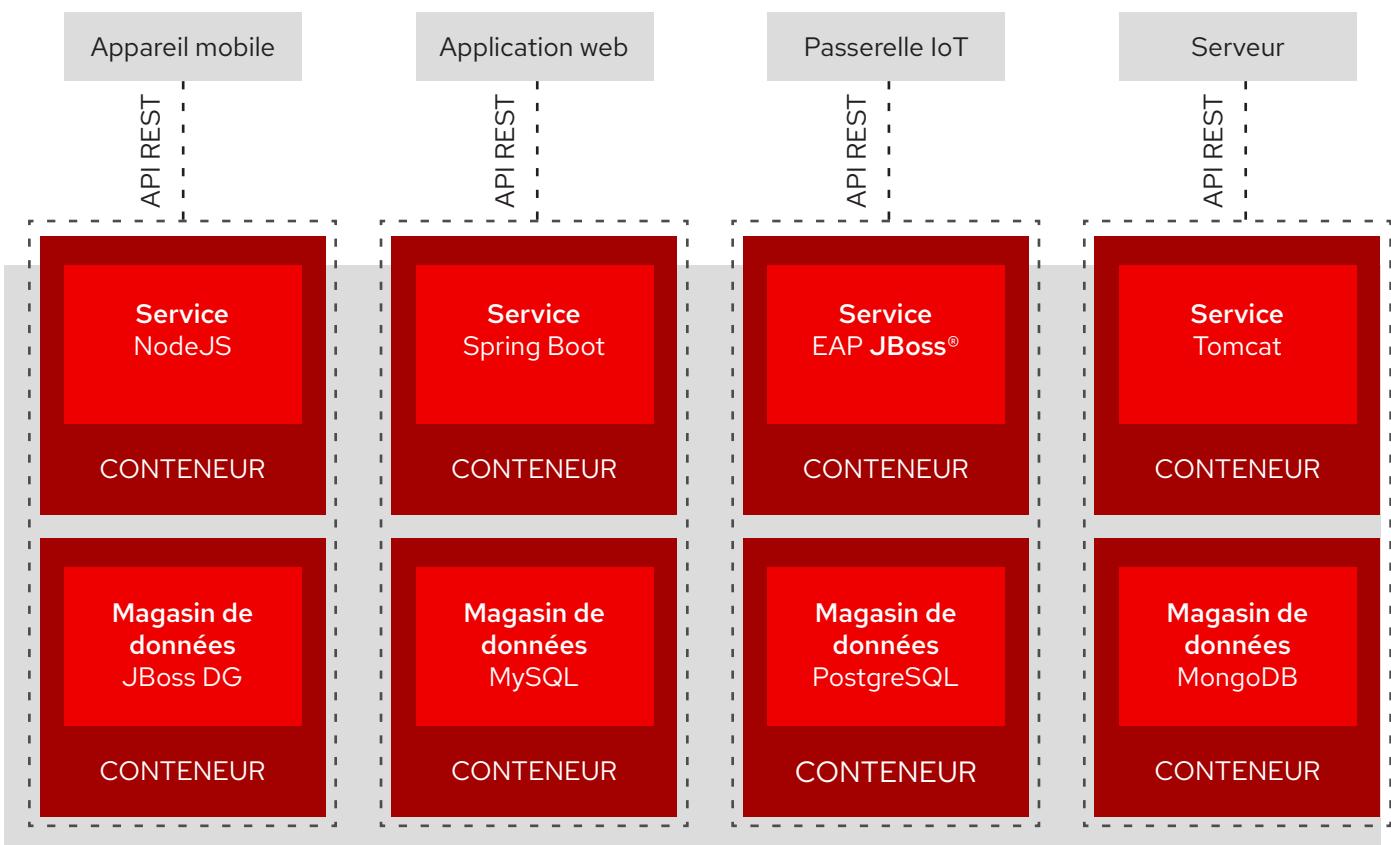


Figure 1 : améliorer la flexibilité du développement et simplifier le déploiement avec les conteneurs

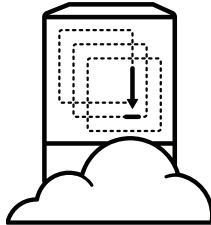
Devenez la perle rare

Abordons sérieusement ce dernier point. Au cours de votre carrière, si vous ne faites pas évoluer vos compétences au rythme des nouvelles technologies, vous risquez de vous retrouver à la traîne, ou pire, de travailler sur d'anciennes applications jusqu'à la fin de votre carrière. Les opportunités de croissance et de développement sont essentielles à votre réussite. De plus, la plupart des développeurs aiment découvrir de nouvelles technologies. C'est l'un des avantages de ce travail.

Les microservices, conteneurs et solutions d'orchestration des conteneurs présentent la double qualité d'être à la fois des défis techniques et des technologies très demandées. D'après un sondage de Red Hat® en 2018, l'utilisation des conteneurs est amenée à croître de 89 % au cours des deux prochaines années et leur adoption connaîtra une forte accélération à l'avenir.⁶ Ainsi, si vous maîtrisez le développement des microservices conteneurisés, vous vous donnez de nouvelles opportunités de développer votre carrière et de fournir de la valeur à votre entreprise.

Chapitre 3 : des conteneurs dans la nature

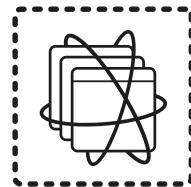
Si ce n'est pas déjà le cas, on vous demandera sûrement de fournir votre code dans des conteneurs dans un futur proche. Tandis que les entreprises passent par la transformation numérique, les équipes de direction découvrent comment les conteneurs permettent aux services informatiques de mieux utiliser les ressources, humaines comme informatiques, afin de réduire les coûts, améliorer l'efficacité et fournir de la valeur. Lorsque vous planifiez vos projets, gardez ces cas d'utilisation à l'esprit et abordez le sujet des conteneurs lors de votre prochaine réunion d'équipe.



Approche lift-and-shift

Même si les applications cloud-native font beaucoup parler d'elles, la plupart des applications d'entreprises sont encore de type monolithique. L'expression « lift-and-shift » fait référence à la migration des applications existantes vers une architecture cloud-native plus moderne. Les applications existantes sont simplement transférées dans le cloud avec le moins de modifications possible au niveau du code. Ce processus rapide plaît généralement aux responsables d'entreprise, car il permet d'utiliser les applications critiques existantes pour fournir une plus grande valeur et de meilleures performances sans investir massivement pour tout moderniser.

Bien que la méthode du lift-and-shift soit simple et rapide, elle est souvent considérée comme une solution temporaire, car elle n'exploite pas entièrement les avantages des outils spécifiques au cloud. Si vous utilisez cette méthode pour migrer l'une de vos applications, vous serez amené à réévaluer cette solution pour vous assurer qu'elle fournit toujours de la valeur et vous devrez recommander d'autres solutions comme le réusinage ou le remplacement du code.



Réusinage de code

Le réusinage du code est à la base de la modernisation des applications depuis que la première application est devenue obsolète. Dans notre cas, nous parlerons spécifiquement du réusinage d'une application pour les conteneurs. Bien que le réusinage soit plus complexe qu'une migration de type lift-and-shift, il permet à l'application de profiter pleinement des avantages des environnements conteneurisés et cloud-native.

Par le passé, les efforts de réusinage, jugés trop longs et complexes, étaient souvent laissés de côté. Avec les conteneurs, le réusinage devient un processus progressif. Pour commencer, vous modifiez uniquement les éléments les plus critiques d'une application et vous utilisez la méthode du lift-and-shift pour le reste. Ensuite, vous pouvez réusinage le reste au fil du temps. Cela vous permet de maintenir l'activité tout en effectuant des améliorations supplémentaires quand votre agenda le permet. Vous verrez souvent apparaître le processus « strangling the monolith » (littéralement « tordre le cou du monolithe ») dans les publications et les blogs liés aux microservices.

Bien sûr, il est toujours possible de réusinage à l'ancienne, en modifiant l'application entière avant de la publier. Ces deux stratégies ont toutes les deux des avantages et des inconvénients. Le principal avantage du réusinage progressif est le temps. Son inconvénient réside dans sa tendance à ne jamais migrer entièrement vers une plateforme cloud-native, ce qui génère de la complexité et des risques au niveau de la gestion.

Développement de nouvelles applications

Comme nous l'avons vu, les conteneurs sont des outils puissants et flexibles pour développer de nouvelles applications cloud-native. Le développement à partir d'une page vide vous permet de libérer le plein potentiel des conteneurs. Les conteneurs fournissent une plateforme idéale pour les microservices, les applications hybrides, l'automatisation des tâches et fonctions répétitives, ainsi que les applications avant-gardistes telles que l'intelligence artificielle (IA) et l'apprentissage automatique.

Microservices

Les conteneurs sont étroitement liés aux architectures de microservices. Vous pouvez facilement isoler, déployer et modifier les microservices et les applications distribuées en utilisant des conteneurs individuels comme blocs de construction.

Applications hybrides

Les conteneurs vous permettent de standardiser le déploiement du code et ainsi de créer des workflows pour les applications qui sont exécutées à la fois sur des environnements cloud et sur site.

Fonctions et tâches répétitives

Les tâches répétitives, telles que le traitement par lot et les tâches d'extraction, de transformation, de chargement, peuvent être facilement développées dans des conteneurs pour être démarrées rapidement. Elles peuvent ensuite être automatisées pour simplifier l'exploitation et évoluer dynamiquement en fonction de la demande.

Intelligence artificielle et apprentissage automatique

Les conteneurs offrent une nouvelle manière de créer et de déployer des applications cloud portables qui intègrent l'intelligence artificielle (IA) et l'apprentissage automatique (AA). Ces conteneurs permettent de faire évoluer rapidement les modèles d'IA et d'AA pour s'adapter aux besoins en traitement des algorithmes d'apprentissage avancés. De plus, les conteneurs d'IA et d'AA peuvent être déployés à proximité des sources de données afin d'améliorer les performances et de réduire les cycles d'apprentissage.

Chapitre 4 : réflexions et défis

Simplifiez-vous la vie, ne sautez pas ce chapitre

Nous avons beaucoup parlé de l'influence et de la puissance des conteneurs, mais comme toute autre technologie, ils posent certains défis à ne pas négliger. À mesure que le nombre de conteneurs déployés en production augmente, ces difficultés peuvent devenir critiques si elles ne sont pas résolues. Vous devrez garder un œil sur l'évolution continue de la technologie et sur la façon dont votre entreprise adopte, gère et entretient son architecture basée sur des conteneurs.

Avant de se lancer...

Vous avez probablement déjà quelques idées en tête sur la manière de lancer un nouveau projet ou de réusiner une application existante. Il n'y a pas de meilleure manière d'acquérir de l'expérience et des connaissances sur les conteneurs que d'entrer dans le vif du sujet et de commencer à coder.

Cependant, il convient de prendre en compte quelques éléments avant de se lancer. La création d'applications conteneurisées pose un certain nombre de défis qui n'existent pas dans les environnements traditionnels. Voici un aperçu des principales considérations qui peuvent affecter vos efforts de développement. De nombreuses idées présentées dans cette section sont inspirées du [Project Atomic](#). N'hésitez pas à le consulter quand vous aurez terminé ce livre numérique.

Définissez votre stratégie de données

Si la plupart des applications cloud-native sont sans état, de nombreuses applications requièrent un stockage permanent des données. Les conteneurs impliquent un stockage immuable ; vos données seront donc perdues une fois le conteneur désactivé. Gardez cela à l'esprit et pensez à concevoir vos applications de manière à ce que les données persistent, quel que soit l'état du conteneur.

S'il est nécessaire de préserver les données de votre application une fois le conteneur désactivé, vous pouvez attribuer un volume de stockage à votre conteneur. Ces volumes de stockage persisteront, quel que soit l'état du conteneur. Les développeurs devraient s'assurer de concevoir leurs applications pour l'écriture sur un magasin de données partagé. Pour les applications d'entreprise, des outils tels que

la solution Red Hat OpenShift® Container Storage fournissent un système de stockage logiciel spécialement conçu pour les environnements de conteneurs. En plus d'attribuer un espace de stockage permanent à vos données, même lors de l'activation et de l'arrêt des conteneurs, la solution OpenShift® Container Storage autorise le développement cloud, sur conteneurs et sur systèmes nus, afin d'améliorer la portabilité de vos conteneurs sans les restreindre à une architecture de stockage.

Établissez la communication entre vos conteneurs

Les composantes des applications distribuées doivent communiquer entre elles pour effectuer des workflows. Les technologies des conteneurs encouragent les développeurs à créer des points d'interconnexion explicites et à fournir un mécanisme de communication entre les conteneurs basés sur des API. C'est parfait pour la communication entre les conteneurs, mais qu'en est-il des bases de données ?

Les bases de données traditionnelles communiquent généralement à l'aide d'un socket à travers un réseau. Étant donné que l'espace de nom d'un conteneur change en fonction de son état, ce type de communication ne peut pas fonctionner. Dans une application conteneurisée, vous avez besoin d'une plateforme d'orchestration de conteneurs, telle que Kubernetes ou Red Hat OpenShift Container Platform, afin de faciliter la communication réseau entre vos conteneurs, vos bases de données et d'autres ressources réseau.

Synchronisez et standardisez

Certaines applications conteneurisées demandent la synchronisation du conteneur et de l'hôte au niveau de certains attributs, pour assurer un comportement uniforme. Par exemple, imaginons un serveur de journaux centralisé qui reçoit des données à partir de plusieurs conteneurs déployés dans différentes régions du monde. Les horodatages et les informations des journaux seraient pratiquement inutiles pour l'équipe d'exploitation si chaque conteneur donnait une heure différente de celle de l'hôte sans indiquer l'emplacement du serveur. En synchronisant et en standardisant un ensemble d'attributs liés à l'environnement, vous pouvez garantir que les données communiquées au magasin de données central sont précises, pertinentes et utilisables.

Collectez tous les journaux

Chaque application devrait enregistrer les informations nécessaires pour simplifier la résolution des problèmes. Si votre application enregistre des actions, des erreurs et des avertissements avec un mécanisme de journalisation, il vous faudra potentiellement permettre aux utilisateurs d'obtenir, de consulter et de conserver ces journaux. Étant donné que vos conteneurs se trouvent sur des espaces de nom différents et ne peuvent pas accéder directement aux composants nus, y compris les disques durs locaux, vous devrez repenser votre stratégie de connexion.

La manière la plus simple de recueillir les journaux est d'utiliser un outil conçu spécialement pour cela. Les plateformes d'orchestration de conteneurs comme OpenShift Container Storage collectent automatiquement les données journalisées des conteneurs. Vous avez alors le choix de sauvegarder les données sur un espace de stockage persistant ou de suivre la procédure recommandée qui consiste à envoyer les journaux aux flux de sortie standard et erreur standard, afin de pouvoir les visualiser sur la console de gestion centrale.

Renforcez la sécurité

Le stockage des données sensibles, y compris les identifiants d'accès, est essentiel pour une application conteneurisée. Cela permet aux conteneurs de communiquer sans impliquer l'utilisateur de façon répétée. Cependant, le stockage des identifiants peut être problématique et exposer votre application à des risques de sécurité. En général, pour transmettre des données sensibles dans un conteneur, on utilise des variables d'environnement qui ne sont pas exposées publiquement. Les plateformes d'orchestration de conteneurs, telles que Kubernetes et OpenShift Container Platform, fournissent des fonctionnalités natives pour sécuriser les variables d'environnements et transmettre des données sensibles à travers une application conteneurisée.

Défis

Prenez une longueur d'avance sur les technologies à venir

L'écosystème des conteneurs connaît une évolution et une expansion extrêmement rapides. Il comprend des outils de création, déploiement, configuration, automatisation et gestion des conteneurs. Les projets Open Source qui prennent en charge cet écosystème sont très actifs. S'il s'agit d'un signe encourageant pour le futur de ces technologies, il peut s'avérer compliqué de développer ses compétences au même rythme afin de fournir des applications basées sur ces solutions. En tant que développeur, vous devez vous tenir informé des projets Open Source en cours, participer à des formations et faire évoluer vos compétences pour suivre les avancées en matière de conteneurs.

Adoptez la culture DevOps

Comme nous l'avons mentionné au début de ce livre numérique, les conteneurs transforment fondamentalement la manière dont les applications sont développées. De leur côté, les équipes chargées des applications doivent s'adapter pour modifier leurs processus et leur culture. Pour utiliser efficacement les conteneurs, les entreprises doivent adopter une culture DevOps saine, sans la séparation traditionnelle qui s'est développée sous le modèle monolithique. Pour créer un workflow rapide, fiable, cohérent et axé sur la sécurité, du développement jusqu'au déploiement, les entreprises doivent suivre et automatiser des principes DevOps. En tant que développeur, vous ferez partie du processus complet, du développement au déploiement, jusqu'à la livraison au client. Votre nouveau rôle implique d'assumer la responsabilité de vos conteneurs et de vous assurer que toute logique d'application que vous créez fonctionne en production. Puisqu'il est possible de publier des correctifs en temps réel, vous devrez déboguer, corriger et déployer votre code rapidement afin d'assurer le fonctionnement de l'application à son potentiel maximum.

Restez au top de la sécurité

Garantir la sécurité représente toujours un défi pour des développeurs d'application, que vous travailliez avec ou sans conteneurs. Si les conteneurs offrent l'avantage d'isoler les applications, leur prolifération dans une entreprise représente un nouveau type de risque pour la sécurité.

Les conteneurs impliquent généralement de diviser les applications en microservices plus petits, ce qui entraîne une augmentation du trafic de données et un alourdissement des règles de contrôle d'accès. À mesure que le nombre de conteneurs croît, le risque de relâchement des contrôles d'accès entre les conteneurs augmente. Le non-respect des protocoles de sécurité et d'accès peut vous amener à introduire des vulnérabilités dans votre environnement de production. De même, de nombreuses entreprises utilisent des référentiels d'images de conteneurs, cependant, elles doivent vérifier que ces images répondent bien à leurs exigences de sécurité et de conformité. Assurez-vous d'avoir parfaitement compris les risques liés à l'utilisation des conteneurs et d'adhérer aux politiques de contrôle d'accès.

Gestion et surveillance

La surveillance des conteneurs peut se complexifier à mesure que le nombre de conteneurs déployés augmente. Idem pour la gestion lorsque vous déployez des solutions de cloud hybride avec des conteneurs exécutés à la fois sur des clouds publics et privés. De plus, il n'est pas simple de mettre en œuvre un magasin de données cohérent pour accumuler, analyser et agir selon des événements générés à partir de tous les conteneurs et applications. Une fois un événement capturé, il peut être difficile de localiser l'origine des erreurs. Heureusement, certaines technologies émergentes telles que Istio, Prometheus et Jaeger peuvent vous aider à relever ces défis.

Conclusions

La conteneurisation est devenue l'outil favori des développeurs d'applications modernes d'entreprise. En effet, 91 % des développeurs cloud et des responsables du développement utilisent désormais des conteneurs pour certaines capacités sur site.⁷ Les conteneurs vous permettent de travailler plus intelligemment et efficacement, dans des environnements de développement cohérents qui permettent de développer et fournir rapidement des applications cloud-natives, qui s'exécutent partout.

L'objectif de ce livre numérique est de vous apporter les bases pour appréhender les conteneurs et comprendre l'influence et le potentiel qu'ils peuvent avoir pour vous et votre entreprise. De nombreuses ressources sont disponibles pour vous aider à profiter des connaissances acquises ici et les appliquer à vos projets actuels. Vous serez bientôt en mesure de séparer vos applications de votre architecture, de contribuer à l'évolution de votre entreprise et de votre propre carrière grâce à la méthode de développement la plus moderne, agile et avant-gardiste : la conteneurisation.

N'attendez plus pour vous lancer, c'est le moment idéal. Nous espérons que vous rejoindrez rapidement les rangs des développeurs d'applications cloud-native. Bonne chance !

En savoir plus

Vous souhaitez obtenir un coup de pouce pour apprendre à déployer des applications conteneurisées ? Découvrez les concepts de la conteneurisation et des cas pratiques en suivant un cours Red Hat en ligne. C'est gratuit. Vous découvrirez comment conteneuriser des applications et des services, les tester avec Docker, puis les déployer sur un cluster Kubernetes à l'aide de la solution Red Hat OpenShift Container Platform. Vous apprendrez également à développer et déployer une application à partir d'un code source grâce à l'outil S2I (source-to-image) d'OpenShift Container Platform.

En savoir plus et s'inscrire ▶

Ressources supplémentaires

[Udemy](#)
[Stack Overflow](#)
[Docker](#)
[AWS](#)
[Google](#)
[Red Hat](#)

À propos de Red Hat

Premier éditeur mondial de solutions Open Source d'entreprise, Red Hat s'appuie sur une approche communautaire pour fournir des technologies Linux, de cloud hybride, de conteneurs et Kubernetes fiables et performantes. Red Hat aide ses clients à intégrer des applications nouvelles et existantes, à développer des applications cloud-native, à standardiser leur environnement sur son système d'exploitation leader sur le marché ainsi qu'à automatiser, sécuriser et gérer des environnements complexes. Red Hat propose également des services d'assistance, de formation et de consulting primés qui lui ont valu le titre de conseiller de confiance auprès des entreprises du classement Fortune 500. Partenaire stratégique des prestataires de cloud, intégrateurs système, fournisseurs d'applications, clients et communautés Open Source, Red Hat aide les entreprises à se préparer à un avenir toujours plus numérique.

Copyright © 2019 Red Hat, Inc. Red Hat, Red Hat Enterprise Linux, Ansible, Ceph, CloudForms, Gluster, JBoss et OpenShift sont des marques de Red Hat, Inc., déposées aux États-Unis et dans d'autres pays. Linux® est la marque déposée de Linus Torvalds aux États-Unis et dans d'autres pays.

La marque verbale OpenStack et le logo en forme de lettre O carrée, ensemble ou séparément, sont des marques commerciales ou des marques déposées de l'OpenStack Foundation aux États-Unis et dans d'autres pays et sont utilisés avec l'autorisation de l'OpenStack Foundation. Nous ne sommes pas affiliés à l'OpenStack Foundation ou à la communauté OpenStack, ni approuvés ou sponsorisés par celles-ci.

Java, ainsi que toutes les marques commerciales et tous les logos basés sur Java, sont des marques commerciales ou des marques déposées d'Oracle America, Inc. aux États-Unis et dans d'autres pays.