Red Hat

La voie vers les applications cloud-native

Réussir la transition en 8 étapes

Red Hat
Application Services

Sommaire

Introduction

Aujourd'hui, la grande majorité des entreprises utilisent des applications pour interagir avec leurs clients, leurs partenaires et leurs équipes. Cette émergence rapide de nouvelles options numériques a profondément bouleversé les modèles économiques traditionnels, obligeant les acteurs historiques et les secteurs d'activité à évoluer et à moderniser leurs processus d'exploitation.

L'innovation dans le domaine des expériences numériques passe par l'adoption d'une culture plus agile qui permet d'adapter les modèles de développement et de distribution pour suivre le rythme effréné de la demande. Cependant, la plupart des entreprises n'ont pas les moyens de reconstruire leur base technologique, ni d'impulser du jour au lendemain de nouvelles pratiques et un nouvel état d'esprit. Pour gagner en rapidité et en agilité, elles décident alors de changer doucement mais sûrement leur culture, leurs processus et leurs technologies. Ce livre numérique décrit les huit étapes à suivre pour adopter une approche cloud-native des applications.

« Dans une stratégie de développement cloud-native, la collaboration entre les équipes compte tout autant que les technologies… Elle joue un rôle essentiel dans la mise en œuvre de cette approche itérative et flexible, et tous, l'équipe de développement comme les autres parties prenantes, doivent participer au processus de création, de programmation, de tests et de déploiement. »

Rapport de recherche Red Hat : La vision de Red Hat sur le développement d'applications cloud-native, juin 2021

Étape 1 : Adapter la culture et les pratiques au cloud

La transition vers les applications cloud-native suppose une multitude de changements au sein des équipes métier, de développement et d'exploitation, afin de gagner en rapidité et en efficacité lors de la création et du déploiement des applications. Il est important de coordonner l'ensemble des activités, technologies, équipes et processus de manière collaborative pour réussir le déploiement des applications cloud, quels que soient la taille et le secteur d'activité de l'entreprise. Les approches traditionnelles de gestion de l'utilisation des ressources des clouds publics ou hybrides favorisent l'autonomie et la réactivité des équipes. Elles cloisonnent cependant les processus liés aux données et aux environnements, ce qui freine l'innovation.

Dans le contexte actuel d'innovation rapide, la gestion des environnements distribués, des applications hautement personnalisées et des charges de travail liées aux nouvelles applications est un exercice délicat pour les entreprises qui développent une stratégie cloud unifiée pour leurs applications. Sans une stratégie cloud à l'échelle de l'entreprise, le potentiel des applications reste bien souvent inexploité.

L'instauration d'une culture cloud collaborative dans l'entreprise simplement avec de nouveaux outils et technologies est impossible. Il faut inciter et préparer les équipes à adopter une approche plus intégrée et collaborative du développement et de la distribution des applications. La culture des projets logiciels Open Source peut orienter la définition d'une stratégie cloud cohérente pour les applications.

Étape 2 : Rendre les applications plus rapides avec des microservices

Le développement de nouvelles applications ne doit pas devenir l'unique obsession des entreprises qui entament la transition vers une stratégie basée sur les applications cloud-native. La plupart des applications existantes sont en effet essentielles à la bonne marche de l'entreprise et à la génération de son chiffre d'affaires, et ne peuvent tout simplement pas être remplacées. Il faut plutôt les associer aux nouvelles applications cloud-native pour qu'elles fonctionnent toutes ensemble. Pour accélérer une architecture monolithique, il est nécessaire de la migrer vers une architecture plus modulaire basée sur des microservices et vers une stratégie de communication qui repose sur des interfaces de programmation d'application (API).

Avant de débuter l'opération onéreuse qui consiste à retravailler le code des applications existantes pour les convertir en microservices, il faut d'abord établir un socle solide pour l'architecture de microservices.

Plutôt que de migrer directement toute l'architecture, il est conseillé de progresser par phases, à son rythme, et de découper le monolithe en composants plus petits. L'utilisation d'une approche progressive permet de s'assurer que les applications seront créées selon des principes de conception fiables et avec des limites de domaine clairement définies. Ce type d'approche favorise une transition plus progressive et moins risquée vers une architecture de microservices, si besoin est, et pose les bases d'une architecture de microservices efficace.

Pour accélérer la mise à jour et le déploiement des applications hautement dépendantes des plateformes existantes, il est toujours possible de migrer le monolithe vers une plateforme basée sur des conteneurs. Cette migration offre l'avantage d'accélérer le déploiement et d'augmenter le retour sur investissement, sans compter que l'entreprise peut ajouter par la suite des fonctions ou des intégrations grâce aux techniques et approches cloud-native.

Étape 3 : Accélérer le développement avec des services d'applications

La capacité à réutiliser les ressources existantes a toujours représenté un enjeu majeur dans le domaine du développement logiciel. C'est également vrai avec les applications cloud-native. Pour en tirer tous les bénéfices en matière de rapidité et d'évolutivité, il convient cependant d'optimiser les composants réutilisables de ces applications et de les intégrer à l'infrastructure sous-jacente.

Au lieu de recréer des ressources (service de mise en mémoire cache, moteur de workflows ou de règles, connecteurs d'intégration, capacités de gestion des API et des appareils mobiles, service de virtualisation des données, broker de messages, framework serverless, etc.), il est plus intéressant de réutiliser des composants déjà optimisés et intégrés à l'infrastructure de conteneurs. Ces services d'applications sont directement mis à la disposition des équipes de développement sous forme de SaaS (Software-as-a-Service), de PaaS (Platform-as-a-Service) ou d'iPaaS (integration Platform-as-a-Service).

Si les approches DevOps et de conteneurisation accélèrent la distribution et le déploiement des applications, pour raccourcir les cycles de développement et de mise sur le marché des nouvelles applications cloud-native, il est possible d'utiliser l'un de ces services ou de les combiner. Les équipes de développement peuvent tirer parti de services d'applications conçus spécialement pour les infrastructures de conteneurs et capables d'en exploiter toutes les capacités, notamment les pipelines d'intégration et de distribution continues (CI/CD), les techniques de déploiement bleu-vert et de mises à jour continues, l'évolutivité automatisée et la tolérance aux pannes.

Étape 4 : Choisir l'outil adapté à chaque tâche

Grâce aux applications cloud-native, le langage ou le framework de développement est de plus en plus adapté aux besoins spécifiques de l'application métier. Cette situation entraîne une hausse de la complexité et de la diversité des applications, qui oblige les entreprises à se doter d'une plateforme d'applications basée sur des conteneurs pouvant combiner les frameworks, langages et architectures nécessaires au développement cloud-native.

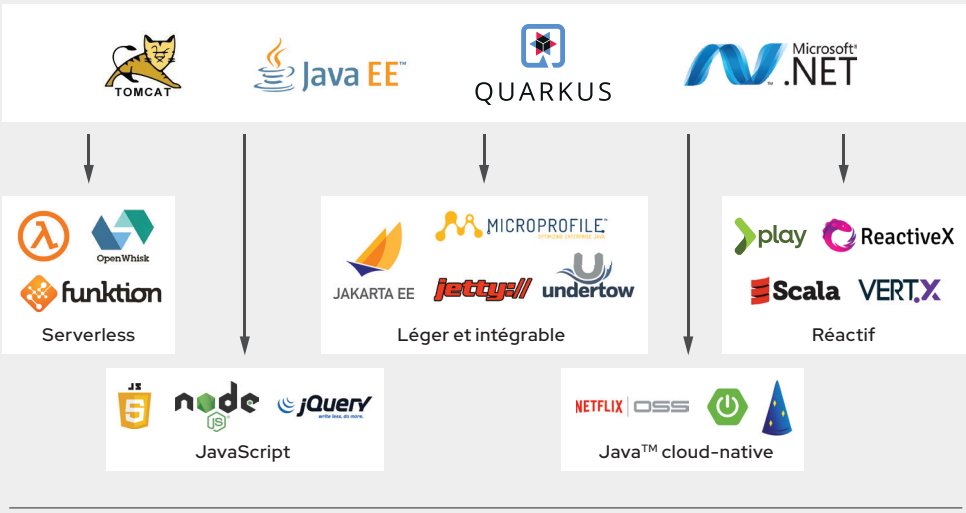

Figure 1 : le développement des applications cloud-native se diversifie

L'approche de développement cloud-native implique également de choisir l'outil adapté à chaque tâche. Il existe plusieurs stratégies pour le développement de ce type d'applications : approche à 12 facteurs, conception basée sur des domaines, conception et développement basés sur des tests, stratégie « monolith-first » (consistant à créer d'abord un monolithe) ou « fast monolith » (consistant à refondre un monolithe), miniservices et microservices. Dans tous les cas, la plateforme cloud-native doit proposer une combinaison de frameworks, de langages et d'architectures adaptée aux besoins de développement. En outre, la plateforme de conteneurs sous-jacente doit prendre en charge toute une sélection de frameworks et d'environnements d'exécution qui sont mis à jour en continu à mesure que les technologies évoluent.

Étape 5 : Fournir une infrastructure à la demande et en libre-service

Si les méthodes agiles ont permis aux équipes de développement d'accélérer la création et la mise à jour des logiciels, il est encore difficile d'accéder rapidement, à tout moment et depuis n'importe où à l'infrastructure. Les délais de mise sur le marché s'en trouvent immanquablement allongés. Il n'est plus viable d'attendre plusieurs semaines que l'équipe informatique mette à disposition les ressources demandées, car les compétences techniques représentent désormais un coût bien supérieur à celui de l'infrastructure.

Le provisionnement à la demande et en libre-service, qui permet à l'équipe de développement d'accéder à l'infrastructure quand elle en a besoin, représente une solution intéressante pour lutter contre le phénomène du shadow IT, ou informatique de l'ombre. Ce modèle n'est cependant efficace que lorsque l'équipe d'exploitation dispose d'un haut niveau de contrôle et de visibilité sur l'environnement, souvent dynamique et complexe.

Les technologies de conteneurs et d'orchestration des conteneurs séparent l'infrastructure sous-jacente, en facilitent l'accès et gèrent efficacement le cycle de vie des applications dans différents types d'environnements (datacenter, cloud privé et cloud public, etc.). Les plateformes de conteneurs offrent des fonctionnalités supplémentaires de libre-service, d'automatisation et de gestion du cycle de vie des applications. Avec ce modèle, les équipes de développement et d'exploitation peuvent mettre en œuvre rapidement des environnements cohérents. Au final, les équipes de développement peuvent se concentrer sur le processus de création sans subir les problèmes et délais induits par le provisionnement de l'infrastructure.

Les conteneurs assurent la portabilité des applications, qui permet notamment de déployer et d'exécuter une application cloud-native via n'importe quel fournisseur cloud. La portabilité offre la liberté de choisir son fournisseur, d'en changer, de diminuer les coûts et de développer une application multicloud sans être dépendant de l'API d'un fournisseur donné.

Étape 6 : Automatiser pour accélérer la distribution des applications

Afin d'accélérer la distribution des applications cloud-native, il est primordial d'automatiser l'infrastructure ou l'exploitation informatique, et d'éliminer ainsi les tâches manuelles. L'automatisation peut être intégrée et s'appliquer à toute tâche et tout composant, du réseau à la gestion des configurations et au déploiement des applications en passant par le provisionnement de l'infrastructure.

Les outils de gestion et d'automatisation fournissent des frameworks, des règles et des processus reproductibles qui peuvent remplacer ou limiter les interactions humaines retardant la mise sur le marché des applications. Ils peuvent eux-mêmes s'étendre à des technologies spécifiques telles que les conteneurs, des méthodes comme le DevOps et des domaines plus vastes comme le cloud computing, la sécurité, les tests, la surveillance ou les alertes. L'automatisation joue ainsi un rôle prépondérant dans l'optimisation de l'environnement informatique et la transformation numérique de l'entreprise, et accélère la création de valeur.

Découvrez le rôle majeur de l'automatisation informatique dans le livre numérique « L'entreprise automatisée ».

Télécharger le livre numérique

Conseils pour automatiser l'environnement informatique

1. Adoptez une approche programmatique de l'automatisation à l'échelle de l'entreprise. Favorisez le dialogue dans l'entreprise pour définir vos exigences en matière de services.
2. Considérez les sandbox d'automatisation comme le point de départ pour l'apprentissage du langage et des processus d'automatisation.
3. Automatisez le plus de tâches possible. Veillez à éliminer chaque étape manuelle superflue, même s'il est plus simple et tentant de les conserver.
4. Démarrez petit à petit, en fixant des objectifs réalisables et en suivant des méthodes systématiques. Chaque étape s'appuie sur la précédente pour forger des pratiques d'automatisation à l'échelle de l'entreprise.
5. Commencez par automatiser une tâche ou un service, comme le système de calcul, le réseau, le stockage ou encore le provisionnement. Partagez votre processus avec vos collègues pour qu'il serve de base aux prochaines initiatives d'automatisation.
6. Proposez des catalogues en libre-service pour offrir tous les éléments nécessaires aux utilisateurs et accélérer la distribution.
7. Mettez en œuvre des politiques et processus de mesure, surveillance et facturation réelle.

La distribution continue (CD) est une pratique de développement logiciel qui utilise l'automatisation pour accélérer le déploiement du nouveau code. Elle établit un processus par lequel les modifications qu'apporte une équipe de développement à une application peuvent être transférées vers un référentiel de code ou un registre de conteneurs grâce à l'automatisation.

Au fil du temps, les processus intégrés d'automatisation vont prendre de l'ampleur et offrir à l'entreprise une efficacité accrue, des pipelines DevOps accélérés et des innovations plus rapides.

Étape 7 : Mettre en œuvre la distribution continue et des techniques de déploiement avancées

Les cycles de lancement longs retardent la résolution des bogues et empêchent les entreprises de s'adapter rapidement à l'évolution des attentes du client et de la demande du marché. Dans le cas des applications à fort trafic (comme celles pour les appareils mobiles, le Web, l'Internet des objets et l'edge computing), un bogue non résolu peut affecter de nombreux utilisateurs et avoir des conséquences négatives sur l'expérience client, causer des problèmes de sûreté ou de

sécurité, ainsi qu'entraîner une baisse de la productivité ou du chiffre d'affaires. Le problème se pose aussi avec les autres applications métier internes : les pannes ou les bogues corrigés tardivement peuvent induire des coûts substantiels pour l'entreprise.

Les méthodes de développement agile ont évolué vers un modèle itératif. Les approches DevOps et de distribution continue s'inscrivent dans cette lignée et rapprochent les équipes de développement, d'exploitation, d'assurance qualité et de sécurité, ce qui améliore les processus de distribution des logiciels. Résultat : comme les modifications de code passent plus vite en production et de façon fiable, l'équipe de développement bénéficie d'un retour d'informations rapide. L'approche CI/CD crée ainsi une boucle d'itérations et aboutit à un système complet et automatisé qui couvre l'ensemble des aspects de la distribution d'applications : les tests automatisés, l'analyse des vulnérabilités, la conformité de la sécurité et les vérifications réglementaires. L'objectif de ces pipelines automatisés est de pouvoir lancer les mises à jour sans affecter la capacité d'exploitation et de limiter ainsi les risques lors de la distribution.

« Les techniques de déploiement avancées structurent et clarifient les projets novateurs. Avec ces méthodes bien développées, vous pouvez tirer pleinement parti du retour d'informations et des analyses, et vous disposez ainsi d'un véritable terrain d'expérimentation. Or, plus une idée peut être testée, meilleurs seront les résultats. »

Burr Sutter
Directeur de l'expérience développeur, Red Hat

L'intégration continue (CI) constitue la première étape vers la distribution continue (CD). Les systèmes de création de versions de type CI surveillent les modifications au niveau des référentiels de contrôle de code source. Ils réalisent les tests nécessaires, puis appliquent automatiquement les changements à la version actualisée de l'application.

Les techniques de déploiement avancées limitent les risques lors du lancement des logiciels et permettent d'effectuer des expérimentations dans un environnement spécifique. Les résultats sont contrôlés et le client n'est pas affecté de façon fortuite. L'atténuation des risques s'avère essentielle pour stimuler l'innovation au sein de l'entreprise.

Les techniques de déploiement avancées transforment intrinsèquement la distribution. Jusqu'à présent, les applications étaient distribuées le week-end, en dehors des heures de travail, selon des plages de maintenance bien définies et le système devait être arrêté. Désormais, le processus se déroule pendant la journée sans que la production ne soit interrompue, et le client a toujours accès aux applications.

Avec ces techniques, les entreprises ne subissent pas les inconvénients inhérents aux nouveaux déploiements et peuvent proposer des mises à jour et de nouvelles versions en fonction des besoins. Voici quelques-unes des techniques de déploiement sans temps d'arrêt couramment employées en fonction de cas d'utilisation spécifiques :

Déploiement de type mises à jour continues : les instances de l'application sont mises à jour individuellement, et non ensemble. Chacune est exclue de l'équilibreur de charge pour ne plus recevoir de trafic. L'instance exclue est ensuite actualisée, puis réintégrée dans le module, et ainsi de suite jusqu'à ce que toutes les instances aient été mises à jour.

Déploiement bleu/vert : ce modèle de déploiement repose sur l'exécution de deux environnements identiques, l'un actif et l'autre inactif. Après validation en production des modifications apportées à l'environnement inactif, le trafic réel est transféré vers l'environnement mis à jour. La restauration de la version précédente revient à rétablir le trafic (les données doivent également être transférées).

Déploiement canary : ce modèle de déploiement utilise deux environnements identiques, comme la technique bleu/vert, mais n'est pas contrôlé de la même manière. La nouvelle version est déployée auprès d'une poignée de clients qui peuvent la tester en production avant de la valider. Le trafic est ensuite transféré progressivement vers la nouvelle version. Les opérations sont surveillées et validées jusqu'à ce que celle-ci soit accessible à tous.

Étape 8 : Évoluer vers une architecture plus modulaire

Les microservices constituent une approche de développement logiciel qui consiste à décomposer les applications en leurs éléments les plus simples, indépendants les uns des autres. Contrairement à une approche monolithique classique, selon laquelle tous les composants forment une entité indissociable, les microservices fonctionnent en synergie pour accomplir les mêmes tâches, tout en étant séparés. Granulaire et léger, ce type de développement logiciel permet d'utiliser des processus similaires dans plusieurs applications. Bien que le choix de l'infrastructure soit libre avec les architectures de microservices, la base idéale reste une plateforme de conteneurs.

Une architecture de microservices peut constituer un avantage supplémentaire pour les équipes élargies ou celles qui effectuent des déploiements en production d'envergure plusieurs fois par jour. Ce type d'architecture nécessite de décomposer les différents services en unités de déploiement distinctes. Chaque microservice est alors géré et déployé de façon indépendante, potentiellement par différents équipes responsables de leur cycle de vie.

Il existe une autre solution possible : les miniservices. Dans ce type de configuration, les différents services sont découpés par domaine et exécutés généralement sur un serveur d'applications. Les miniservices améliorent l'agilité et l'évolutivité de l'environnement, sans la complexité liée à la conception et à l'infrastructure basées sur des microservices, mais ne dispensent pas d'investir dans les approches agiles, DevOps et CI/CD. L'idéal avec les miniservices est de combiner un serveur d'applications moderne ou une offre composée de plusieurs frameworks, architectures et langages, et une infrastructure de conteneurs.

La réussite de la stratégie basée sur les applications cloud-native repose en partie sur une plateforme capable de prendre en charge une diversité de frameworks, de langages et d'approches de développement (monolith-first, microservices ou miniservices).

Nos solutions

Quelles que soient vos priorités et votre situation dans la transition vers les applications cloud-native, nous vous proposons les technologies et services qui répondent à vos besoins.

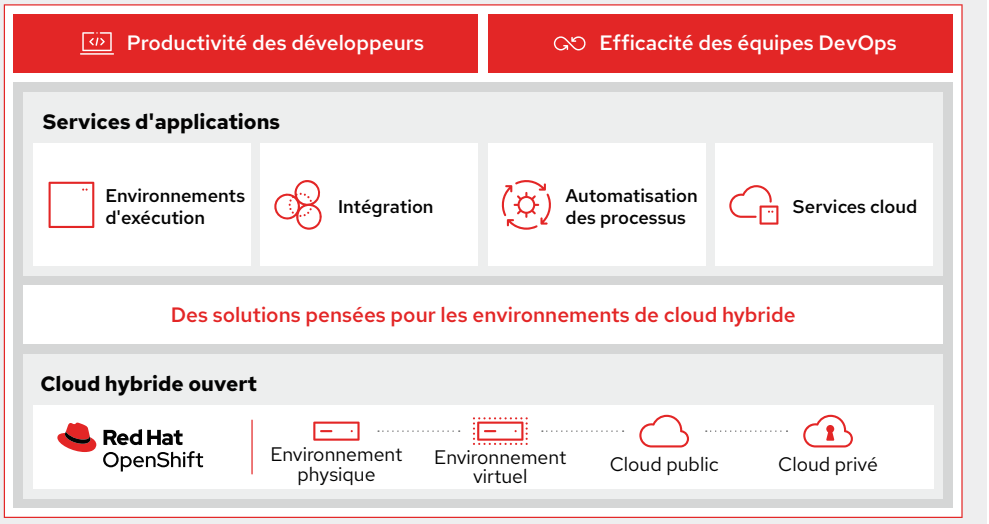

Figure 2 : la transition vers le cloud avec les solutions Red Hat

Certaines entreprises peuvent décider de se concentrer sur un cas d'utilisation cloud-native précis, alors que d'autres préfèreront mener de front plusieurs initiatives. Que votre démarche soit évolutive ou révolutionnaire, elle vous est propre et ne sera pas nécessairement linéaire. Peu importe le chemin emprunté, la mise sur le marché rapide des applications passe par des technologies, une culture et des pratiques DevOps adaptées.

Nous pouvons aider votre entreprise à réussir cette transition grâce à Red Hat® OpenShift®, notre plateforme cloud-native de développement et d'orchestration des conteneurs. Notre gamme de produits Red Hat Application Services pour le développement d'applications, qui fonctionne avec Red Hat OpenShift, comprend tous les frameworks, outils et solutions nécessaires pour développer, déployer et mettre à l'échelle des applications cloud-native. Si vous souhaitez accélérer la mise sur le marché de vos applications cloud-native, la gamme Red Hat OpenShift Application Services (incluse dans Red Hat Cloud Services) propose des services cloud gérés et hébergés pour Red Hat OpenShift qui simplifient le développement, le déploiement et la mise à l'échelle.

Pour vous aider à maîtriser la complexité du développement d'applications cloud-native, l'équipe des services de consulting Red Hat vous offre des conseils stratégiques et une expertise technique approfondie. Que ce soit lors de stages Red Hat Open Innovation Labs, dans le cadre de sessions de découverte ou pour établir un plan de mise en œuvre, nos consultants peuvent vous accompagner à chaque étape de votre transition.

À propos de Red Hat

Premier éditeur mondial de solutions Open Source, 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 développer des applications cloud-native, à intégrer des applications nouvelles et existantes ainsi qu'à gérer et à automatiser des environnements complexes. Conseiller de confiance auprès des entreprises du Fortune 500, Red Hat propose des services d'assistance, de formation et de consulting reconnus qui apportent à tout secteur les avantages de l'innovation ouverte. Situé au cœur d'un réseau mondial d'entreprises, de partenaires et de communautés, Red Hat participe à la croissance et à la transformation des entreprises et les aide à se préparer à un avenir toujours plus numérique.

facebook.com/redhatinc
@RedHatFrance
linkedin.com/company/red-hat

Europe, Moyen-Orient et Afrique (EMEA)
00800 7334 2835
europe@redhat.com

France
00 33 1 41 91 23 23
fr.redhat.com

fr.redhat.com
#F30307_1020