



LA VOIE VERS LES APPLICATIONS NATIVES POUR LE CLOUD

Réussir la transition en huit étapes

LIVRE NUMÉRIQUE

TABLE DES MATIÈRES

1. LA VITESSE : UN IMPÉRATIF À L'ÈRE DU NUMÉRIQUE	3
2. DÉFINITION D'UNE APPLICATION NATIVE POUR LE CLOUD	3
3. APPLICATIONS TRADITIONNELLES ET APPLICATIONS NATIVES POUR LE CLOUD	4
4. LES QUATRE PRINCIPES DU DÉVELOPPEMENT ET DU DÉPLOIEMENT D'APPLICATIONS NATIVES POUR LE CLOUD	6
5. LA VOIE VERS LES APPLICATIONS NATIVES POUR LE CLOUD, EN HUIT ÉTAPES.....	7
ÉTAPE 1 : Instaurer une culture et des pratiques DevOps.....	7
ÉTAPE 2 : Rendre les monolithes existants plus rapides	7
ÉTAPE 3 : Accélérer le développement avec des services d'applications	8
ÉTAPE 4 : Choisir l'outil adapté à chaque tâche	8
ÉTAPE 5 : Fournir une infrastructure à la demande et en libre-service	9
ÉTAPE 6 : Automatiser les processus informatiques pour distribuer plus rapidement les applications	9
ÉTAPE 7 : Mettre en œuvre la distribution continue et des techniques de déploiement avancées	10
ÉTAPE 8 : Évoluer vers une architecture plus modulaire	11
6. DÉFIS MÉTIER	12

« Les entreprises en avance sur la question du numérique ont huit fois plus de chances de croître, mais elles seront toujours à la traîne par rapport aux entreprises digital natives. »

Enquête de Bain : For Traditional Enterprises, the Path to Digital and the Role of Containers

« La locution adjective "natif pour le cloud" décrit les applications, les architectures, les plateformes/infrastructures et les processus qui, ensemble, créent un environnement de travail économique qui nous permet de faire face au changement plus rapidement et de diminuer la part d'imprévu. »

CHRISTIAN POSTA
ARCHITECTE EN CHEF CHEZ
RED HAT ET AUTEUR DU LIVRE
MICROSERVICES FOR JAVA
DEVELOPERS

SOURCE : INFOQ, « DEFINING
CLOUD NATIVE: A PANEL
DISCUSSION », 2017,

1. LA VITESSE : UN IMPÉRATIF À L'ÈRE DU NUMÉRIQUE

Appareils mobiles, capteurs intelligents, appareils connectés, réalité virtuelle, chatbots, blockchain, apprentissage automatique... l'univers du numérique évoque une multitude de technologies novatrices. Pour certains, il marque également la naissance rapide d'entreprises « digital natives », ou natives du numérique, qui bouleversent les modèles traditionnels et déstabilisent les secteurs et structures historiques. Ce qui est sûr, c'est qu'une entreprise qui décide de passer au numérique doit adopter une culture plus agile qui lui permet d'adapter ses modèles de développement et de distribution pour pouvoir suivre le rythme effréné de la demande. Or, la plupart des entreprises ne peuvent s'offrir le luxe d'abandonner complètement les technologies sur lesquelles elles se sont construites ou de changer du jour au lendemain les pratiques et les mentalités. Aussi, elles transforment en profondeur leur culture, les processus et les technologies de manière progressive pour gagner en vitesse et en agilité.

Les logiciels jouent un rôle de plus en plus important dans les interactions des utilisateurs avec l'entreprise et dans la course à l'innovation. Résultat : la vitesse de développement et de déploiement des applications est devenue le nouvel impératif à l'ère du numérique.

L'approche native pour le cloud consiste à **moderniser les applications existantes et à en créer de nouvelles** en appliquant les principes du cloud, c'est-à-dire en utilisant des services et des processus optimisés pour créer un environnement de cloud computing agile et automatisé. Ce livre numérique détaille les différentes étapes à suivre pour réussir la transition vers une stratégie basée sur les applications natives pour le cloud.

2. DÉFINITION D'UNE APPLICATION NATIVE POUR LE CLOUD

Une application native pour le cloud est conçue pour tirer parti des modèles de cloud computing et ainsi atteindre de hauts niveaux de rapidité, de flexibilité et de qualité, avec des risques moindres lors de son déploiement. Malgré son nom, une approche native pour le cloud ne s'intéresse pas à **où** les applications sont déployées, mais à **comment** elles sont créées, déployées et gérées.

Les approches natives pour le cloud sont semblables aux architectures de microservices. Toutefois, même si les microservices peuvent marquer l'aboutissement du développement d'applications natives pour le cloud, il faut franchir de nombreuses étapes pour atteindre un tel niveau de maturité et gérer des microservices en production. Il est cependant possible de profiter de tous les avantages de ce type d'application sans passer par les microservices. Nombre d'entreprises y parviennent en appliquant les mêmes principes, mais en s'attachant à créer des monolithes modulaires plus performants.

L'évolution vers le développement et la distribution d'applications natives pour le cloud englobe plusieurs aspects : la culture, les processus, l'architecture et les technologies. Il s'agit bien d'une transition et non d'une finalité, un cycle de changement souvent jalonné de difficultés.

3. APPLICATIONS TRADITIONNELLES ET APPLICATIONS NATIVES POUR LE CLOUD

Les processus de développement des applications traditionnelles et des applications natives pour le cloud présentent des différences, qui constituent autant d'arguments en faveur du changement.

TABEAU 1 : DÉVELOPPEMENT DES APPLICATIONS TRADITIONNELLES ET DES APPLICATIONS NATIVES POUR LE CLOUD

	TRADITIONNELLES	NATIVES POUR LE CLOUD
AXE	Longévité et stabilité	Commercialisation rapide
MÉTHODE DE DÉVELOPPEMENT	En cascade, semi-agile	Agile, pratiques DevOps
ÉQUIPES	Cloisonnement des équipes de développement, d'exploitation, de contrôle qualité et de sécurité	Collaboration des équipes DevOps
CYCLES DE DISTRIBUTION	Longs	Courts et continus
ARCHITECTURE D'APPLICATIONS	Couplage fort Monolithique	Couplage faible Basée sur des services Communication basée sur des API
INFRASTRUCTURE	Basée sur des serveurs Conçue pour les déploiements sur site Crée une dépendance Mise à l'échelle verticale Capacité maximale préapprouvisionnée	Basée sur des conteneurs Conçue pour les déploiements sur site et dans le cloud Portabilité des applications Mise à l'échelle horizontale Capacité à la demande

3.1 DÉVELOPPEMENT ET DISTRIBUTION DES APPLICATIONS TRADITIONNELLES

La plupart des applications essentielles pour les activités d'exploitation d'une entreprise n'ont pas été conçues pour offrir une expérience numérique. Caractérisées par des cycles de vie longs, elles sont créées sous la forme de monolithes fortement couplés et répondent à des spécifications qui sont clairement définies, souvent bien avant leur distribution.

Ces projets de développement reposent essentiellement sur des approches en cascade et séquentielles, et s'étalent dans le temps. Les pratiques semi-agiles ne s'y sont greffées que très récemment. Les étapes de développement, de tests, de mise en conformité pour la sécurité, de déploiement et de gestion des applications incombant à des équipes distinctes, qui travaillent de manière isolée et communiquent selon un schéma linéaire.

L'idée est de développer de grandes applications multifonctionnelles, fortement couplées et constituées, entre autres, d'une interface utilisateur, de plusieurs services d'applications et du code pour accéder aux données, le tout sans tenir compte de l'environnement technologique. Par exemple, une application de commerce en ligne créée selon ce modèle inclut toutes les fonctionnalités nécessaires pour l'interface utilisateur web, les catalogues de produits, le panier, les recommandations et évaluations de produits, les avis des clients, le système de paiement et les autres composants pour effectuer des achats sur le site marchand. Tout est regroupé dans une seule application.

La plupart du temps, l'infrastructure est déjà approvisionnée pour faire face aux pics de demande et, pour la mettre à l'échelle, il faut augmenter physiquement la capacité du serveur (mise à l'échelle verticale).

D'ici 2020, plus de la moitié des applications du mode 1* transférées depuis des datacenters privés vers un cloud public seront réécrites d'après les principes de l'architecture native pour le cloud. En 2017, le chiffre ne dépassait pas 10 %.

*Gartner : Why You Must Begin
Delivering Cloud-Native
Offerings Today, Not Tomorrow,
janvier 2018.*

3.2 DÉVELOPPEMENT ET DISTRIBUTION DES APPLICATIONS NATIVES POUR LE CLOUD

L'objectif ici consiste à commercialiser rapidement le produit, ce qui suppose un développement plus agile, basé sur des services et des API, ainsi qu'une approche de distribution continue. Cela passe par la collaboration DevOps entre les développeurs et l'équipe de distribution, une architecture plus modulaire et une infrastructure flexible qui peut être mise à l'échelle horizontalement à la demande, prendre en charge plusieurs environnements et assurer la portabilité des applications.

Les technologies cloud modernes procurent flexibilité et agilité, ce qui incite les entreprises à abandonner les applications traditionnelles au profit d'environnements de type cloud afin d'accroître leur agilité et la capacité de calcul à la demande.

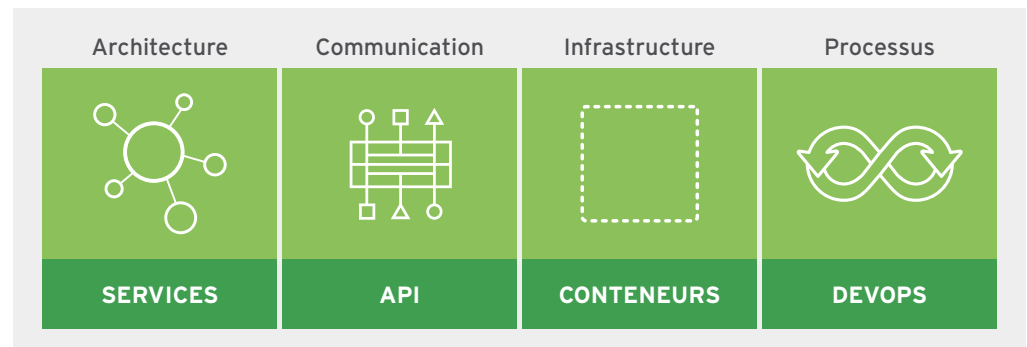
Or, dans le cloud, les capacités d'exploitation des plateformes traditionnelles deviennent bien souvent obsolètes, voire superflues. Parfois aussi, l'environnement cloud fournit d'emblée ces ressources et en assure l'exploitation. Ce type d'environnement simplifie effectivement la gestion du cycle de vie de l'hôte, car les entreprises peuvent exploiter les principes immuables de l'infrastructure et optimiser les hôtes par rapport aux besoins d'une instance d'application spécifique.

Il n'existe pas de chemin tout tracé pour adopter une stratégie basée sur les applications natives pour le cloud. Chaque situation est unique. Le numérique demande de fournir des services de qualité à un rythme soutenu. Pour y arriver, il ne suffit pas de créer des microservices. Pas plus que des outils de développement agile ou d'automatisation ne suffisent pour accélérer la cadence. Pour réussir, l'entreprise doit actionner toute une série de leviers : les pratiques, les technologies, les processus et les mentalités.

* Pour Gartner, l'approche bimodale consiste à gérer deux modes de fonctionnement distincts mais cohérents : le premier concerne la prévisibilité et l'autre l'exploration. Le mode 1 est optimisé pour tout ce qui est facile à prévoir et à comprendre. Il s'agit d'exploiter ce qui est connu et de transformer l'environnement existant pour l'adapter au numérique. Le mode 2 est celui de l'exploration, de la recherche de solutions aux nouveaux problèmes et de l'optimisation des aspects plus incertains.

4. LES QUATRE PRINCIPES DU DÉVELOPPEMENT ET DU DÉPLOIEMENT D'APPLICATIONS NATIVES POUR LE CLOUD

L'approche de développement et d'exécution appliquée aux applications natives pour le cloud exploite tout le potentiel du modèle de cloud computing. Elle est régie par quatre grands principes : une architecture basée sur des services, une communication basée sur des API, une infrastructure de conteneurs et des processus DevOps.



ARCHITECTURE BASÉE SUR DES SERVICES

L'architecture basée sur des services (comme les microservices) défend un modèle de services modulaires faiblement couplés. Ces architectures modulaires (miniservices et autres) accélèrent le processus de création sans pour autant compliquer l'environnement.



COMMUNICATION BASÉE SUR DES API

Des API légères, indépendantes des technologies, exposent les services et réduisent la complexité et le coût des opérations de déploiement, de mise à l'échelle et de maintenance. Elles offrent de nouvelles options et opportunités à la fois au sein de l'entreprise et en dehors.

Dans cette configuration, la communication s'effectue nécessairement via des appels d'interface de service sur le réseau. Il n'y a donc plus de risque à craindre concernant les liaisons directes, les modèles de mémoire partagée et le fait que le magasin de données d'une autre équipe puisse être consulté directement. Cela ouvre la voie à une multitude d'applications et de services, pour différents types d'appareils.



INFRASTRUCTURE DE CONTENEURS

Les applications natives pour le cloud reposent sur un modèle d'exploitation commun à base de conteneurs compatible avec différents environnements technologiques et sur la portabilité dans divers environnements et infrastructures, dont les clouds publics, privés et hybrides. Grâce aux conteneurs et aux fonctions de virtualisation du système d'exploitation, les ressources de calcul sont réparties entre plusieurs applications, lesquelles sont protégées et isolées les unes des autres.

La mise à l'échelle des applications natives pour le cloud s'effectue horizontalement. Autrement dit, la capacité peut être augmentée (souvent de façon automatisée) simplement en ajoutant des instances d'application à l'infrastructure de conteneurs.

Avec des coûts moindres et une grande densité d'applications, les conteneurs peuvent même résider pour la plupart dans la machine virtuelle ou dans le serveur physique, ce qui en fait une solution idéale pour la distribution des applications natives pour le cloud.



PROCESSUS DEVOPS

Le processus de développement d'applications natives pour le cloud respecte des méthodes agiles ainsi que les principes DevOps et de distribution continue. L'objectif est de développer et distribuer les applications en créant une synergie entre les équipes chargées du développement, de l'assurance qualité, de la sécurité, de l'exploitation et des autres activités en lien avec la distribution.

5. LA VOIE VERS LES APPLICATIONS NATIVES POUR LE CLOUD, EN HUIT ÉTAPES

ÉTAPE 1 : INSTAURER UNE CULTURE ET DES PRATIQUES DEVOPS

La transition vers les applications natives pour le cloud suppose une multitude de changements au sein des équipes de développement et d'exploitation, ceci afin de gagner en rapidité et en efficacité lors de la création et du déploiement des applications. Quel que soit son secteur d'activité ou sa taille, chaque entreprise doit considérer l'ensemble des activités, technologies, équipes et processus qui forment une culture DevOps. Si elle veut tirer parti des nouvelles technologies, de plus de rapidité et d'une collaboration renforcée, elle n'a pas d'autre choix qu'adhérer pleinement aux principes et aux valeurs du DevOps et s'organiser sur cette base.

Gérer plusieurs environnements distribués, des applications hautement personnalisées et les charges de travail liées aux nouvelles applications, alors même que l'innovation numérique ne cesse de s'accélérer, est un exercice délicat. Dans ce contexte, les pratiques DevOps peuvent être difficiles à mettre en œuvre dans certaines entreprises. Pourtant, le fait d'élargir les pratiques DevOps à la gamme d'applications offre un potentiel inexploité.

Instaurer une culture DevOps au sein de l'entreprise simplement avec des outils et des technologies est impossible. Il faut que les équipes soient convaincues et prêtes à adopter une approche plus intégrée et collaborative du développement et de la distribution. La culture autour des projets de logiciels Open Source peut vous servir de modèle pour instiller une culture DevOps dans votre entreprise.

Les stages [Red Hat Open Innovation Labs](#) visent précisément à guider les entreprises et à encourager l'expérimentation, le « Fast Fail » (ou échec rapide), la transparence dans les prises de décision, la reconnaissance et les récompenses, afin de renforcer la confiance et la coopération. Dans cet environnement conçu pour encourager l'innovation, les équipes exploitent des technologies Open Source novatrices pour créer rapidement des prototypes, tester les pratiques DevOps et adopter des workflows agiles.

Découvrez comment passer facilement au DevOps grâce à l'offre Red Hat Open Innovation Labs :

[TÉLÉCHARGER LE LIVRE NUMÉRIQUE](#)

« Si vous n'arrivez pas à structurer parfaitement une application monolithique, imaginez la difficulté de l'exercice avec plusieurs microservices ! »

SIMON BROWN
[CODINGTHEARCHITECTURE.
COM/PRESENTATIONS/
SA2015-MODULAR-MONOLITHS](https://codingthearchitecture.com/presentations/sa2015-modular-monoliths)

ÉTAPE 2 : RENDRE LES MONOLITHES EXISTANTS PLUS RAPIDES

Développer de nouvelles applications ne doit pas devenir l'unique obsession des entreprises qui entament la transition vers une stratégie basée sur les applications natives pour le cloud. En effet, la plupart des applications existantes s'avèrent fondamentales pour la bonne marche de l'entreprise et pour assurer son chiffre d'affaires, et ne peuvent tout simplement pas être remplacées. Ce qu'il faut au contraire, c'est les intégrer aux nouvelles applications natives pour le cloud. Comment rendre une architecture monolithique plus rapide ? En la faisant évoluer vers une architecture plus modulaire basée sur des services et vers une stratégie de communication qui repose sur des API.

Avant de débiter 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 monolithique. Même si les applications monolithiques sont connues pour leur manque d'agilité, leur mauvaise réputation s'explique, en partie, par leur mode de développement. Pourtant, un monolithe rapide peut parfaitement atteindre le niveau d'agilité des microservices, sans augmenter la complexité et les coûts.

S'intéresser à une approche monolithique rapide, c'est 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 autorise une transition plus progressive et moins risquée vers une architecture de microservices, si le besoin se fait sentir, et garantit par là même la réussite de la démarche.

Et quand bien même les applications n'auraient pas été conçues selon une approche monolithique rapide, il est possible d'augmenter leur cadence grâce à une plateforme de 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 natives pour le cloud.

Elle a aussi le choix de progresser par phases, à son rythme, et de découper son monolithe en composants plus petits.

ÉTAPE 3 : ACCÉLÉRER LE DÉVELOPPEMENT AVEC DES SERVICES D'APPLICATIONS

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

Quel est l'intérêt de recréer un service, un moteur de workflows ou des règles de mise en mémoire cache, des connecteurs d'intégration, des capacités de gestion des API et des appareils mobiles, un service de virtualisation des données, un broker de messages ou une structure sans serveur alors que vous pouvez réutiliser des éléments déjà optimisés et intégrés dans l'infrastructure de conteneurs ? Ces services d'applications sont directement mis à la disposition des développeurs sous forme de SaaS (logiciel en tant que service), de PaaS (plateforme en tant que service) ou d'iPaaS.

Pour raccourcir les cycles de développement et de mise sur le marché des nouvelles applications natives pour le cloud, les développeurs peuvent utiliser l'un de ces services ou les combiner. Alors que l'approche DevOps et les conteneurs accélèrent la distribution et le déploiement des applications, les services d'applications, eux, en accélèrent le développement.

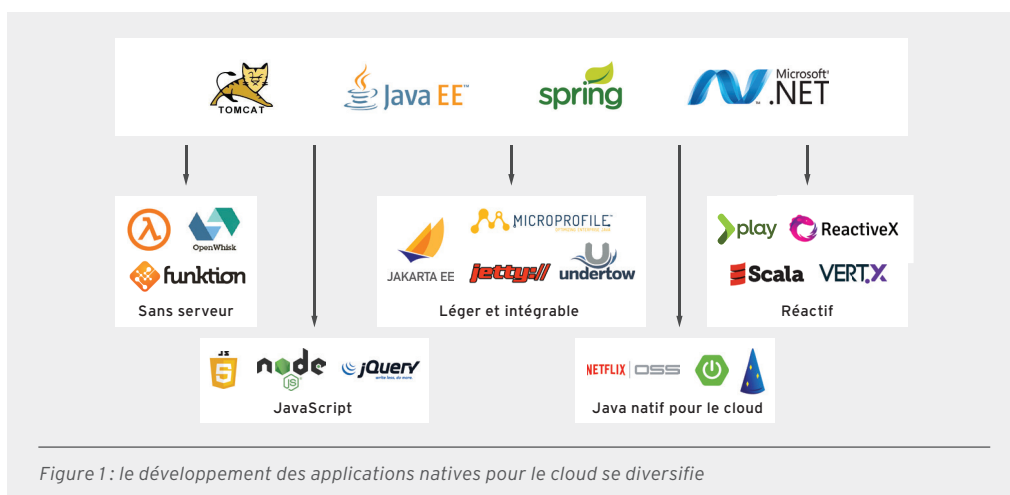
Les développeurs peuvent intégrer des services dans l'application pour s'assurer que celle-ci fonctionnera correctement au sein d'une infrastructure de conteneurs. Cela peut aussi être l'occasion de profiter des fonctionnalités de la plateforme, y compris des pipelines CI/CD, du déploiement blue-green et des mises à jour continues, de l'évolutivité automatique, de la tolérance aux pannes et plus encore.

ÉTAPE 4 : CHOISIR L'OUTIL ADAPTÉ À CHAQUE TÂCHE

La forte croissance de l'ingénierie logicielle, marquée notamment par l'avènement de l'Internet des objets, de l'apprentissage automatique, de l'intelligence artificielle, de l'exploitation des données, de la reconnaissance d'images et des véhicules autonomes, a donné naissance à une multitude de structures, langages et approches pour le développement logiciel.

Le processus de développement n'obéit plus à un modèle précis, puisque le choix du langage ou de la structure dépend de plus en plus des exigences spécifiques de l'application métier. Pour faire face à cette complexité accrue, les entreprises doivent se doter d'une plateforme d'applications basée sur des conteneurs qui peut combiner les structures, langages et architectures nécessaires au développement natif pour le cloud.

L'approche de développement natif pour le cloud implique également de choisir l'outil adapté à chaque tâche. Il est effectivement impératif que la plateforme offre la bonne combinaison de structures, langages et architectures en fonction de l'approche de développement retenue, que ce soit l'approche 12 facteurs, la conception par domaines, la conception et le développement basés sur



des tests, la stratégie MonolithFirst (Monolithe d'abord), le monolithe rapide, les miniservices ou les microservices. En outre, la plateforme de conteneurs sous-jacente doit prendre en charge toute une série de structures et de systèmes d'exécution organisés 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 développeurs 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. Attendre plusieurs semaines que l'équipe informatique mette à disposition les ressources demandées n'est plus viable, car les compétences techniques représentent désormais un coût bien supérieur à celui de l'infrastructure.

L'approvisionnement à la demande et en libre-service, qui permet aux développeurs d'accéder à l'infrastructure dont ils ont besoin, quand ils en ont besoin, représente une solution intéressante pour lutter contre le phénomène du shadow IT. Ce modèle n'est cependant efficace qu'à condition que l'équipe d'exploitation bénéficie du contrôle et de la visibilité sur l'environnement, souvent dynamique et complexe.

Les conteneurs et les technologies d'orchestration des conteneurs extraient l'infrastructure sous-jacente, en facilitent l'accès et assurent une gestion efficace du cycle de vie des applications dans différents types d'environnements (datacenter, cloud privé et cloud public, etc.). Une plateforme de conteneurs offre 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 développeurs se concentrent sur le processus de création sans subir les problèmes et délais induits par l'approvisionnement de l'infrastructure.

La standardisation est une autre composante majeure du modèle de libre-service. Elle permet aux entreprises d'automatiser les tâches et de distribuer les applications de manière cohérente afin d'atteindre leurs objectifs métier. La standardisation des processus consiste à mettre en correspondance la séquence exacte des événements et des activités nécessaires pour accomplir une tâche, comme la mise en production d'une application dans un nouvel environnement.

Les conteneurs assurent la portabilité des applications, qui permet notamment de déployer et d'exécuter une application native pour le cloud via n'importe quel fournisseur de 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.

Apprenez-en davantage sur les pratiques et techniques à mettre en œuvre pour adopter une stratégie basée sur les applications natives pour le cloud :

[DÉCOUVRIR L'OPEN PRACTICE LIBRARY](#)

ÉTAPE 6 : Automatiser les processus informatiques pour distribuer plus rapidement les applications

Afin d'accélérer la distribution des applications natives pour le cloud, il est primordial d'automatiser l'infrastructure ou l'exploitation informatique, et d'éliminer ainsi les tâches manuelles.

L'automatisation peut être intégrée et peut s'appliquer à toute tâche et tout composant, du réseau à la gestion des configurations et au déploiement des applications en passant par l'approvisionnement de l'infrastructure.

Les outils de gestion et d'automatisation fournissent des structures, des règles et des processus reproductibles qui peuvent remplacer ou limiter les interactions humaines qui retardent 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.

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 finalement forger des pratiques d'automatisation à l'échelle de l'entreprise.
5. Commencez par automatiser une tâche ou un service : le système de calcul, le réseau, le stockage ou l'approvisionnement. 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 plus d'autonomie aux utilisateurs et accélérer la distribution.
7. Mettez en œuvre des politiques et processus de mesure, surveillance et facturation réelle.

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

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

[TÉLÉCHARGER LE LIVRE NUMÉRIQUE](#)

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

La distribution continue (CD) désigne une approche d'ingénierie logicielle dans le cadre de laquelle les équipes développent des logiciels utiles selon des cycles courts et veillent à ce qu'ils puissent être lancés à tout moment. Cette stratégie de distribution, peu risquée et particulièrement fiable, permet d'adapter en continu les logiciels pour tenir compte de l'avis des utilisateurs, des évolutions du marché et des changements de stratégie de l'entreprise.

Définition de Gartner

Les cycles de lancement longs retardent la résolution des bogues et empêchent 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 et l'Internet des objets), 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 de lancements précoces et fréquents (« release early, release often »). 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, les développeurs bénéficient 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.

L'intégration continue (CI) constitue la première étape vers la distribution continue (CD). Les systèmes de création de versions de type CI, tels que Jenkins, surveillent les modifications au niveau de différents référentiels de contrôle de code source. Ils réalisent les tests nécessaires, puis créent automatiquement une version actualisée de l'application correspondant à chaque changement de contrôle du code source.

Découvrez comment les technologies d'automatisation modernes, telles que Red Hat Ansible® Automation, facilitent l'intégration et la distribution continues :

[TÉLÉCHARGER LE LIVRE BLANC](#)

« Les techniques de déploiement avancées structurent et clarifient les projets novateurs.

Avec ces méthodes matures, vous pouvez tirer pleinement parti du retour d'informations et des analyses, et 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

[REDHAT.COM/FR/ENGAGE/](https://redhat.com/fr/engage/teaching-an-elephant-to-dance)

[TEACHING-AN-ELEPHANT-TO-DANCE](https://redhat.com/fr/engage/teaching-an-elephant-to-dance)

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.

Ces techniques 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 du module d'équilibrage de charge pour éviter de recevoir du trafic. Elle 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 blue-green : 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 blue-green, 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.

Apprendre la danse à un éléphant

[TÉLÉCHARGER LE LIVRE NUMÉRIQUE](#)

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 un processus similaire 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 et les déploiements en production d'envergure, et ce à plusieurs reprises au cours de la journée de travail. Ce type d'architecture nécessite en effet de décomposer les différents services en unités de déploiement distinctes. Chaque microservice est alors géré et déployé de façon indépendante, potentiellement par différentes équipes responsables de leur cycle de vie.

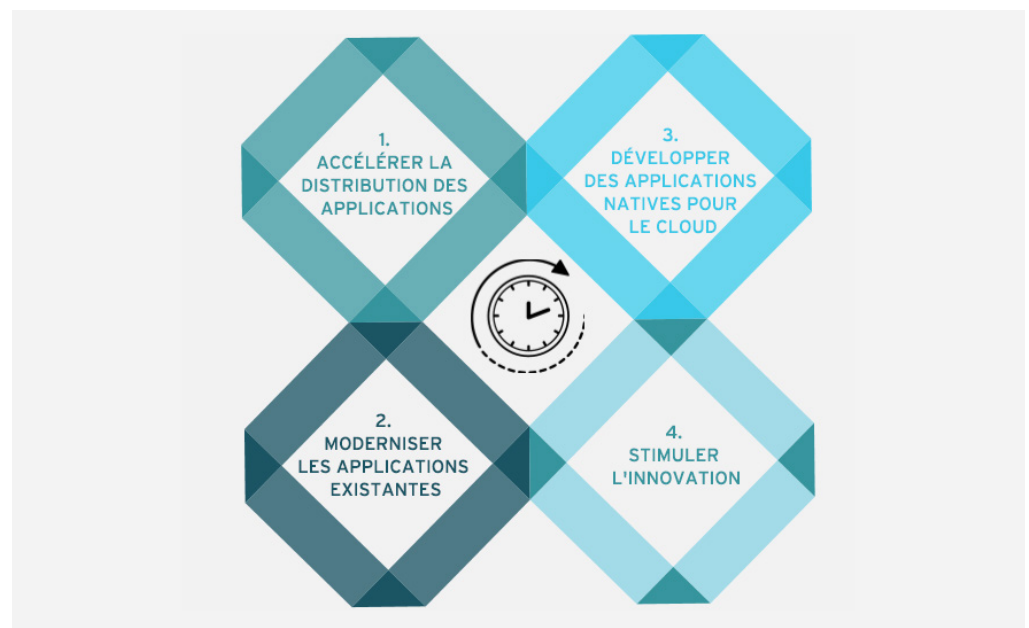
La mise en œuvre d'une architecture de microservices demande des investissements et des compétences, et peut représenter un changement trop radical pour l'entreprise. C'est pourquoi les analystes et les experts recommandent l'approche **MonolithFirst**, qui consiste à développer d'abord un monolithe, même si la finalité est de créer une architecture modulaire. Il s'agit donc, en premier lieu, de bien délimiter le domaine métier de votre application et de le découper ensuite en contextes bornés (ou « bounded contexts »), qui pourront être convertis en microservices. Cette approche supprime la dette technique, notamment les frais de réparation induits si les microservices avaient été créés sans avoir identifié le domaine et les contextes bornés de l'application.

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 compatible avec plusieurs structures, architectures et langages, et une infrastructure de conteneurs.

La réussite de la stratégie basée sur les applications natives pour le cloud repose en partie sur une plateforme capable de prendre en charge une diversité de structures, de langages et d'approches de développement (MonolithFirst, microservices ou miniservices).

6. DÉFIS MÉTIER

Les entreprises n'ont pas toutes les mêmes priorités en matière de transformation numérique. Certaines décident d'appliquer des stratégies basées sur des applications natives pour le cloud pour moderniser l'infrastructure et l'architecture d'applications existantes, tandis que d'autres innovent avec de nouveaux modèles économiques et des applications inédites. Quelles que soient leur intention et leur situation, elles partagent toutes un même objectif : gagner en rapidité et en flexibilité pour mieux se préparer au numérique. Les principaux cas d'utilisation des applications natives pour le cloud peuvent être classés selon quatre défis métier :



« Dès le premier jour qui a suivi l'achat de notre nouvelle banque, 10 modifications passaient en production, sans aucune faille. »

JOHN RZESZOTARSKI
RESPONSABLE DEVOPS, KEYBANK

Réduction du délai de déploiement : 1 semaine au lieu de 12

D'ici 2023, 90 % des applications actuelles seront encore utilisées, mais la plupart n'auront pas été suffisamment modernisées.

Gartner : Application Modernization Should Be Business-Centric, Continuous, and Multiplatform, janvier 2018.

« Grâce à l'assistance experte dont nous bénéficions pour Red Hat JBoss Enterprise Application Platform, le fonctionnement des opérations au quotidien se déroule sans souci. »

ANCIEN DIRECTEUR GÉNÉRAL
RESPONSABLE DES
TECHNOLOGIES D'ENTREPRISE,
AUSTRALIAN SECURITIES EXCHANGE

Redémarrage des applications
60 fois plus rapide

Réduction des frais et des délais d'assistance et augmentation des ressources disponibles pour le développement de services novateurs

DÉFI MÉTIER N° 1 : ACCÉLÉRER LA DISTRIBUTION DES APPLICATIONS

Objectif :

Accélérer la distribution des applications nouvelles et existantes

Approche :

Les conteneurs fournissent une plateforme commune aux équipes de développement, d'exploitation, de sécurité et de contrôle qualité, entre autres fonctions, et leur permettent d'adopter des pratiques DevOps, peu importe l'infrastructure et les technologies d'application en place. Grâce à l'approche DevOps, les équipes ont recours à l'automatisation, de même qu'aux principes CI/CD, pour distribuer les logiciels de manière rapide et fiable. L'automatisation basée sur les conteneurs résout les problèmes de déploiement. Elle permet ainsi d'accélérer le cycle de distribution et de l'adapter en fonction de l'activité, et non l'inverse.



Témoignage client :

Le groupe KeyBank, l'un des 15 principaux établissements bancaires aux États-Unis, s'est lancé dans un projet de modernisation de son canal numérique dans l'optique de rajeunir l'interface web et de développer une nouvelle application pour appareil mobile. Grâce à la solution Red Hat OpenShift®, il a pu transformer ses applications monolithiques en microservices et ainsi mettre en place un pipeline de distribution continue et automatisée, ce qui lui permet de déployer chaque semaine de nouvelles versions, contre une fois par trimestre avec l'ancien processus.

DÉFI MÉTIER N° 2 : MODERNISER LES APPLICATIONS EXISTANTES

Objectif :

Opérer plus rapidement des changements grâce à la **modernisation des applications existantes** afin de s'adapter à la demande des marchés et des clients

Approche :

De nombreuses applications métier utiles à l'entreprise n'ont pas été conçues pour le monde numérique. Il n'est cependant pas toujours faisable ni même économiquement viable d'annuler et de remplacer l'existant. En outre, la démarche de modernisation ne convient pas pour toutes les applications existantes.

Lorsque le projet de migration des applications traditionnelles vers le cloud est viable, la meilleure approche consiste à utiliser des conteneurs pour ainsi éliminer toute forme de dépendance vis-à-vis de l'infrastructure sous-jacente. Ce modèle assure la portabilité des applications depuis l'infrastructure sur site vers le cloud, où celles-ci peuvent, au besoin, être retravaillées et restructurées pour devenir des applications natives pour le cloud. Pour simplifier la migration des applications existantes dans le cadre d'une approche basée sur une plateforme de conteneurs, il est aussi possible d'exploiter les fonctionnalités d'automatisation de la plateforme et les pratiques DevOps.



Témoignage client :

L'ASX (Australian Securities Exchange) est la première place boursière à ouvrir chaque jour. Elle joue donc un rôle majeur dans le secteur international des services financiers. Cette institution doit bénéficier d'un niveau élevé de stabilité, de sécurité et de performances pour la conduite de ses opérations. Or, sa plateforme de serveur d'applications devenait instable, coûteuse et de moins en moins fiable. C'est pourquoi l'ASX a entrepris de moderniser sa plateforme numérique grâce aux nouvelles technologies et a déployé la solution Red Hat JBoss® Enterprise Application Platform pour fournir une base solide à son serveur d'applications. Le projet de migration a d'abord concerné ASX Online, une des applications web B2B stratégiques pour l'institution. C'est via cette application que l'ASX publie les cours des actions, des communiqués ainsi que des rapports sensibles à destination du marché, et respecte ses obligations réglementaires.

« Je suis conquis par la solution Red Hat OpenShift Container Platform. Elle est vraiment novatrice et nous permet de déployer nos conteneurs très rapidement tout en les contrôlant sans difficulté. »

MICHAEL AALBERS
COORDINATEUR D'APPLICATIONS
TECHNIQUES SENIOR,
AÉROPORT SCHIPHOL D'AMSTERDAM

Accélération de 50 %
du développement des
nouvelles API

DÉFI MÉTIER N° 3 : DÉVELOPPER DES APPLICATIONS NATIVES POUR LE CLOUD

Objectif :

Accélérer le cycle de **développement des nouvelles applications** pour saisir d'autres opportunités commerciales

Approche :

L'évolution de la demande des clients et du marché ouvre la voie à de nouvelles opportunités qui peuvent rapidement se concrétiser et donner naissance à des services et des produits. Les entreprises ont alors la possibilité d'évaluer les bénéfices qui en découlent dans ce nouvel environnement et d'opérer des ajustements si besoin. Avec l'architecture basée sur des services, les intégrations d'API, les services conteneurisés et l'orchestration des conteneurs, ainsi que les pratiques, les outils et l'automatisation DevOps, l'approche basée sur les applications natives pour le cloud donne un coup d'accélérateur à l'innovation.



Témoignage client :

Avec 64 millions de passagers par an, l'aéroport international de Schiphol est le troisième aéroport le plus fréquenté d'Europe. Il s'est fixé pour objectif d'être à la pointe du numérique d'ici 2018. Pour ce faire, il lui fallait se doter d'une plateforme compatible avec tout type de cloud pour accélérer le développement de ses applications. La stratégie numérique de l'aéroport de Schiphol s'appuie essentiellement sur des services distribués via des API, comme l'API propriétaire Flight, qui fournit aux passagers des informations relatives aux portes d'embarquement, aux terminaux et aux heures d'enregistrement, entre autres. La solution Red Hat OpenShift Container Platform permet aux informaticiens et aux partenaires de l'aéroport d'accéder à une plateforme multicloud en libre-service pour développer plus rapidement de nouveaux services.

« Le plus intéressant dans ce projet, c'est que nous allons repenser totalement notre environnement informatique. Nous allons modifier radicalement notre mode de fonctionnement en tant qu'entité et entamer la même transformation à l'échelle de tout le groupe. »

WAYNE MARCHANT
DIRECTEUR INFORMATIQUE,
HERITAGE BANK

DÉFI MÉTIER N° 4 : STIMULER L'INNOVATION

Objectif :

Accélérer l'innovation dans l'entreprise pour suivre la cadence imposée par le marché

Approche :

Dans un contexte professionnel qui évolue rapidement, l'immobilité est devenue synonyme de perte de vitesse. Les équipes informatiques sont à pied d'œuvre pour introduire rapidement de nouvelles fonctions et de nouveaux services qui répondent aux besoins des clients et qui permettent aux salariés de travailler plus efficacement. La réussite repose sur l'innovation constante et ne peut plus s'appuyer uniquement sur les technologies et les outils. Elle nécessite d'adopter une nouvelle culture et des outils et processus qui bouleversent les habitudes et soutiennent l'innovation et l'expérimentation dans les différents services.



Témoignage client :

Fondé il y a 142 ans, le groupe Heritage Bank est l'un des plus anciens établissements financiers en Australie. Dans un contexte de plus en plus concurrentiel et pour répondre aux nouveaux impératifs imposés par le marché, Heritage Bank devait trouver le moyen d'accélérer la distribution de ses logiciels. L'équipe a effectué un stage Red Hat Open Innovation Labs qui lui a permis de développer une solution bancaire novatrice et de devenir plus performante. Résultat : Heritage Bank parvient aujourd'hui à développer des logiciels d'avant-garde et plus efficaces, à un rythme plus soutenu.

[Vidéo sur Heritage Bank](#)

LES AVANTAGES DES SOLUTIONS RED HAT

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

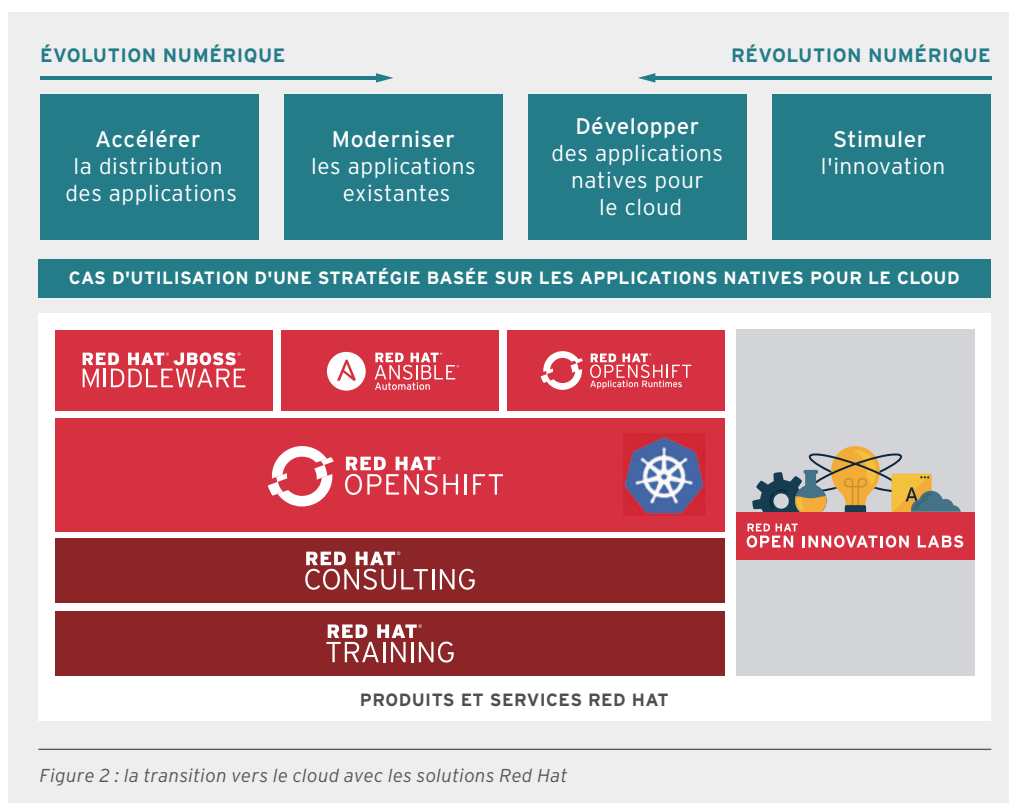


Figure 2 : la transition vers le cloud avec les solutions Red Hat

Certaines entreprises peuvent décider de se concentrer sur un scénario natif pour le cloud précis, alors que d'autres préféreront 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 de développement basée sur les conteneurs et native pour le cloud. Avec la solution [Red Hat OpenShift Application Runtimes](#), vous bénéficiez de structures et d'environnements d'exécution Open Source conteneurisés sur OpenShift qui accélèrent le développement des applications natives pour le cloud. Vous pouvez également déployer un large choix de technologies Red Hat JBoss Middleware sur OpenShift, y compris les [technologies d'automatisation et de gestion Ansible](#).

Pour vous aider à maîtriser la complexité de la transformation numérique, les [services de consulting Red Hat](#) vous offrent leurs conseils stratégiques et une expertise technique approfondie. Que ce soit lors des 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.

VOUS AVEZ ENTAMÉ UNE DÉMARCHE NATIVE POUR LE CLOUD ?

Découvrez comment nous pouvons vous aider tout au long du parcours :

- Faites appel aux services de consulting Red Hat : ils vous informeront sur les bonnes pratiques et vous guideront dans la mise en œuvre de votre projet lors d'une [session de découverte](#).
- Obtenez des renseignements précieux, des conseils et plus encore sur notre [blog Services Speak](#).
- Où en êtes-vous dans votre approche DevOps ? Êtes-vous prêt à vous tourner vers une stratégie basée sur les applications natives pour le cloud ? Passez l'évaluation [Ready to Innovate](#) pour le savoir.



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

fr.redhat.com
#F12255_0518

À PROPOS DE RED HAT

Premier éditeur mondial de solutions logicielles Open Source pour les entreprises, Red Hat s'appuie sur une approche communautaire pour proposer des technologies Linux, de cloud hybride, de conteneur et Kubernetes fiables et performantes. Red Hat aide ses clients à intégrer des applications nouvelles et existantes, à développer des applications natives pour le cloud, 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 certification primés qui lui ont valu le titre de conseiller de confiance auprès des entreprises du 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.

EUROPE, MOYEN-ORIENT
ET AFRIQUE (EMEA)
00800 7334 2835
europe@redhat.com

FRANCE
00 33 1 4191 2323
fr.redhat.com