

RÉUSSIR LA MISE EN PLACE D'UNE ARCHITECTURE DE MICROSERVICES

AVANTAGES

- Simplification du développement et de la maintenance
- Déploiement plus rapide et indépendant
- Évolutivité indépendante pour chaque composant de l'application
- Fin des engagements technologiques sur le long terme

SYNTHÈSE

Une architecture de microservices est un nouveau style d'architecture qui permet de créer des services faiblement couplés mais autonomes. Avec l'émergence de nouvelles tendances technologiques comme le DevOps, le PaaS (Platform-as-a-Service), les conteneurs et la CI/CD, les entreprises sont en mesure de créer et de gérer ces systèmes modulaires à une échelle plus grande que ne le permettaient les approches précédentes, par exemple l'architecture orientée services (SOA, « Service-Oriented Architecture »). Cependant, lorsqu'elles décomposent leurs applications monolithiques en microservices, le succès n'est pas toujours au rendez-vous. Pour utiliser efficacement les microservices, il faut bien comprendre pourquoi et comment ceux-ci doivent être mis en œuvre pour la création d'applications.

AMÉLIORER L'ARCHITECTURE ORIENTÉE SERVICES

L'architecture orientée services (SOA) est souvent définie comme un ensemble de composants applicatifs qui communiquent entre eux pour fournir des services à d'autres composants par l'intermédiaire d'un réseau. L'objectif de la SOA était de créer des applications distribuées résilientes sans composants centralisés et complexes.

Cette architecture était cependant fortement liée aux structures organisationnelles et destinée à soutenir les nouvelles structures internes. Son succès dépendait donc en grande partie des capacités organisationnelles résultant de cette restructuration et de la structure des équipes responsables de sa conception. Au lieu d'aboutir à des systèmes faiblement couplés mais autonomes, la SOA créait des systèmes extrêmement fragiles nécessitant une infrastructure complexe. Qui plus est, les premières mises en œuvre de SOA ont établi une dépendance des entreprises vis-à-vis de leur fournisseur, car le Middleware propriétaire mettait souvent l'accent sur la centralisation de la logique, de la persistance, de la gouvernance et de l'administration.

Avec les architectures de microservices, le potentiel de la SOA commence à se concrétiser à chaque étape de la création d'applications : développement, déploiement, opérations. Axée sur la simplification des technologies, l'architecture de microservices permet d'édifier des systèmes complexes aux composants rationalisés. La logique centralisée et l'infrastructure d'intégration, basées sur des plateformes lourdes et non standardisées, sont remplacées par des communications qui passent par des bus simples et standardisés, basés sur des protocoles asynchrones (HTTP ou protocoles de messagerie). SOAP, XML et autres protocoles et formats de données lourds cèdent la place au format léger JSON, associé à une architecture REST basée sur HTTP. Chaque microservice dispose de son propre magasin de données ; la gouvernance et la persistance centralisées ne sont pas nécessaires.

Les microservices s'appuient sur des méthodologies et des pratiques d'intégration continue (CI) et de livraison continue (CD), ainsi que sur plusieurs composants critiques qui n'étaient pas aussi courants dans la SOA, par exemple :

- la programmation et la persistance polyglottes ;
- les conteneurs ou les machines virtuelles immuables ;
- IaaS (Infrastructure-as-a-Service) et PaaS (Platform-as-a-Service) élastiques et programmables.



facebook.com/redhatinc
[@redhatnews](https://twitter.com/redhatnews)
linkedin.com/company/red-hat

ÊTES-VOUS PRÊT POUR LES MICROSERVICES ?

Votre service informatique a-t-il :

- créé une application monolithique structurée ?
- défini le besoin que les microservices sont censés satisfaire ?
- restructuré ses équipes autour des microservices ?
- adopté les bonnes pratiques en matière de DevOps et d'intégration/de livraison continue (de l'anglais « Continuous Integration/Continuous Delivery » ou CI/CD) ?
- délimité les différentes fonctionnalités métier au sein de l'application ?
- mis en œuvre l'orchestration des microservices, ainsi que des outils et des processus de gestion ?

UNE SOLUTION INNOVANTE POUR DES SERVICES INFORMATIQUES FLEXIBLES ET RÉACTIFS

DÉPLOIEMENT ACCÉLÉRÉ

Les microservices sont plus compacts en raison de l'importance accordée à la délimitation des domaines et à la modélisation cohérente du domaine ; ils nécessitent donc moins de code. Les stratégies de déploiement comme les archives ciblées et autonomes (souvent empaquetées sous forme de conteneurs Linux) et les CI/CD favorisent des déploiements plus rapides et plus fiables. En conséquence, le cycle de vie de développement du logiciel a tendance à s'accélérer. Les nouvelles fonctions et les correctifs logiciels, ainsi que les correctifs de sécurité intégralement testés, sont publiés plus rapidement.

CONTRÔLE MODULAIRE

Dans une architecture de microservices, chaque service peut évoluer de façon indépendante pour répondre à un pic temporaire ou saisonnier, effectuer un traitement par lots ou répondre à d'autres besoins métier. Les fonctions améliorées d'isolation des pannes font que les problèmes de fonctionnement, par exemple les fuites de mémoire ou les connexions ouvertes à la base de données, sont circonscrits au service concerné. L'évolutivité des microservices s'allie à la flexibilité des services dans le cloud pour vous permettre d'améliorer la prestation de service, tout en traitant simultanément un plus grand nombre de clients sans interrompre le service.

PLUS DE CHOIX

Le marché des microservices est dominé par les solutions technologiques Open Source et les méthodes organisationnelles. Par conséquent, avec les microservices, vous ne dépendez plus exclusivement d'un seul fournisseur et vous n'avez plus à vous engager sur le long terme en matière de technologies. C'est vous qui choisissez les outils nécessaires à la réalisation de vos objectifs informatiques et métier.

ÉTABLIR DES BASES SOLIDES POUR LES MICROSERVICES

Pour réussir la mise en œuvre des microservices, l'équipe informatique doit tout d'abord établir des bases solides sur lesquelles reposera son architecture monolithique. La modularité, la délimitation des domaines et la théorie fondamentale des systèmes distribués doivent être prises en compte et instaurées pour profiter au mieux des microservices.

D'autre part, c'est pour les systèmes complexes que les microservices sont les plus bénéfiques. Même si chaque service est indépendant des autres, certains critères opérationnels doivent être remplis, notamment en ce qui concerne les fonctions suivantes :

- DevOps.
- PaaS.
- Conteneurs ou machines virtuelles immuables.
- Réplication, registre et découverte de services.
- Surveillance proactive et alertes.

La satisfaction de ces conditions entraînant souvent des investissements considérables sans retour immédiat, les microservices ne sont pas forcément rentables pour toutes les équipes et tous les projets. En prenant une architecture monolithique comme point de départ, vous avez la certitude que vos applications obéiront à des principes de conception solides et que les domaines seront correctement délimités. Vous pourrez ensuite, le cas échéant, évoluer progressivement vers une architecture de microservices. Même les applications les plus simples, par exemple pour un panier d'achat, doivent respecter les règles suivantes :

- SoC.
- Forte cohésion et couplage faible, avec des interfaces de programmation d'application (API) bien définies.
- Interfaces, API et mises en œuvre séparées, selon la loi de Déméter ou « principe de la moindre connaissance ».
- Conception orientée domaine basée sur le regroupement d'objets apparentés.

Si le service informatique d'une entreprise mène à bien son projet de microservices, c'est grâce non pas à ses technologies, mais à sa structure organisationnelle. La clé du succès réside dans la flexibilité d'équipes dotées d'une structure organisationnelle horizontale, de capacités interfonctionnelles et d'une certaine autonomie.

Une fois que l'application monolithique devant être mise à l'échelle a été créée selon des principes d'architecture logicielle solides, elle peut être remodelée sous forme de microservices. La méthode de remodelage la plus efficace consiste à suivre les étapes suivantes :

1. Délimiter les frontières métier et les différences sémantiques dans le domaine de l'application et commencer à décomposer les domaines en microservices.
2. Trouver le composant qui subit les modifications les plus fréquentes (par exemple des mises à jour de règles métier liées à des calculs de prix ou des évolutions réglementaires) ou qui est souvent corrigé pour résoudre des problèmes de sécurité.
3. Après avoir défini les microservices de base en fonction des domaines, affiner les API qui permettent l'interaction entre les services. Composer ces API à l'aide de modèles de conception comme Aggregator, Proxy, Chained, Branch, Event-Driven, etc.

ÉQUIPES COMPÉTENTES ET BIEN ORGANISÉES

Si le service informatique d'une entreprise mène à bien son projet de microservices, c'est grâce non pas à ses technologies, mais à sa structure organisationnelle. La clé du succès réside dans la flexibilité d'équipes dotées d'une structure organisationnelle horizontale, de capacités interfonctionnelles et d'une certaine autonomie.

Pour qu'une équipe soit efficace et compétente, elle doit être organisée selon le principe de fonctionnalité et non pas selon l'architecture. C'est le cas des fameuses équipes « deux pizzas » d'Amazon, composées de 8 à 10 personnes chargées de la création et de la maintenance de leur service. La loi de Conway dit que les logiciels reflètent forcément la structure des équipes qui les créent. Les équipes segmentées en plusieurs parties, par exemple développement, opérations, assurance qualité et sécurité, voient leur flexibilité amoindrie et leur progression ralentie.

Ces problèmes peuvent être évités, ou du moins atténués, par l'établissement d'une méthode DevOps avant la transition vers les microservices. Cette approche, qui implique de déterminer les stratégies de communication à l'avance, permet également de ne pas se retrouver avec une SOA ratée au lieu d'une architecture de microservices réussie.

OUTILS DE GESTION EFFICACES

En plus de logiciels bien conçus et d'une équipe efficace et bien organisée, toute architecture hautement évolutive nécessite des outils qui facilitent la gestion des services et des composants d'application supplémentaires, par exemple :

- des outils de registre et de découverte de services comme Kubernetes ;
- des normes de packaging pour la mise en conteneurs d'applications, Docker par exemple, et des outils d'orchestration pour la réplication de conteneurs à grande échelle (p. ex. Kubernetes). OpenShift de Red Hat inclut ces deux technologies Open Source éprouvées ;
- des outils de création d'environnement d'intégration continue, comme Jenkins ou Shippable pour Docker et Kubernetes ;
- des outils de résolution des dépendances comme Nexus ;
- des outils de basculement et de résilience, notamment des bibliothèques comme Hystrix et Ribbon ;
- des outils de surveillance des services, d'alerte et d'événements, par exemple la pile ELK (ElasticSearch, LogStash, Kibana).

GESTION DES DONNÉES

La gestion des données est un autre facteur important à prendre en compte lors de la transition vers les microservices. Contrairement à la SOA, les microservices ne se partagent pas les données. Chaque microservice possède son propre magasin de données physique, le principe de la persistance polyglotte permettant à plusieurs moteurs de base de données de s'exécuter sous chacun des microservices. Par conséquent, il est possible de choisir tel ou tel magasin de données pour tel ou tel service, au lieu de stocker toutes les données dans un système de gestion de base de données relationnelle (SGBDR) d'entreprise.

L'existence de plusieurs copies de la base de données d'entreprise peut toutefois entraîner une hausse des coûts de licence et une complexité accrue. Qui plus est, il peut être nécessaire d'harmoniser les magasins de données pour plus de cohérence. Les outils génériques d'ETL (extraction, transformation, chargement) ou de virtualisation des données peuvent faciliter la normalisation des données. De même, le modèle de conception bien connu Event Sourcing permet d'ajuster les magasins de données en cas de changements rétroactifs.

CONCLUSION

L'architecture de microservices peut apporter de nombreux avantages aux services informatiques, que ce soit l'évolutivité indépendante des divers composants de l'application ou l'accélération et la simplification du développement et de la maintenance des logiciels. Les microservices ne conviennent cependant pas à toutes les équipes et à tous les projets, car ils nécessitent des investissements importants sans produire de retour immédiat. La transition vers les microservices doit être progressive ; la restructuration de certaines parties des applications existantes, sans aller jusqu'à la transition complète, présente elle aussi un certain intérêt. Pour réussir la mise en place de microservices, le service informatique doit commencer par créer une application bien pensée en fonction des normes de plateforme existantes, puis la restructurer en un ensemble de microservices adaptés aux besoins métier. Si les équipes, les processus et les outils sont bien choisis, les microservices peuvent procurer de nombreux avantages à l'entreprise : accélération du développement et du déploiement, simplification de la maintenance, amélioration de l'évolutivité, fin des engagements technologiques sur le long terme.

À PROPOS DE RED HAT

Red Hat est le premier fournisseur mondial de solutions Open Source, s'appuyant sur une approche communautaire pour fournir des technologies cloud, Linux, middleware, de stockage et de virtualisation fiables et performantes. Red Hat propose également des services d'assistance, de formation et de consulting reconnus. Situé au cœur d'un réseau mondial d'entreprises, partenaires et communautés open source, Red Hat participe à la création de technologies innovantes qui permettent de libérer des ressources pour la croissance et de préparer ses clients au futur de l'informatique.



facebook.com/redhatinc
[@redhatnews](https://twitter.com/redhatnews)

linkedin.com/company/red-hat

fr.redhat.com
INC0336100_0216

EUROPE, MOYEN-ORIENT
ET AFRIQUE (EMEA)
00800 7334 2835

fr.redhat.com
europe@redhat.com

TURQUIE
00800-448820640

ISRAËL
1-809 449548

ÉAU
8000-4449549