

# Réussir sa démarche DevOps

Réussissez votre initiative DevOps  
en passant de la collecte de données  
aux réponses

L'économie numérique avance à la vitesse d'Internet. De même, les entreprises doivent livrer et améliorer leurs logiciels et leurs services distribués plus vite que jamais, en appliquant des méthodes agiles et itératives de développement et de livraison des logiciels. Des mises à jour fréquentes et réduites permettent aux développeurs d'essayer rapidement de nouvelles fonctionnalités.

Des mises à jour fréquentes aident également à :

- tenir les applications à jour et rappeler leur existence aux clients ;
- apporter des réponses rapides aux menaces et à l'évolution des besoins opérationnels ;
- limiter le risque de casser un composant critique.

Les cycles de publication courts sont monnaie courante dans les secteurs numériques, où les entreprises sont constamment remodelées par des technologies comme les services cloud avec des applicatifs orientés logiciel, les applications mobiles et les réseaux de communication sociaux. Ces cycles donnent naissance à de nouvelles architectures, habitudes de conception, piles d'application et approches de livraison en continu.

Le DevOps est une approche de cycle de vie complet, basée sur une culture de collaboration. C'est la principale méthodologie employée par les entreprises modernes pour développer, fournir et prendre en charge des applications. Les entreprises qui adoptent le DevOps sont plus rapides, elles parviennent à répondre aux demandes des clients dans un monde numérique en évolution constante.

Pourtant, les rêves et les espoirs des adeptes du DevOps se brisent bien souvent face à la dure réalité de l'entreprise. Le DevOps ne tient pas toujours ses promesses, bien souvent à cause de complexités technologiques et bureaucratiques. Créer une culture agile, réactive, orientée données, et les processus qui la soutiennent, est plus facile à dire qu'à faire. Si les défis sont nombreux, la complexité de l'infrastructure et des applications, la télémétrie et l'analyse des données sont autant d'obstacles courants sur la voie de l'excellence du DevOps.

Les données ne s'arrêtent pas aux logs. Elles sont produites par les applications, le cloud, les infrastructures, les conteneurs, les applications mobiles, les bases de données, l'IoT, les réseaux et bien d'autres sources encore. L'activité et les performances de l'ensemble des applications, systèmes, clients, API et systèmes industriels créent des journaux, des métriques, des traces et des événements.

Pour atteindre l'excellence DevOps, il faut aller plus loin que les approches conventionnelles de la supervision des performances. Il faut analyser toutes ces données pour que les développeurs et les professionnels des opérations puissent consacrer leur énergie à faire avancer l'entreprise plutôt qu'à éteindre des incendies. Ils doivent chercher des réponses dans le contexte de l'ensemble de la pile technologique en temps réel, afin de comprendre rapidement l'impact exercé sur les utilisateurs tout en évitant les problèmes susceptibles de dégrader les activités.

## La réalité du DevOps et le cycle en huit phases

Le DevOps introduit des méthodes de développement agiles dans des applicatifs éphémères orientés services et des opérations axées sur une itération constante. Le code est défini, testé, déployé, observé et mesuré. En cas de problème, le code est modifié et le processus recommence. Autant de fois qu'il le faut.

Le « code » comprend le code de l'application, bien sûr, mais aussi l'architecture déclarative, la configuration du réseau, la supervision, etc. L'ensemble du processus exige de la vitesse, de l'agilité et une culture d'organisation favorable à des améliorations incrémentielles et au principe d'échec rapide, plutôt qu'à la publication de grandes mises à jour révolutionnaires.

La clé d'une boucle de feedback DevOps réussie réside dans les données : mesure et analyse de l'utilisation, des performances, des erreurs et des indicateurs commerciaux clés. Flexibles par nature, les plateformes d'observabilité permettent d'importer et analyser tout type de données et donnent des réponses basées sur l'IA, ce qui en fait un atout décisif pour les processus DevOps.

La complexité est l'ennemie de l'agilité, et l'agilité est indispensable au DevOps. Un processus DevOps typique comprend huit étapes associées à différents outils (Figure 1). Comme ces outils sont souvent open source ou ponctuels, ils ne sont quasiment pas intégrés. Leurs fonctionnalités peuvent se chevaucher, créant ainsi des pertes d'efficacité qui nuisent à la visibilité et à l'agilité. En raison de ce manque d'agilité, les problèmes durent plus longtemps que nécessaire, vous ralentissent et entraînent des coûts inutiles.



Figure 1 : Les grandes étapes d'un processus DevOps classique

Le symbole de l'infini est souvent employé pour décrire les étapes du DevOps, car il illustre élégamment les deux responsables interconnectés du processus. Le côté gauche décrit le cycle de développement et de test des logiciels (l'aspect « dev ») et le côté droit, le cycle de déploiement et d'exploitation (l'aspect « ops »). Pourtant, chaque étape du cycle correspond à un marché d'outils distincts, car on utilise couramment une dizaine d'outils spécifiques, voire plus, à chacune d'elle.

Cette abondance est un luxe pour les développeurs et les équipes d'exploitation, qui peuvent choisir les produits qui répondent le mieux à leurs besoins. Mais le mélange qui en résulte est un véritable cauchemar pour les chefs de projet, les responsables DevOps et les cadres commerciaux. Cette palette de systèmes DevOps produisant chacun des données de façon isolée ne fait que dégrader la visibilité sur l'ensemble du processus et entraîner la multiplication des outils. De plus, les développeurs et les équipes des opérations n'utilisent pas les mêmes langages et flux de dépannage, ce qui prolonge le MTTR en cas de défaillance.

Sans outils communs, il devient impossible de savoir si les « correctifs » résolvent la cause profonde du problème ou ne font qu'apporter une solution temporaire. Sans une visibilité totale sur l'ensemble du cycle de développement et de publication et sur les données d'utilisation, on ne peut pas valider la qualité, les performances et la sécurité d'une application. Par exemple, un logiciel de supervision de sécurité peut identifier une vulnérabilité, mais sans moyen de retracer le module dont elle est originaire, la vulnérabilité ne sera pas facilement résolue.

D'autre part, lorsqu'une entreprise manque de visibilité sur les métriques DevOps, elle ne peut atteindre l'agilité nécessaire pour essayer de nouvelles idées ou progresser de façon itérative face à l'évolution du contexte professionnel. Le résultat de ces défaillances est la piètre qualité du logiciel, facteur d'insatisfaction, voire de départ, des clients.

Avec une approche DevOps, des pipelines de développement découpus créent de multiples transferts chronophages tout au long du processus de livraison de l'application. Ces transferts peuvent introduire des fonctionnalités redondantes et réduire l'efficacité des équipes en imposant de nombreux changements de tâches. En moyenne, il faut 10 à 15 minutes pour recommencer à éditer du code après une seule interruption.\*

Demander aux développeurs de changer de contexte dans une opération classique de dépannage est catastrophique pour la productivité. Un workflow de livraison déconnecté dégrade également la collaboration et creuse les fossés entre les équipes qui utilisent des terminologies différentes et qui ont des perspectives diverses sur l'ensemble du projet. Ensemble, ces frictions dans la communication et le partage d'informations sapent la confiance et la coopération, ce qui contredit le postulat initial du DevOps.

---

**Sans une plateforme offrant un feedback basé sur les données, le DevOps n'a guère de chance d'atteindre son but.**

---

## Libérer toute la valeur du DevOps

DevOps est surtout un terme à la mode pour la plupart des dirigeants. Le manque de maîtrise du sujet conduit souvent à des attentes irréalistes. Les cadres espèrent beaucoup, mais sans comprendre correctement les principes, les processus et les défis du DevOps, ils vont au-devant de la déception.

Sans une plateforme d'observabilité offrant un feedback basé sur les données, le DevOps n'a guère de chance d'atteindre son but. La seule manière pour les entreprises d'éviter cet écueil consiste à utiliser les métriques pour mesurer, rendre compte et démontrer la réussite du DevOps, tout en utilisant les informations pour optimiser et améliorer sa pratique.

Pour libérer la véritable valeur du DevOps, il ne suffit pas d'adopter un mot à la mode : elle est le fruit d'un langage commun et d'une approche partagée de la qualité et de la livraison des logiciels.

## Le DevOps à l'œuvre

Les entreprises performantes sont en phase avec les besoins de leurs clients. Pour cela, elles s'appuient sur une plateforme d'observabilité qui fournit constamment des retours afin d'aider tous les domaines d'une entreprise : IT, développeurs d'applications, direction des lignes d'activité, équipes de sécurité, audit et autres. En fournissant des informations basées sur des faits et en informant la prise de décision, une plateforme d'observabilité permet d'agir rapidement pour réduire les coûts, adopter les bonnes idées et éliminer les mauvaises.

Pour le faire correctement, vous avez besoin de données, d'outils et de processus qui fournissent à toutes les parties prenantes des informations continues sur tous vos applicatifs. Le DevOps vous permet de réagir plus rapidement à ces informations, sur l'ensemble des unités commerciales, des applications et des opérations IT.

En pratique, l'association réussie du modèle DevOps avec une analyse complète et basée sur l'IA apporte trois grands avantages : l'accélération de la livraison des applications, l'amélioration de la qualité des applications et l'amélioration de la qualité des fonctions métier.

\* <http://blog.ninlabs.com/2013/01/programmer-interrupted/>

## Les fonctionnalités essentielles pour accélérer les résultats

Pour devenir agiles et assurer la satisfaction des clients, les équipes DevOps doivent disposer de quatre capacités essentielles :

- **Observabilité.** Chaque composant du workflow de développement et de livraison du DevOps doit être mesuré, et les données recueillies doivent être rassemblées dans un dépôt maître à des fins d'analyse. Sans données, il est impossible de comprendre et de résoudre les problèmes.
- **Itération.** Les corrections et améliorations de code doivent être rapidement identifiées, triées et développées à l'aide de données corrélées tout au long de la chaîne d'outils pour produire des informations plus détaillées.
- **Collaboration.** La rapidité de la livraison exige que les équipes DevOps soient toutes synchronisées, utilisent les mêmes données et agissent sur la base des mêmes mesures.
- **Optimisation.** Les dirigeants doivent s'efforcer d'améliorer constamment le processus en prenant des décisions fondées sur des faits. L'optimisation des processus nécessite d'apporter des réponses axées sur les données à des questions telles que :
  - Quel est notre délai « idée-revenus », notre débit de bout en bout jusqu'à la rentabilité ?
  - Combien de temps dure chaque phase de notre pipeline de livraison ?
  - Combien de temps les équipes passent-elles à écrire, à tester ou à revoir du code ?
  - Quelles équipes de développement sont les plus productives ?

**Les avantages du DevOps axé sur les données se répercutent directement sur les revenus : amélioration de l'efficacité des activités et de la productivité des développeurs, accélération de la livraison des applications, réduction des coûts, augmentation de la satisfaction des clients et croissance des revenus.**

## Créer de meilleures applications avec les informations en temps réel

Les données de toute la chaîne d'outils DevOps fournissent des informations qui permettent d'apporter une réponse proactive aux problèmes, aussi tôt que possible dans le cycle de développement et de test. Avec l'observabilité, les développeurs et les équipes d'exploitation sont en mesure de trouver des réponses aux problèmes avant que les clients et les utilisateurs ne soient confrontés à des dysfonctionnements frustrants. Pour apporter une réponse proactive, il faut une plateforme d'observabilité commune que tous, y compris les équipes DevOps, puissent utiliser comme version unique de la vérité, tout comme les systèmes de contrôle du code permettent aux développeurs de rassembler et partager leur travail.

En mesurant et en analysant l'intégralité du processus DevOps, on peut mettre au jour les données réelles de performance, d'utilisation et d'erreur, qui sont essentielles pour améliorer à la fois le produit final et le processus global de livraison. Une plateforme de données commune permet de corréler les informations de différents outils sur toute l'infrastructure, et de signaler les problèmes au plus tôt. Par exemple, en corrélant les données des contrôles de code à celles des systèmes de supervision des performances, on peut mettre au jour des problèmes avant que les utilisateurs ne remplissent des rapports de bug.

Pour harmoniser les données de tout le cycle de livraison DevOps, il faut une plateforme capable d'assimiler en temps réel les données de tous les outils utilisés pendant le cycle à huit phases décrit plus haut. Les outils de développement changent constamment, et les données aussi.

Les données sont les matières premières de la supervision du DevOps, mais il doit s'agir de métriques objectives capables de quantifier si le code répond aux spécifications fonctionnelles et opérationnelles, ainsi qu'aux SLA de qualité. Une fois le code déployé, les équipes doivent comprendre ce qui se passe au sein de leurs applications, au fil de chaque nouvelle publication dans le pipeline de distribution. Si vous ne comprenez pas ce qui se passe à l'intérieur de votre application, vous ne pouvez pas comprendre votre pipeline ni corréler les événements du pipeline avec les performances de l'application et l'expérience de l'utilisateur final. L'utilisation d'informations d'observabilité en temps réel pour analyser l'ensemble du cycle de livraison des applications offre de nombreux avantages :

- des références cohérentes, mesurables et traçables pour les taux de bugs, couvrant les équipes de développement et les versions du code ;
- une visibilité accrue sur les indicateurs d'assurance qualité pour identifier les problèmes avant la mise en production ;
- une compréhension plus rapide des microservices distribués en constante augmentation, permettant notamment de localiser les problèmes lorsqu'ils surviennent ;
- l'optimisation des dépenses cloud ;
- la génération d'alertes plus rapides et plus précises, un dépannage guidé et des informations rapides grâce à l'IA de flux ;
- une intégration plus étroite de la sécurité dans le processus de livraison DevOps, approche que l'on désigne de plus en plus souvent « DevSecOps ».

Les équipes de développement peuvent produire un code plus sécurisé et conforme en repérant et en éliminant les vulnérabilités dès le début d'un cycle de build. Le résultat net des informations en temps réel et basées sur les données en provenance du pipeline de création d'applications permet aux équipes de sécurité d'identifier les comportements dangereux et d'avertir la bonne équipe avant que le problème de sécurité ne soit enregistré dans la base de code. Disposer d'un processus automatisé pour sécuriser ces comportements fait de la sécurité une priorité dans votre processus de développement, ce qui est indispensable pour les applications actuelles.

## De meilleurs résultats métier

Les avantages du DevOps axé sur les données et de l'observabilité se répercutent directement sur les revenus : amélioration de l'efficacité des activités et de la productivité des développeurs, accélération de la livraison des applications, réduction des coûts, augmentation de la satisfaction des clients et croissance des revenus.

Les entreprises qui élargissent l'utilisation d'une plateforme complète d'observabilité, allant ainsi au-delà de son rôle traditionnel pour les opérations IT, en retirent des bénéfices multiples :

- une visibilité en temps réel sur l'utilisation, les performances, la fiabilité, les erreurs et les incidents de sécurité pour les nouvelles versions d'applications, à l'opposé des retards de plusieurs heures ou jours induits par le recours aux signalements ad hoc ;
- un MTTR considérablement plus bas, une résolution des problèmes de production jusqu'à 70 % plus rapide et une résolution des problèmes de pré-production accélérée de 40 % ;
- une efficacité accrue grâce à l'assimilation et à l'analyse automatiques des données. Les développeurs et les équipes des opérations peuvent se concentrer sur les besoins de l'entreprise, sans perdre de temps à créer et à maintenir des outils de supervision.

Pour que les équipes DevOps restent en phase avec les objectifs commerciaux, il faut mettre en place un processus de livraison en continu avec des publications fréquentes, mesurées et corrélées aux résultats commerciaux réels.

Ce processus de livraison accélère l'expérience client en produisant du code plus performant et plus utile. Avec un meilleur code, les clients sont plus satisfaits, et des clients satisfaits sont fidèles. En utilisant les données d'une plateforme d'observabilité, on peut relier directement les publications de code à la satisfaction des clients en analysant des métriques comme l'utilisation et les ventes d'une application. Plusieurs métriques métier sont utiles pour l'observabilité :

- les taux d'inscriptions de clients et de téléchargements ;
- l'évolution des revenus, et notamment les pics et les creux associés aux sorties de produits ;
- l'évolution de l'engagement des clients, des conversions et des abandons de paniers.

## L'analyse quantifie et améliore les bienfaits du DevOps

Il faut être agile pour réussir dans le monde numérique. Pour y parvenir, il faut un processus de publication DevOps efficace et une plateforme d'observabilité pour exploiter tout le potentiel d'un programme DevOps.

Les professionnels DevOps doivent continuellement améliorer la réactivité, la collaboration, la sécurité et la conformité réglementaire dans leur travail. Ces métriques influencent directement la réputation de l'entreprise et la satisfaction des clients grâce au développement rapide de produits innovants. Pourtant, l'adoption du DevOps peut présenter de nombreux défis, qui sont souvent le fruit de processus et de chaînes d'outils complexes. Envoyer les données de sécurité vers un outil, les données de performances des systèmes vers un autre et les données d'application vers un troisième peut freiner la corrélation. Nous devons aider les dirigeants d'entreprise, les décideurs IT et les équipes de développement à comprendre les avantages du DevOps et apprendre à les concrétiser.

Le secret de la réussite du DevOps réside dans une validation quantifiée. L'adage « On n'améliore pas ce qu'on ne mesure pas » est toujours aussi pertinent, et une plateforme d'observabilité centralisée est idéale pour réaliser des mesures et des évaluations.

Le DevOps offre des avantages importants s'il est correctement mis en œuvre. C'est un marathon, et non un sprint. Pour adopter pleinement le DevOps, vous devrez modifier non seulement les méthodes de développement des logiciels, mais également la façon dont ils sont publiés, exploités, supervisés et dépannés. Les changements sont nombreux, mais les avantages en termes de productivité, de satisfaction des développeurs et des clients, de même que les gains de vitesse et de fiabilité des versions, parlent d'eux-mêmes.

Pour réussir dans votre démarche DevOps aujourd'hui, vous devez faire de l'observabilité un élément essentiel de votre pratique de développement logiciel. L'observabilité vous aide à voir et à comprendre l'état de chaque applicatif en contexte. Face à la complexité des applicatifs cloudw-native d'aujourd'hui, l'observabilité permet aux équipes DevOps de maîtriser les « inconnues inconnues » en obtenant des réponses en temps réel sur les applications et en apportant des informations stratégiques aux différents acteurs de l'entreprise. Pour en savoir plus, consultez le [Guide pratique de l'observabilité Splunk](#), ou essayez [Splunk Observability Cloud](#) dès aujourd'hui.

Découvrez [Splunk pour le DevOps](#) ou essayez le logiciel Splunk [gratuitement](#).



En savoir plus : [www.splunk.com/asksales](http://www.splunk.com/asksales)

[www.splunk.com](http://www.splunk.com)