

Les 5 pratiques fondamentales du DevOps

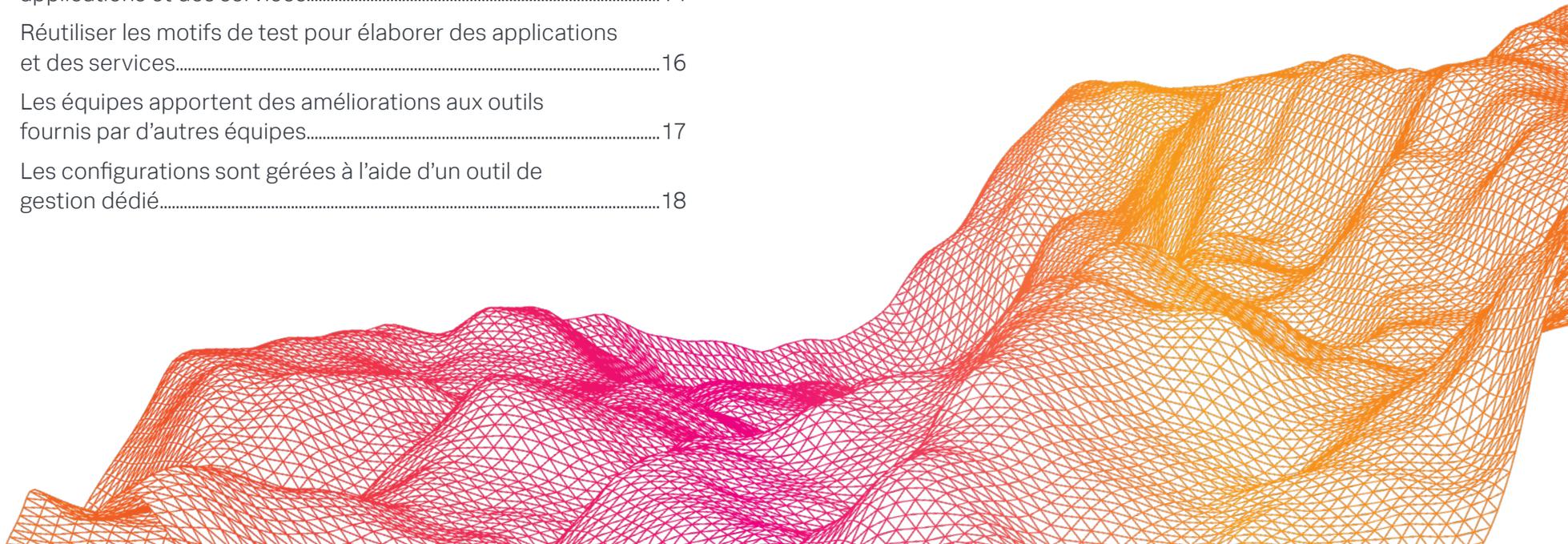
Comment les mettre en place et les développer



splunk>
turn data into doing

Sommaire

Introduction	3	Mettre en place les pratiques fondamentales dans votre organisation	20
Choisissez votre aventure.....	4	Mais... commencez là où vous en êtes.....	21
Pratiques fondamentales et modèle CAMS	5	La meilleure voie est celle qui donne des résultats rapides.....	22
Tout est question de partage.....	6	À propos de l'auteur	23
Les cinq pratiques fondamentales, une par une.....	7		
La supervision et les alertes sont configurables par l'équipe qui gère le service.	8		
Réutiliser les motifs de déploiement pour élaborer des applications et des services.....	14		
Réutiliser les motifs de test pour élaborer des applications et des services.....	16		
Les équipes apportent des améliorations aux outils fournis par d'autres équipes.....	17		
Les configurations sont gérées à l'aide d'un outil de gestion dédié.....	18		



Introduction

Il a fallu moins d'une décennie pour que le DevOps passe du statut de nouveauté à la mode à celui de bonne pratique reconnue pour la gestion des changements technologiques et la livraison des logiciels dans un environnement commercial extrêmement dynamique et concurrentiel. Mais en dépit des nombreux exemples présentés lors des conférences techniques du monde entier, dans les livres et dans les blogs réputés, la plupart des équipes ont encore du mal à mettre en place le DevOps et à savoir quoi faire ensuite.

Nous avons cherché à savoir comment les organisations progressent dans leur parcours DevOps et, après avoir analysé les données, nous avons découvert que celles qui réussissent le mieux ont suivi des étapes bien spécifiques. Notre étude a également mis au jour un ensemble de pratiques de base que nous appelons « pratiques fondamentales » et qui sont indispensables à la réussite de l'ensemble de la transformation DevOps.

Choisissez votre aventure

Le point de départ de chaque entreprise est unique. Il est constitué des technologies héritées, des façons de faire établies, des missions commerciales spécifiques de l'entreprise et de sa culture particulière. Il n'existe donc pas de voie unique pour la transformation DevOps, mais plutôt de nombreux parcours d'évolution possibles.

L'analyse des données du rapport État de l'observabilité en 2021 a révélé les pratiques fondamentales utilisées par les équipes performantes. Ces pratiques sont si étroitement corrélées au succès du DevOps que nous en avons conclu qu'elles sont essentielles à chaque étape du développement du DevOps. En d'autres termes, les pratiques qui doivent être adoptées à une étape donnée pour passer à la suivante conservent leur pertinence même pour les entreprises qui sont plus avancées dans leur parcours et qui affichent déjà des résultats brillants.



Chaque pratique fondamentale peut être résumée en une phrase :

- **la supervision et les alertes** sont configurables par l'équipe qui gère le service ;
- **les motifs de déploiement** sont réutilisés pour élaborer des applications et des services ;
- **les motifs de test** sont réutilisés pour élaborer des applications et des services ;
- **les équipes apportent des améliorations** aux outils fournis par d'autres équipes ;
- **les configurations sont gérées** à l'aide d'un outil de gestion dédié.

Lorsque nous avons examiné plus précisément chacune de ces pratiques, nous avons découvert que les entreprises les plus évoluées étaient bien plus susceptibles d'utiliser ces pratiques tout au long du parcours d'évolution que les entreprises moins avancées. Nos conclusions sont nettes : les pratiques fondamentales énumérées ci-dessus font partie intégrante du DevOps et sont indispensables à leur réussite.

Pratiques fondamentales et modèle CAMS

Quiconque a pour le DevOps un intérêt authentique ne sera pas surpris par l'importance des éléments fondamentaux de cette approche. D'autres concepts réputés sont élaborés sur les mêmes fondements. Le modèle CAMS¹, qui est l'une des toutes premières descriptions du DevOps, englobe ces pratiques fondamentales et reconnues, de l'importance des indicateurs au partage en passant par l'impératif d'automatisation. D'autres méthodologies courantes dans le discours DevOps (décalage vers l'amont, autonomie des équipes, développement basé sur les tests, etc.) viennent encore renforcer ces pratiques fondamentales.

Culture
Automatisation
Mesure
Partage

¹ Culture, Automatisation, Mesure, Partage (Sharing). Le modèle CAMS a été défini par Damon Edwards et John Willis en 2010. Pour une discussion plus détaillée du modèle CAMS et du DevOps, consultez « CAMS et le modèle d'évolution DevOps », un chapitre du [Rapport Puppet 2018 sur l'état du DevOps](#).

Tout est question de partage

En étudiant les pratiques fondamentales, nous avons pris conscience qu'elles reposent toutes sur le partage et que toutes l'encouragent. Pour comprendre le partage au sein d'une pratique DevOps, nous commençons par la configuration, la supervision et les alertes.

La supervision et les alertes sont :

- configurables par l'équipe qui exploite le service ;
- essentielles pour partager des informations sur le fonctionnement des systèmes et des applications ;
- conçues pour donner à tout le monde une vision commune du fonctionnement idéal des systèmes.

Cette compréhension commune est vitale pour apporter des améliorations, que ce soit au sein d'une seule équipe ou fonction ou à une échelle plus vaste. On observe également d'autres modèles de partage :

Les motifs de déploiement et de test sont réutilisés dans l'élaboration des applications et des services. Partager les motifs qui fonctionnent entre plusieurs applications et services implique souvent de les partager entre différentes équipes, en établissant des méthodes de travail convenues servant de fondement à de futures améliorations.

Les équipes apportent des améliorations aux outils fournis par d'autres équipes. Cette forme de partage encourage les discussions entre les équipes au sujet des priorités et des projets d'amélioration des outils, des processus et des mesures.

Les configurations sont gérées à l'aide d'un outil de gestion dédié. Un outil de gestion des configurations permet aux équipes de développement et de sécurité, ainsi qu'aux autres équipes en-dehors des opérations, d'apporter des modifications aux configurations du système et des applications. La responsabilité de l'opérabilité et de la sécurité est ainsi répartie entre les différentes fonctions de l'entreprise.

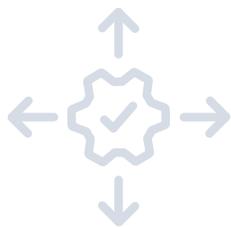
Nous avons observé que toutes les pratiques fondamentales permettent ou reposent sur le partage, ce dont nous pouvons déduire que la clé de l'élargissement du DevOps réside dans l'adoption de pratiques qui encouragent le partage.

C'est assez logique : quand on voit une approche qui fonctionne bien, on cherche à reproduire cette réussite et à la faire connaître. Imaginons que votre équipe ait réussi à déployer une application 10 fois, alors que ce type de déploiement est généralement source de nombreuses difficultés pour elle (et pour d'autres). Il y a fort à parier que quelqu'un s'en apercevra et voudra savoir comment vous avez fait. C'est comme cela que les pratiques DevOps se diffusent chez plusieurs équipes.



Les cinq pratiques fondamentales, une par une

Dans cette section, nous décrivons chacune des pratiques fondamentales en détail, en précisant de quelle façon la pratique en question contribue à l'évolution du DevOps.



D1



La supervision et les alertes sont configurables par l'équipe qui gère le service.

Au cœur du mouvement DevOps se trouvent l'autonomie et la responsabilité (deux faces d'une même médaille) comme le résume Werner Vogels, Directeur technique d'Amazon, dans cette célèbre citation : « Ce que vous développez, vous le gérez. »² Nous avons donc cherché, dans le cadre de notre étude, à savoir combien d'équipes qui exploitent des applications et des services en production, qu'elles soient composées de développeurs, d'opérateurs, d'ingénieurs de publication ou autres, ont la possibilité de définir leurs propres critères de supervision et d'alerte.

Les équipes autonomes qui administrent des applications et des services en production savent définir ce qu'est un bon service, comment déterminer s'il fonctionne correctement et comment savoir si ce n'est pas le cas. Cette approche autonome de la supervision peut prendre de nombreuses formes. Par exemple :

- « Déposez une configuration de supervision à tel emplacement et nous la mettrons en place. »
- « Connectez-vous à cette interface web pour configurer votre supervision. »
- « Ajoutez des sorties que l'on puisse interroger à votre code d'infrastructure. »
- « Voici une API pour configurer la supervision sous forme de code. »

Selon nos observations, les entreprises qui enregistrent de bons résultats avec le DevOps sont en mesure de définir leurs propres critères de supervision et d'alertes pour les applications et les services en production. Les équipes autonomes qui administrent des applications et des services en production savent définir ce qu'est un bon service, comment déterminer s'il fonctionne correctement et comment savoir si ce n'est pas le cas. Elles s'appuient également sur l'automatisation pour démultiplier l'effet de leur travail et minimiser l'effort (surtout les tâches répétitives à faible valeur ajoutée).

² Gray, J., Vogels, W., A Conversation with Werner Vogels, ACM Queue, <https://queue.acm.org/detail.cfm?id=1142065>, juin 2006, consulté en août 2018.

Donner aux équipes les moyens de définir, gérer et partager leurs propres critères de mesure et d'alerte favorise plusieurs aspects de la transformation DevOps :

- partage des indicateurs pour encourager l'amélioration permanente ;
- création et promotion d'une culture d'apprentissage continu ;
- collaboration inter-équipes et autonomie des équipes ;
- développement d'une réflexion systémique chez les personnes et au sein des équipes

Ces facteurs sont essentiels pour une culture DevOps solide, comme nous en avons discuté plus tôt, et on ne sera donc pas surpris que les organisations les plus avancées que nous ayons interrogées aient adopté cette pratique de façon précoce.

Que mesurer, et comment

Notre étude a mis en évidence les éléments à superviser et des critères d'alerte :

- **les indicateurs techniques clés** tels que la latence, les temps de réponse, l'utilisation des ressources, etc. Une majorité de participants (56 %) les évoquent ;
- **les objectifs commerciaux** dérivés des indicateurs techniques, par exemple, le délai de réponse peut être une mesure de la satisfaction des clients. Plus d'un tiers des participants (37 %) mesurent ces indicateurs ;
- **l'accès à la demande** à de véritables indicateurs métier tels que le temps passé sur le site, les ouvertures d'applications, les inscriptions de clients, les taux de revenus, etc. Ces indicateurs sont disponibles chez 20 % des participants.

La collecte de ces données offre la possibilité d'analyser et de comprendre pourquoi les problèmes se sont produits dans les systèmes modernes et complexes. La disponibilité de toutes ces données permet à quiconque de localiser les problèmes et de les corriger plus rapidement.



Importance du suivi des métriques commerciales réelles

Nos recherches sur l'État de l'observabilité ont révélé que les problèmes ignorés avaient d'autres conséquences : 45 % des participants évoquent une baisse de la satisfaction client, et 37 % une perte de revenus à la suite d'interruptions de service. Les métriques commerciales peuvent vous aider à identifier ces problèmes avant qu'ils n'aient un impact sur votre entreprise.

Certaines équipes informatiques utilisent ces métriques comme mesure principale des performances IT.

Les métriques métier répondent à des questions quotidiennes :

- Est-ce que les clients sont en mesure de faire des achats chez nous ?
- Combien de personnes servons-nous ?
- Combien gagnons-nous ?
- Ce motif est-il normal ?

Les métriques métier sont également essentielles pour mesurer le succès à plus long terme :

- Est-ce que la nouvelle version rencontre le succès escompté ?
- Est-ce qu'elle donne de meilleurs résultats commerciaux ?
- Quel est son impact sur l'entreprise par rapport aux autres versions ?

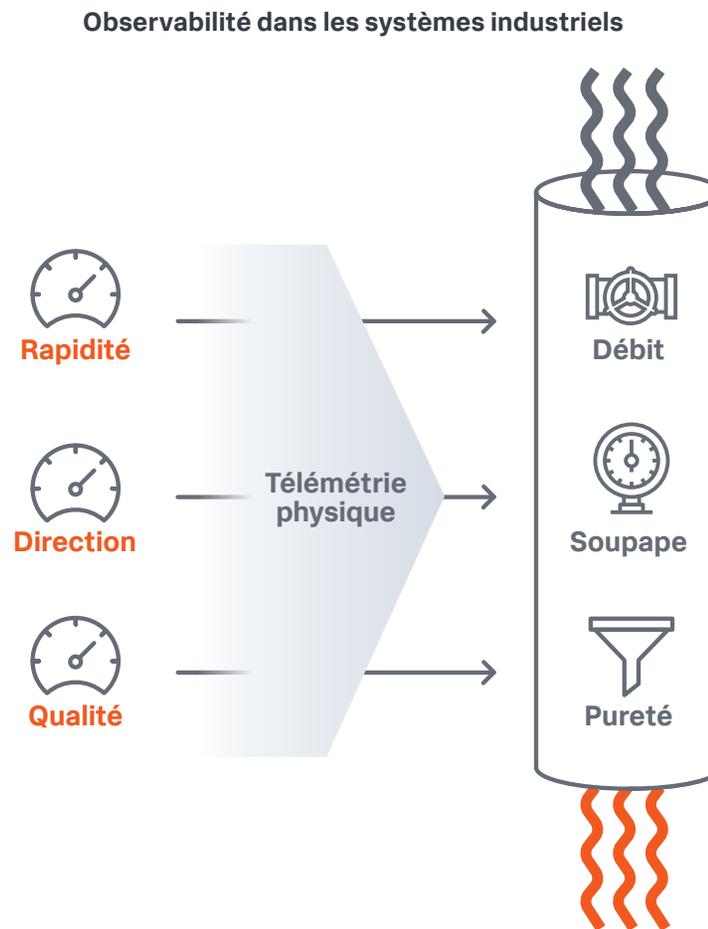
Pour les entreprises plus avancées qui utilisent l'analyse de données, le machine learning et l'intelligence artificielle, les données métier peuvent offrir des signes avant-coureurs d'échec en alertant sur des modifications, telles qu'une baisse des revenus par minute, un taux de rebond plus élevé que d'habitude, etc., pouvant être causées par des erreurs non détectées sur le site web.

La supervision des indicateurs commerciaux peut aussi permettre à une équipe d'obtenir des ressources supplémentaires pour son initiative DevOps en apportant des preuves de succès tangibles dans des termes compréhensibles par les interlocuteurs métier.

Observabilité, télémétrie et journalisation sémantique

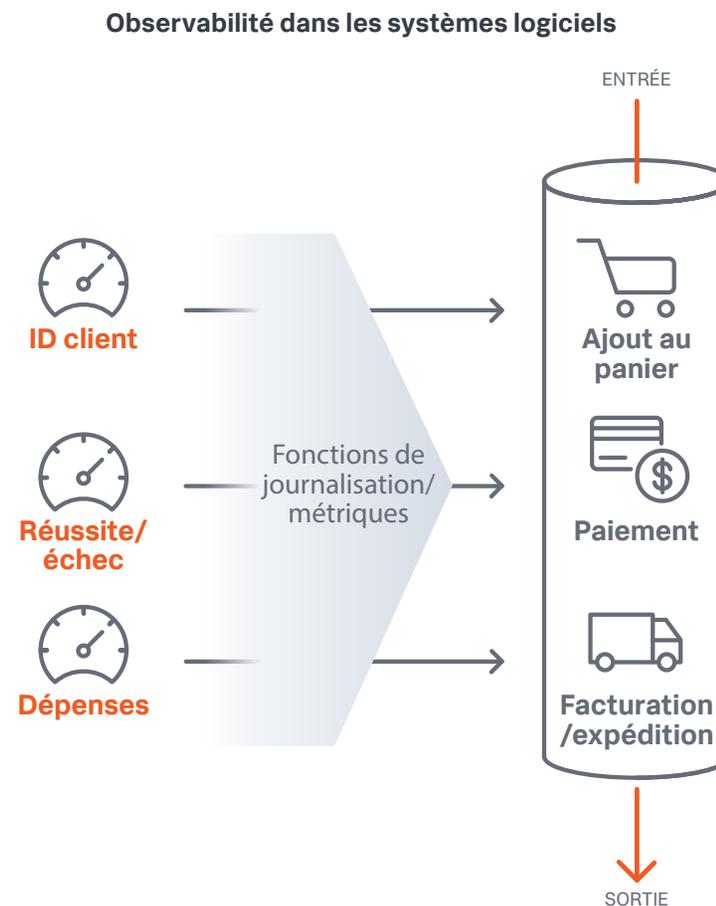
La majorité des entreprises les plus avancées déploient des systèmes de journalisation parallèlement aux applications et services afin de faciliter la supervision et la production d'alertes en aval, dans les opérations de production. Elles acquièrent ainsi une « observabilité », qui permet de répondre à des questions sur votre entreprise sans les avoir forcément anticipées.

Le terme « observabilité » était utilisé à l'origine dans les systèmes industriels ; elle peut être obtenue très simplement, par exemple en ajoutant une jauge de débit dans une conduite d'eau (mesure interne) et en la connectant à un affichage externe (télémétrie). Ce dispositif permet à un opérateur d'observer une propriété interne du système (la vitesse à laquelle l'eau s'écoule dans un tuyau) en observant un résultat externe (le débitmètre monté à l'extérieur de la conduite).



Dans une application, l'observabilité est souvent obtenue par l'ajout de métriques de performance, d'un traçage des appels de microservices et d'une journalisation sémantique pour décrire son activité réelle. Cela englobe des métriques techniques telles que le délai de réponse de bout en bout, et des indicateurs métier comme les revenus par transaction. Cela permet aux équipes de configurer leur supervision à l'aide de véritables indicateurs de succès, aussi bien technique que de métier, plutôt que de se contenter d'approximations.

L'observabilité permet aux équipes de travailler avec des architectures modulaires modernes comme le cloud, les microservices, les conteneurs, les API et les infrastructures serverless. En dépit de leurs nombreux avantages, ces architectures n'offrent souvent aucune visibilité sur les performances de l'infrastructure, aucune possibilité d'injecter du code d'instrumentation (avec une instrumentation native minimale) ni aucun moyen d'exécuter des transactions synthétiques. Dans ces systèmes complexes et opaques, l'émission de métriques, de traces et de logs pour intégrer l'observabilité dans le système dès la publication, confère un contrôle maximal aux équipes DevOps qui peuvent définir et configurer une supervision pertinente pour tous les scénarios à venir.



La collecte de mesures est un élément central du DevOps

La supervision autonome n'intéresse pas seulement les opérations ou les équipes DevOps fraîchement formées : elle est utile à toutes les équipes qui travaillent avec des technologies. L'adoption de la supervision autonome dans toutes les équipes souligne l'un des aspects essentiels du DevOps : nul besoin de créer une nouvelle équipe dotée de superpouvoirs, il vaut mieux donner de nouveaux moyens aux équipes en place afin qu'elles puissent collaborer de façon innovante.

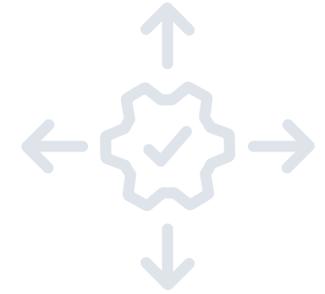
La supervision et les alertes en libre-service s'avèrent aussi accessibles qu'utiles pour :

- les développeurs qui supervisent leur propre code lorsqu'il s'exécute en production ;
- une équipe de développeurs qui collaborent avec leurs collègues des opérations pour livrer des applications administrables ;

- une équipe DevOps travaillant en tant que groupe autonome pour définir ses propres pratiques de supervision ;
- un groupe complexe d'équipes dans lequel des spécialistes des opérations assurent la supervision des applications livrées par les développeurs dans le cadre d'un système plus vaste.

Quelles que soient les circonstances, l'approche libre-service de la supervision et des alertes s'impose comme une contre-mesure face à l'isolement traditionnel du développement et des opérations. Il suffit d'ouvrir l'accès à ces indicateurs clés pour favoriser une culture du partage, créer des boucles de rétroaction et encourager l'apprentissage permanent au sein des équipes.

02



Réutiliser les motifs de déploiement pour élaborer des applications et des services

Notre étude met au jour une deuxième compétence de référence indispensable à la réussite de l'approche DevOps : la possibilité de réutiliser des routines de déploiement, des processus, des systèmes et des outils pour élaborer des applications ou même des services de bout en bout.

La réutilisation du code est essentielle à une ingénierie logicielle de haute qualité et cela s'applique également au code de déploiement et au code de supervision. La supervision en tant que code et l'infrastructure en tant que code accélèrent les cycles de livraison, réduisent les erreurs et permettent aux développeurs comme aux opérations de travailler de leur mieux.

Une récente étude menée par Puppet a demandé « À quelle fréquence avez-vous appliqué ces pratiques après avoir commencé à enregistrer de bons résultats avec le DevOps ? » Et voici la répartition des réponses pour la pratique « Nous réutilisons des motifs de déploiement pour élaborer des applications et des services ».

	FAIBLE	MOYENNE	ÉLEVÉE
TOUJOURS	2 %	14 %	46 %
LA PLUPART DU TEMPS	7 %	44 %	47 %
PARFOIS	34 %	33 %	6 %
RAREMENT	40 %	8 %	1 %
JAMAIS	18 %	1 %	—

Nous avons demandé : « À quelle fréquence avez-vous appliqué ces pratiques lorsque vous élargissez votre approche DevOps ? » Voici la répartition des réponses pour la pratique « Nous réutilisons des motifs de déploiement pour élaborer des applications et des services ».

	FAIBLE	MOYENNE	ÉLEVÉE
TOUJOURS	—	11 %	52 %
LA PLUPART DU TEMPS	7 %	45 %	47 %
PARFOIS	30 %	35 %	1 %
RAREMENT	45 %	8 %	—
JAMAIS	18 %	1 %	—

Nous avons posé la question de la réutilisation des motifs de déploiement en raison de la nature spéciale du déploiement des applications dans la plupart des entreprises, sinon toutes. Situé à la frontière entre développement et production, le déploiement d'applications est le point de rencontre le plus fréquent des développeurs et des opérations, et bien souvent aussi un point de conflit. L'amélioration du déploiement d'applications est donc au cœur du DevOps qui vise à faire tomber le « mur de confusion »³ à l'intersection entre Dev et Ops.

L'emploi de motifs réutilisables, qu'ils soient créés en interne ou qu'ils proviennent d'une source extérieure, fait bien plus qu'alléger la pénibilité et la confusion propres au déploiement. C'est aussi un moyen de partager et de diffuser le savoir-faire du déploiement dans l'entreprise, et donc de permettre à de nouvelles équipes et personnes de travailler ensemble sur ce qui doit devenir une compétence fondamentale dans n'importe quelle entreprise.

3 <http://dev2ops.org/2010/02/what-is-devops/>

03



Réutiliser les motifs de test pour élaborer des applications et des services

L'automatisation des tests logiciels est aussi importante, sinon plus, que celle du déploiement et de la supervision. Des modifications mineures peuvent interagir avec d'autres services de manière imprévisible, et en l'absence de système de test robuste, vous pourriez passer à côté.

Quelques considérations pour bien hiérarchiser cette pratique :

- **Si vos équipes de qualité sont bien séparées des équipes de développement et d'opérations, il est préférable d'attendre et de les intégrer ultérieurement à l'initiative DevOps.** Commencez par établir de solides routines de test avec votre propre équipe. Pour l'équipe des opérations, cela peut consister à mettre en place un processus pour tester les modifications d'infrastructure avant le déploiement en production. Les équipes de développement, quant à elles, pourront mettre en place une méthodologie de développement basée sur les tests (test-driven development ou TDD) ou autre dans le cadre d'un workflow agile.
- **Les activités les plus proches de la production comme le provisionnement, la supervision, les alertes, etc. sont souvent plus prioritaires pour les équipes car c'est souvent là que le plus de problèmes se manifestent.** Commencez par résoudre les problèmes propres au déploiement pour libérer du temps que vous pourrez ensuite consacrer à l'amélioration de vos pratiques de test.
- **Les motifs de test peuvent être moins réutilisables que les motifs de déploiement** dans la mesure où les tests traitent des aspects spécifiques d'une application ou d'un service et couvrent de nombreux processus différents (tests de fumée, tests unitaires, tests fonctionnels, tests de conformité, tests de complexité) dans des environnements statiques (boîte blanche) ou dynamiques (boîte noire).
- **Les tests en production sont souvent plus délicats et complexes qu'en pré-production,** car leurs objectifs ne sont pas les mêmes. Avec la livraison en continu, les équipes peuvent réaliser des expériences en production pour tester de nouvelles idées (déploiement bleu/vert, test du canari). C'est très utile mais ça ne remplace pas un test en production qui se concentre sur l'expérience client, la qualité, la fonctionnalité, la résilience et la stabilité, entre autres.
- **L'adoption de processus de test réutilisables est une pratique fondamentale, mais elle est souvent reportée à un stade ultérieur du parcours d'évolution,** une fois les motifs de déploiement bien établis. Si vous devez faire un choix, nous recommandons d'attendre pour les motifs de test et de donner la priorité aux autres pratiques. Toutefois, une fois cette pratique adoptée, il est important de veiller à ce que les motifs de test puissent être partagés. Par exemple, vous pouvez encoder des tests réutilisables dans des outils de test automatisés et donner l'accès à ces outils, ainsi qu'aux rapports et tableaux de bord qui en résultent, à toutes les parties concernées.

04



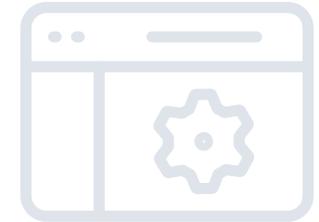
Les équipes apportent des améliorations aux outils fournis par d'autres équipes

La possibilité d'apporter des améliorations aux outils fournis par d'autres équipes s'avère fondamentale.

Lorsqu'une équipe améliore un outil, c'est souvent à la main, de façon ponctuelle, et l'amélioration reste confinée dans le silo de l'équipe jusqu'à ce qu'un changement justifie son élargissement à d'autres. Ce changement peut être une évolution culturelle ou organisationnelle touchant toute une division, l'apparition de nouvelles sources de données interfonctionnelles comme la journalisation sémantique ou la mise en place au sein des équipes d'une nouvelle collaboration entre différentes fonctions, afin d'automatiser le provisionnement ou la publication, par exemple.

Comme l'amélioration des outils d'autres équipes nécessite d'abord que chaque équipe mette de l'ordre chez elle, nous pensons que l'adoption de cette pratique peut être reportée à un stade ultérieur avec les mêmes chances de succès.

05



Les configurations sont gérées à l'aide d'un outil de gestion dédié

La pratique consistant à gérer les configurations à l'aide d'un outil dédié prend vite racine lorsque les organisations commencent à recueillir les fruits de leur évolution DevOps.

Pour les pratiquants DevOps de longue date, il n'est pas surprenant que cette pratique émerge comme un facteur fondamental de succès. La gestion automatisée des configurations a été un moteur essentiel du DevOps pendant de nombreuses années, en particulier quand on commençait tout juste à envisager l'infrastructure comme du code, et le mouvement

DevOps s'est largement articulé autour des premiers innovateurs dans le domaine de la gestion automatisée des configurations. Le passage au cloud observé ces derniers temps renforce encore l'importance des systèmes de gestion des configurations et du déploiement pour gérer la complexité d'un déploiement moderne.

L'examen des premiers moteurs du DevOps explique l'importance de la gestion des configurations.

La consomérisation de l'informatique a stimulé une demande sans précédent en termes de vitesse et de service. Dans ce contexte, l'incapacité des développeurs à provisionner et configurer des ressources clés de développement et de test (sans parler de production) est devenue un frein important à la livraison rapide de logiciels de grande qualité répondant aux besoins des clients et des entreprises.

Parallèlement à cela, les équipes ITOps étaient tout aussi frustrées par l'impossibilité d'apporter des changements substantiels aux applications et services qu'elles administraient, alors que ces changements devaient les rendre plus stables, plus efficaces et plus faciles à exploiter.

L'émergence du cloud computing a donné aux développeurs les moyens de répondre rapidement à leurs propres besoins simplement en sortant une carte de crédit. Mais cette nouvelle capacité est aussi source de problèmes et de frustration pour les opérations. Les développeurs ayant recours au Shadow IT, les équipes des opérations se sont trouvées confrontées à une multitude d'outils aux configurations uniques, difficiles à gérer, extrêmement fragiles et sources d'erreurs.

Lorsque les développeurs et les opérateurs ont commencé à travailler ensemble pour résoudre les difficultés liées au provisionnement, il était naturel que de nombreuses initiatives DevOps commencent par automatiser la configuration et le provisionnement. C'est une solution gagnante pour les développeurs comme pour les opérateurs, car elle permet aux développeurs d'agir à la vitesse du marché tout en encapsulant et en codifiant des configurations éprouvées pour réduire la charge de travail des opérateurs.

Avec l'évolution et l'élargissement du DevOps, et comme les développeurs, dont la charge de travail a été allégée grâce à l'automatisation du provisionnement, adoptent de plus en plus la livraison continue, les équipes opérations sont soumises à une pression accrue pour le maintien de la disponibilité et de la performance en production. Des questions de contrôle et d'audit émergent également avec la maturation des processus d'une organisation. Par conséquent, l'automatisation de la configuration et du provisionnement devient tout aussi importante pour la production que pour le développement et le test. Atteindre la répétabilité grâce à la gestion des configurations assure un environnement de production stable, fiable et contrôlable, et rend également possible des capacités ultérieures (le libre-service, par exemple) qui deviennent de nouveaux objectifs pour l'initiative DevOps.

Enfin, l'importance de l'automatisation du provisionnement et de la configuration n'est pas seulement liée à la vitesse. C'est également une protection contre l'une des toutes premières craintes entourant le DevOps : que des développeurs autonomes puissent faire la loi dans les systèmes de production sans recours, sans audits et sans contrôle. La codification des processus éprouvés et leur exécution à l'aide de routines prédéfinies vient répondre à ces craintes sans empêcher les équipes Dev et Ops de gagner en agilité. Dans les grandes organisations et dans les secteurs réglementés tels que la santé, la finance et l'administration, l'automatisation offre l'avantage supplémentaire de consigner chaque intervention sur un système avec son auteur, son contenu, ses date et heure ainsi que, bien souvent, sa justification.

Mettre en place les pratiques fondamentales dans votre organisation

Devant toutes ces preuves que ces pratiques fondamentales conduisent au succès du DevOps, nous savons que nos lecteurs veulent avoir des conseils sur leur mise en œuvre. Heureusement, notre étude fournit plusieurs indices sur les pratiques à adopter en priorité.

En général, nos recherches suggèrent de commencer par établir des bases qui serviront de fondement à la croissance et au développement à venir, tout en contribuant à fournir une valeur immédiate. La pratique à implémenter en priorité dépend beaucoup de la situation de départ de l'entreprise et de ses besoins les plus urgents. Toutefois, sachez bien que toutes les pratiques fondamentales sont adoptées rapidement par les organisations avancées.

Trois pratiques étaient toujours activement utilisées par la majorité des organisations avancées au moment où leurs initiatives DevOps commençaient à donner des résultats :

- la réutilisation des motifs de déploiement ;
- l'utilisation d'un outil de gestion des configurations ;
- la possibilité pour les équipes de configurer la supervision et les alertes des services qu'elles administrent.

Ces pratiques sont toutes des compétences fondamentales et forment une base pour progresser vers des niveaux supérieurs d'évolution DevOps, mais la séquence de mise en œuvre n'a pas d'importance. Le plus important est de commencer par ces trois pratiques. L'une d'entre elle semblera sans doute prioritaire pour votre entreprise.

Notre enquête suggère fortement que les entreprises aient mis en place plusieurs bases avant de commencer à observer des résultats. Nous recommandons par conséquent de s'efforcer à adopter les trois pratiques citées précédemment, puis de les élargir à d'autres équipes dès que possible.

Les deux autres pratiques fondamentales sont celles que nous recommandons de mettre en œuvre une fois les trois premières bien établies :

- réutiliser les motifs de test pour élaborer des applications ou des services ;
- donner aux équipes les moyens d'apporter des améliorations aux outils des autres équipes

Mais... commencez là où vous en êtes

Chaque entreprise est unique. Vous devez commencer à votre point de départ. Identifiez la situation actuelle dans votre organisation et choisissez l'approche la plus à même d'avoir rapidement un impact positif là où vous en avez le plus besoin. Comme les pratiques fondamentales permettent d'améliorer la supervision, le contrôle des configurations, les motifs de déploiement et les contrôles qualité, mais aussi de collaborer sur les outils, le choix est vaste.

Établissez et partagez des valeurs de référence pour les métriques du système, des clients et des activités, afin que chacun sache d'où il part et ce qui doit être amélioré, et connaisse l'impact de son travail sur le système de bout en bout. Vous aiderez ainsi tous les membres de l'équipe à adopter des concepts DevOps comme le feedback continu et l'amélioration permanente.

Partagez dès que possible les pratiques et les processus pour que toute l'organisation progresse ensemble. Automatisez et élargissez horizontalement l'automatisation de la configuration et du provisionnement pour aider les développeurs à gagner en agilité, et les opérateurs à atteindre la prévisibilité. Les deux équipes seront ainsi plus enclines à croire dans la valeur de cette aventure. Faciliter le travail des gens est une excellente manière de gagner leur confiance et de s'assurer de leur coopération à long terme.

Le décalage en amont (shift left) consiste à accomplir une tâche plus tôt dans le processus qu'on ne le faisait traditionnellement. En déplaçant la planification de la configuration et du provisionnement au début du processus, vous vous assurez d'y consacrer toute l'attention qu'elle mérite.

La meilleure voie est celle qui donne des résultats rapides

Chaque entreprise est unique : commencez par ce qui compte le plus pour ceux que vous avez besoin d'impressionner. Choisissez une initiative qui produira des résultats à même de faire naître la confiance chez les décideurs de votre entreprise. Vous obtiendrez ainsi du temps, des ressources et du budget pour avancer et progresser sur la base de ces premiers succès.

Une fois lancé dans ce parcours DevOps, vous constaterez que vous avez besoin de meilleurs outils pour comprendre l'état de santé et la performance de votre entreprise. Splunk a une solution pour les applications modernes : Splunk Observability Cloud.

[En savoir plus](#)

À propos de l'auteur



Greg Leffler

Greg dirige l'équipe d'experts de l'observabilité de Splunk et s'est donné pour mission de répandre la bonne parole de l'observabilité dans le monde. La carrière de Greg l'a mené du NOC à la SRE, puis de la SRE à la direction de la SRE, avec quelques crochets par des fonctions de sécurité et de rédaction. Outre l'observabilité, les intérêts professionnels de Greg englobent le recrutement, la formation, la culture SRE et la gestion d'équipes distantes efficaces. Greg est titulaire d'un Master en psychologie industrielle/professionnelle de l'université Old Dominion.

splunk>

Splunk Inc. transforme les données en actions grâce à une technologie conçue pour étudier, superviser, analyser et agir sur les données à n'importe quelle échelle.

puppet

Puppet ouvre la voie vers un monde de progrès logiciel sans limites. Sa plateforme révolutionnaire est devenue la norme de l'industrie pour l'automatisation de la livraison et de l'administration des logiciels qui font tourner le monde. Plus de 40 000 entreprises, dont plus de 75 % des Fortune 100, utilisent les solutions open source et commerciales de Puppet pour adopter les pratiques DevOps, acquérir une connaissance parfaite de leur situation et mettre en œuvre des changements logiciels en toute confiance. Basée à Portland dans l'Oregon, Puppet est une entreprise privée comptant plus de 500 employés dans le monde. En savoir plus sur puppet.com.

En savoir plus

Splunk, Splunk> et Turn Data Into Doing sont des marques commerciales de Splunk Inc., déposées aux États-Unis et dans d'autres pays. Tous les autres noms de marque, noms de produits et marques commerciales appartiennent à leurs propriétaires respectifs. © 2022 Splunk Inc. Tous droits réservés.

22-22512-Splunk-The 5 Foundational DevOps Practices-LS-116



splunk>
turn data into doing®