



DevOps pour la transformation numérique

Comment DevOps peut transformer
L'informatique et l'entreprise



Qu'est-ce qu'il y a à l'intérieur

CHAPITRE 1

03 Présentation

CHAPITRE 2

04 Apprendre à connaître DevOps et SRE

CHAPITRE 3

06 Fondamentaux du DevOps

CHAPITRE 4

dix Bonnes pratiques DevOps

CHAPITRE 5

15 avantages de DevOps

CHAPITRE 6

19 Les défis du DevOps

CHAPITRE 7

22 Métriques DevOps

CHAPITRE 8

26 DevOps et observabilité intelligente permettre la transformation numérique



CHAPITRE 1

Introduction

De plus en plus d'organisations ont adopté les pratiques DevOps pour rationaliser le développement de logiciels, augmenter la productivité des développeurs et améliorer les flux de travail de livraison continue pour offrir une meilleure logiciel plus rapidement.

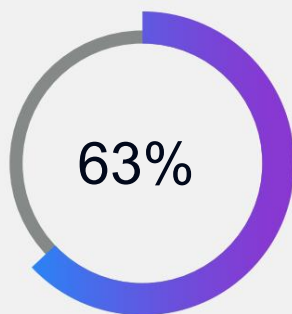
En tant que pionnier du DevOps [Patrick Debois](#) noté en 2009, les tactiques - pas seulement les solutions technologiques - définissent une approche réussie de DevOps qui peut transformer fondamentalement l'informatique. Mais bien que cette orientation tactique offre une flexibilité accrue aux équipes, elle peut rapidement conduire à des silos de données et de communication dans toute l'organisation, ce qui peut nuire à la qualité des logiciels et à la rapidité de livraison. Sans aide, il peut être extrêmement difficile d'obtenir des informations stratégiques sur la façon dont les équipes de développement exécutent leurs tâches quotidiennes, comment automatiser les pipelines DevOps et comment concevoir un logiciel pour la fiabilité et la résilience dans environnements cloud natifs modernes.

Cette aide se présente sous la forme d'intelligence artificielle (IA). AIOps, la discipline consistant à appliquer l'IA et l'analyse avancée aux opérations informatiques, a transformé la façon dont les organisations gèrent des systèmes complexes. En utilisant ces mêmes principes, les organisations peuvent adopter une approche plus intelligente de DevOps - une approche AIOps de DevOps qui tire parti de l'IA tout au long du cycle de vie du développement logiciel (SDLC). DevOps, associé à des technologies et tactiques complémentaires, telles que l'ingénierie de la fiabilité du site (SRE), a le potentiel de

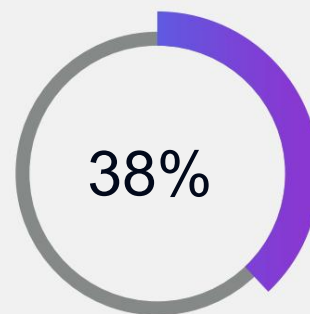
transformer l'entreprise.

Pour mieux comprendre le pouvoir transformateur de DevOps, nous explorerons les bases de DevOps et le rôle croissant de SRE ; plonger dans les principaux avantages et défis de DevOps ; discuter des meilleures pratiques DevOps et des clés métriques DevOps ; et examinez comment l'IA et l'automatisation à chaque étape du cycle de vie DevOps peuvent transformer la façon dont les organisations se développent et offrent de meilleures performances logiciel plus rapidement.

DevOps a le pouvoir potentiel de remodeler votre entreprise en rationalisant votre informatique pour offrir une meilleure valeur :1

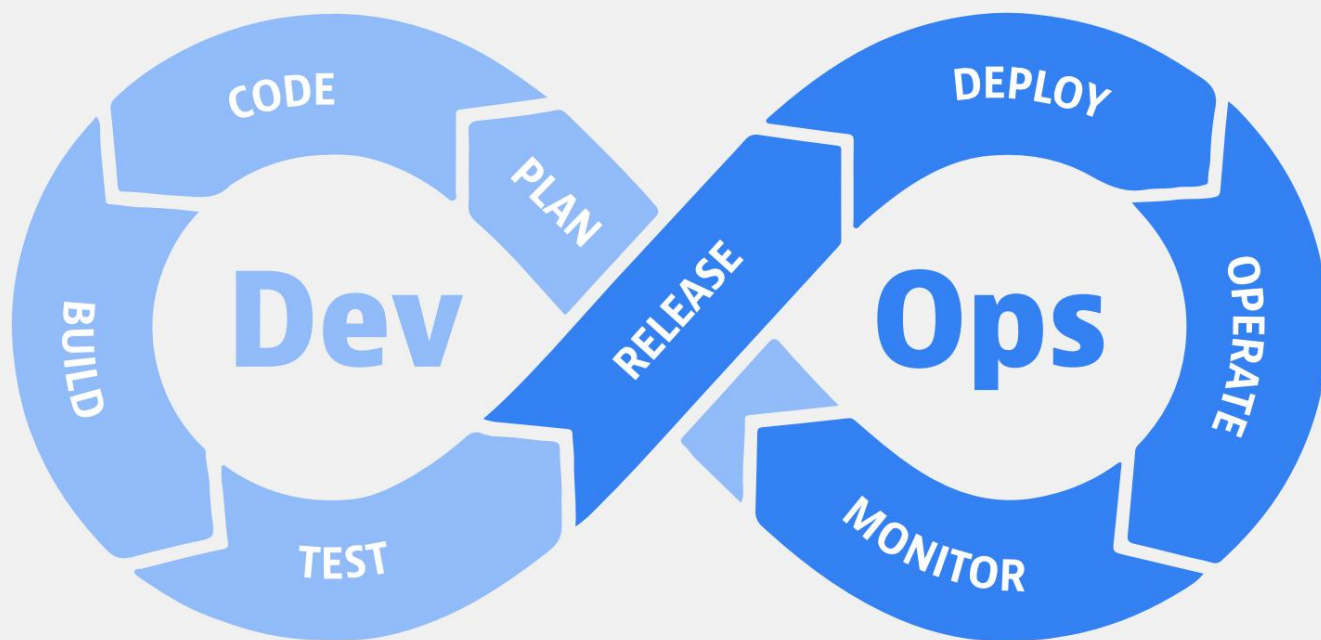


des organisations qui utilisent les pratiques DevOps ont amélioré la qualité de leurs déploiements.



des organisations signalent une amélioration globale de la qualité du code.

¹Statistiques DevOps pour les sceptiques. UpGuard. <https://www.upguard.com/blog/devops-success-stats#toc-1>. Août 2021.



CHAPITRE 2

Apprendre à connaître DevOps et SRE

Qu'est-ce que DevOps ? Le développement rencontre les opérations informatiques

DevOps est un cadre flexible de pratiques de développement logiciel que les organisations utilisent pour créer et fournir des logiciels en alignant et en coordonnant les efforts de développement logiciel - "Dev" - avec les opérations informatiques - "Ops".

La façon la plus simple de conceptualiser DevOps est de boucle continue. Au lieu de processus discrets, le développement et les opérations font partie d'un cycle continu qui comprend la planification, le codage, la création, les tests, la publication, le déploiement, l'exploitation et la surveillance des applications et des services.

Cette approche de flux de travail continu permet aux équipes d'identifier et de résoudre immédiatement les problèmes liés à la forme et à la fonction plus tôt dans le processus afin d'éviter les problèmes avant que le logiciel ne soit mis en production.

Le développement récent des technologies cloud-native, les solutions open source et les API flexibles ont encore amélioré l'efficacité DevOps. Avec ses racines dans Agile développement, DevOps est parfaitement adapté pour aider les équipes suivre le rythme de l'accélération du développement et modèles de version, tels que l'intégration continue et livraison continue (CI/CD).

Qu'est-ce que le SRE ? Résilience logicielle intégrée

SRE est une pratique d'exploitation logicielle qui gère les détails et les préoccupations globales de la résilience logicielle pour garantir la disponibilité, la latence, les performances et la capacité des systèmes logiciels. Les ingénieurs en fiabilité du site comprennent les besoins des systèmes logiciels et mettent en place des processus et des structures pour répondre à ces besoins.

Le vice-président de l'ingénierie de Google, Ben Sloss, a inventé le terme SRE en 2003 lorsque lui et son équipe ont commencé à appliquer les principes d'ingénierie logicielle aux opérations logicielles pour créer des systèmes logiciels plus fiables et évolutifs. La mise en œuvre de SRE peut aider les organisations à réduire les frictions entre les composants de développement et d'exploitation des équipes DevOps, en rationalisant l'efficacité et en réduisant les taux d'erreur.

SRE complète les pratiques DevOps en offrant l'automatisation accrue pour réduire la dépendance au manuel. Ces pratiques aident les équipes de développement à résoudre leurs problèmes et à fournir une fiabilité dès la conception plus tôt dans le processus de développement.

En fin de compte, SRE aide les organisations à atteindre leurs objectifs opérationnels, tels que la réduction des temps d'arrêt ou des résolutions plus rapides, en définissant et en automatisant les objectifs de niveau de service (SLO).

Comment SRE et DevOps interagissent-ils ?

SRE et DevOps sont essentiellement les deux faces d'une même médaille. Alors que les cadres DevOps se concentrent sur la collaboration tout au long du cycle de vie et sur la suppression des silos, un SRE robuste aide à mettre en œuvre et à automatiser les pratiques DevOps à l'aide de SLO et garantit que ces systèmes et les logiciels qu'ils produisent sont résilients.

Selon [Andi Grabner](#), DevOps Activist chez Dynatrace, "DevOps et SRE sont un équilibre entre rapidité et sécurité." Alors que DevOps aide les organisations à se déplacer de gauche à droite tout au long du cycle de vie du développement et des opérations pour augmenter la vitesse globale, SRE se déplace de droite à gauche pour aider à réduire les taux d'échec plus tôt dans le cycle de développement.

Bien qu'il soit possible d'avoir DevOps sans le SRE, ces deux processus fonctionnent mieux ensemble, créant ainsi un cycle continu qui offre des améliorations continues dans les deux directions du pipeline CI/CD. En tant que pratique combinée, les entreprises peuvent chercher à accroître l'automatisation, à accélérer la livraison, à améliorer la qualité des logiciels et bien plus encore.

Avec ces informations de base en main, plongeons maintenant dans les composants fondamentaux de l'état d'esprit DevOps pour avoir une meilleure idée de l'endroit où votre équipe devra concentrer ses efforts pour tirer le meilleur parti des pratiques DevOps.



Les SRE sont entrés en pratique pour accroître la résilience des organisations grâce à l'automatisation de nombreuses tâches manuelles au début du développement.



CHAPITRE 3

Fondamentaux de DevOps

DevOps est un changement culturel qui nécessite une vision, une planification, l'adhésion de la direction et une collaboration étroite pour établir avec succès une manière plus intégrée de développer et de fournir des applications. En adoptant quelques pratiques fondamentales, les équipes peuvent améliorer leur efficacité et développer une compréhension plus approfondie de leurs workflows, ensembles d'outils et processus afin de pouvoir publier de meilleurs logiciels plus rapidement.

Étant donné que DevOps est un continuum, ces pratiques doivent également être continues et continues. Ce chapitre couvre les principes ou pratiques de base qui constituent les principes fondamentaux de l'adoption d'une approche DevOps.

Intégration continue

L'intégration continue (CI) est une pratique de développement logiciel dans laquelle les développeurs valident régulièrement leur code dans un référentiel partagé. Comme l'architecture des microservices est distribuée, CI permet aux développeurs de posséder des morceaux de code discrets et gérables, ainsi que des fonctionnalités individuelles, et de travailler dessus en parallèle.

La nature distribuée de ces applications permet des mises à jour fréquentes, souvent plusieurs fois par jour. Cependant, les développeurs ne peuvent pas simplement pousser les mises à jour de build au hasard.

CI est étroitement contrôlé; les nouveaux commits déclenchent la création de nouvelles versions de test via

le système de gestion de construction. Le code redondant est rejeté et les changements avec rupture sont minimisés une fois les branches principales modifiées. Incrémentale les changements sont encouragés. De plus, une réconciliation réduite empêche les gels de code obligatoires qui découlent généralement de conflits. Dans l'ensemble, l'intégration continue permet aux équipes de créer et de tester des logiciels plus rapidement et plus efficacement. En fusionnant régulièrement le code, les équipes disposent également toujours d'une version à jour qui accélère les tests et la correction des bogues, renforce la confiance de la fusion et contribue à raccourcir le pipeline de développement.

Livraison continue

Alors que CI se concentre sur des mises à jour de code régulières et indépendantes vers un référentiel central, la livraison continue (CD) se concentre sur la publication de blocs de code terminés vers un référentiel à intervalles réguliers. Ces blocs de code doivent toujours être dans un état déployable pour les tests ou la mise en production. Le CD est souvent confondu avec le déploiement continu - le processus suivant en ligne - qui libère le code finalisé en production.

Le déploiement consiste à mettre à la disposition des utilisateurs finaux des logiciels nouveaux et mis à jour. Par conséquent, le CD désigne principalement la « livraison continue », ou les deux « livraison et déploiement continus », mais rarement uniquement le déploiement continu. Le CD prend le code et l'ajoute à un référentiel, tel que GitHub ou, dans le cas d'un environnement basé sur des microservices, un registre de conteneurs. L'objectif final est d'augmenter la cohérence des versions en gardant perpétuellement le code dans un état déployable. Le développement de logiciels devient ainsi plus agile et plus prévisible.

Tests et validation en continu

Les tests continus dans DevOps sont importants à chaque étape du SDLC. Il implique de nombreuses parties prenantes, notamment l'équipe de développement, l'assurance qualité et le personnel opérationnel. L'objectif des tests continus est d'évaluer la qualité du logiciel à mesure qu'il progresse à chaque étape du cycle de vie de la livraison. Ce

non seulement arrête le mauvais codage dans son élan, mais fournit également un retour rapide et continu aux équipes de développement avec les informations dont elles ont besoin pour résoudre tout problème de qualité.

Alors que les tests continus sont importants pour évoluer, la validation manuelle des résultats des tests fait dérailler le processus de livraison du logiciel. C'est là qu'intervient la validation continue, en automatisant le processus d'évaluation des résultats des tests par rapport à vos objectifs de niveau de service prédéfinis. La validation continue complète la mise en œuvre des tests continus en éliminant toute analyse manuelle requise, qu'il s'agisse de comparer des données sur des tableaux de bord ou de cocher des cases sur une feuille de calcul. Au lieu de cela, les équipes DevOps peuvent mettre en place des techniques telles que des barrières de qualité qui appliquent automatiquement des critères de qualité prédéfinis et empêchent le mauvais code de passer à l'étape suivante.



Surveillance continue et observabilité

Bien que les organisations s'efforcent d'avoir des processus CI/CD hermétiques, il existe souvent des possibilités d'amélioration. La surveillance et l'observabilité sont essentielles pour comprendre la viabilité du code à mesure qu'il progresse dans le pipeline. Bien que la détection des problèmes et des vulnérabilités soit toujours importante, la quantité considérable de données d'observabilité associées aux applications multicloud modernes crée signifie qu'il n'y a tout simplement aucun moyen de suivre manuellement tout ce qui se passe à travers la pile logicielle.

Traditionnellement, la surveillance DevOps était étroitement associée aux équipes « ops », mais a depuis évolué tout au long du cycle de vie complet du développement logiciel (SDLC), car les principales parties prenantes ont de plus en plus besoin de réponses pour résoudre leurs problèmes plus rapidement. Ces réponses sont possibles lorsque vous disposez d'un système qui surveille et analyse en permanence les données d'observabilité. La capture continue des données peut être valorisante lorsqu'elle est exploitée intelligemment, et c'est là que l'introduction de l'observabilité est essentielle pour DevOps. L'observabilité ne se limite pas à collecter des métriques et à les organiser dans des tableaux de bord. Avoir un moteur d'IA qui fonctionne 24h/24, 365 jours par an pour analyser les données et fournir des réponses aux anomalies et résoudre les problèmes plus rapidement et améliorer la version des décisions. Cela entraîne une meilleure qualité de code, de meilleures performances d'application, ce qui se traduit par de meilleures expériences pour l'utilisateur final. À mesure que la complexité des logiciels augmente, cela devient de plus en plus difficile pour les équipes DevOps pour fournir de nouvelles fonctionnalités et versions plus rapidement sans sacrifier la qualité. Par conséquent, il est essentiel de doter vos équipes d'une observabilité continue et d'un moteur d'IA pour analyser toutes les données et apporter des réponses pour le succès.

Sécurité continue

Une autre pratique DevOps fondamentale est [la sécurité continue](#) basé sur les tests, la surveillance, l'autorisation et le suivi des stocks. C'est l'évolution vers DevSecOps. En termes simples, la sécurité continue consiste à intégrer la sécurité au processus CI/CD, couvrant l'intégralité du SDLC, en ajoutant une couche supplémentaire au processus DevOps et aux pipelines pour garantir que votre infrastructure et vos applications ne ont des vulnérabilités et des risques qui leur sont associés.

Aujourd'hui, alors que nous voyons les environnements devenir de plus en plus complexes, une approche « bouloignée » de la sécurité n'est ni évolutive ni durable, et il est donc impératif d'intégrer la sécurité dans vos processus automatisés aux problèmes aide les équipes pour permettre des tests continus tout au long de votre cycle de vie de développement.

Semblable à la mentalité de gauche en ce qui concerne la qualité des tests, les mesures de sécurité doivent être intégrées à la planification et à la création dès le premier jour, et elles doivent se produire constamment tout au long du cycle de vie du développement, y compris lorsque le logiciel est exécuté en production. En outre, toute mesure de sécurité continue mise en œuvre doit être automatisée, le cas échéant, afin de ne pas entraver l'efficacité. En terme

DevOps est un changement culturel

Cette nouvelle culture nécessite l'adoption de bonnes pratiques telles que :



Continu
livraison



Continu
l'intégration



Continu
essai



Continu
surveillance/
observabilité



Inter-équipe
collaboration

de culture, le personnel de sécurité doit être considéré comme un partenaire à part entière dans le processus DevOps, au même titre que les développeurs et les spécialistes des opérations - d'où le virage vers DevSecOps.

Collaboration inter-équipes

La suppression des silos est primordiale pour assurer une bonne communication et une unification à travers le pipeline DevOps. Une exécution DevOps efficace signifie établir une source unique de vérité, c'est-à-dire agréger les données de nombreuses sources dans un emplacement collectif.

Les testeurs, les ingénieurs, l'assurance qualité et même les parties prenantes non techniques peuvent obtenir des informations précieuses à partir de ces informations. Selon ce paradigme, chacun contribuera à sa manière à la création de logiciels qui génèrent des résultats commerciaux de haut niveau.

Une coopération transparente entre les développeurs et les équipes opérationnelles est particulièrement critique. Les processus DevOps coexistent dans un cycle continu appelé [boucle de rétroaction](#).

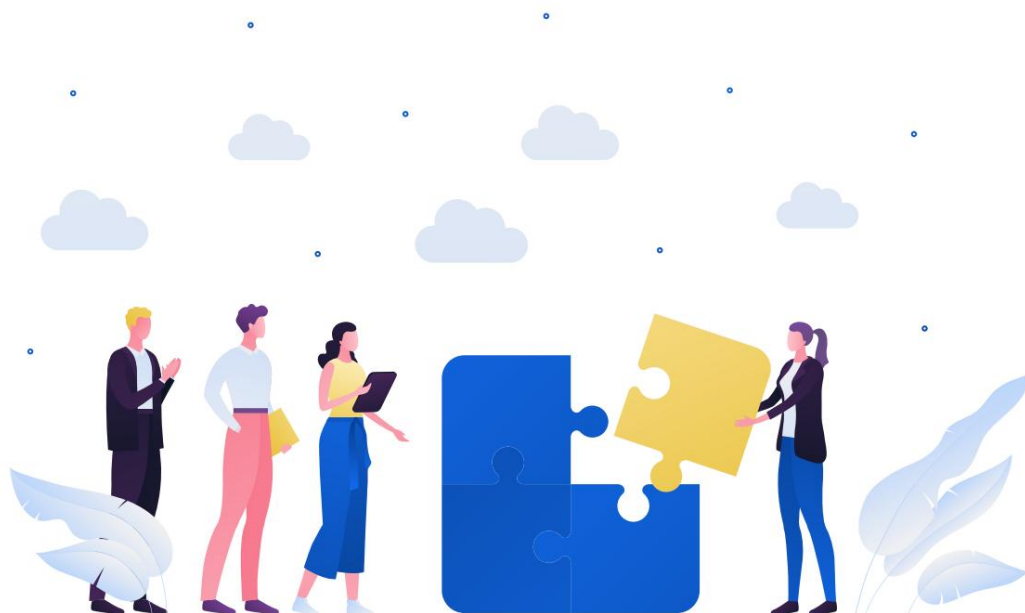
Différentes parties d'un projet sont terminées et examinées par les parties prenantes, et des commentaires sont renvoyés à partir de ces étapes. Le code doit être écrit, testé, validé, livré, construit et finalement déployé

pour les utilisateurs finaux.

La collaboration inter-équipes bénéficie de l'accélération de ce cycle. En conséquence, l'automatisation est devenue un ingrédient clé pour raccourcir le cycle de bout en bout, principalement en réduisant les frictions causées par plusieurs parties travaillant en même temps.

Faire passer DevOps au niveau supérieur

Bien que nous ayons établi les fondamentaux, les organisations ne veulent pas se contenter des fondamentaux, elles veulent faire passer leur DevOps au niveau supérieur. Pour suivre le rythme de l'innovation et la nécessité de fournir des services plus rapidement sur le marché, les organisations doivent faire passer leurs pratiques DevOps au niveau supérieur. Faire évoluer ces pratiques fondamentales vers des DevOps performants d'élite nécessite certaines bonnes pratiques, que nous aborderons dans le chapitre suivant.





CHAPITRE 4

Bonnes pratiques DevOps

De nombreuses organisations affirment disposer d'un processus DevOps entièrement fonctionnel. Mais DevOps est plus qu'un simple flux de travail et quelques outils que votre organisation peut mettre en œuvre et passer à autre chose. Il est utile de le considérer comme une philosophie - une culture et un état d'esprit - qui nécessite une optimisation, une créativité et une flexibilité continues à maintenir. Peut-être qu'une organisation a mis en œuvre des changements organisationnels et des outils qui jettent les bases d'un bon processus DevOps, mais elle peut manquer certains des avantages que DevOps peut offrir.

Avec ce parcours d'amélioration à l'esprit, explorons quelques meilleures pratiques DevOps qui peuvent faire passer votre investissement dans les fondamentaux au niveau supérieur et vous assurer que vous tirez le meilleur parti de votre stratégie DevOps.



Automatisation

L'automatisation est la pierre angulaire de la stratégie DevOps de chaque entreprise. En bref, l'automatisation réduit la charge de travail, vous aide à accélérer vos pipelines de livraison sur l'ensemble du SDLC et vous permet de faire évoluer votre pratique DevOps.

Traditionnellement, les processus tels que les tests, la surveillance, la découverte d'erreurs et la correction comprenaient un peu d'automatisation et beaucoup d'interventions manuelles. Ce n'a fonctionné lorsque de petites équipes travaillaient sur des applications monolithiques. Mais avec les microservices modernes

applications et la transformation numérique mettant encore plus de pression sur l'informatique, l'automatisation est cruciale pour augmenter la vitesse et la qualité en pilotant des processus cohérents à chaque étape du cycle de vie DevOps.

En conséquence, vous pouvez pousser le code en production plus fréquemment et produire des logiciels cohérents, fiables et sécurisés tout en faisant gagner à votre équipe DevOps un temps précieux qu'elle peut consacrer à l'innovation.



Surveillance et observabilité

Le suivi et l'observabilité sont essentiels pour intégrer à chaque étape du logiciel cycle de développement, de la pré-production à production. Tout en automatisant autant de processus que possible augmente l'efficacité de votre DevOps les flux de travail, le suivi et l'observabilité fournissent vos équipes avec une visibilité sur celles automatisées processus pour détecter et identifier les causes profondes de tout problème ou goulot d'étranglement.

De nombreux outils fournissent des données et des tableaux de bord pour suivre la santé des systèmes individuels. Mais pour développer une stratégie d'observabilité efficace qui donne des réponses exploitables sur les systèmes tout au long de la chaîne d'outils DevOps, vous avez besoin de plus que de simples données sur des tableaux de bord - vous avez besoin d'une approche intelligente.

1. Rendez vos systèmes observables — adoptez une norme, comme OpenTelemetry. Effet de levier une plate-forme d'observabilité basée sur l'IA qui peut instrumenter et détecter automatiquement les anomalies, vous n'avez pas à le faire manuellement.
2. Établir une observabilité de bout en bout à partir de la pré-production à la production pour chaque application ou environnement.
3. Assurez-vous de pouvoir comprendre l'impact commercial d'un événement ou d'une transaction en l'analysant dans son contexte des processus en amont et en aval de celui-ci.

4. Tirez parti de l'IA pour détecter automatiquement les problèmes et fournir une analyse pour identifier immédiatement les causes profondes et déclencher une correction automatique.

Il s'agit d'une pratique importante à mettre en œuvre afin que votre équipe puisse identifier les pannes ou les problèmes de performances avant que vos clients n'en ressentent l'impact.



IAOps

Les données peuvent être le meilleur ami d'une équipe informatique, en particulier lorsqu'il s'agit de tester et de fournir plus efficacement du code et des services de surveillance. Cependant, le traitement de la quantité massive de données créées par les applications d'aujourd'hui dépasse les capacités des humains seuls. Cela ouvre la voie à un moteur d'IA capable d'analyser en permanence toutes les données observables jusqu'au détail au niveau du code, et qui donne à l'équipe de développement le pouvoir d'identifier les problèmes, d'obtenir des réponses et de résoudre rapidement les problèmes lorsqu'ils surviennent.

L'exploitation de l'IA dans le cadre de vos processus DevOps vous permet d'améliorer les fonctionnalités et l'automatisation des cycles de développement, de test, de sécurité, de livraison et de publication, ainsi que de surveiller en permanence les performances des logiciels déployés de manière beaucoup plus efficace qu'en utilisant des efforts manuels.



Qualité Shift-gauche

Les SRE vivent et respirent les objectifs de niveau de service (SLO). S'assurer que les niveaux de service de production sont sur la bonne voie nécessite une évaluation continue des indicateurs de niveau de service (SLI) par rapport aux SLO. Mais cela soulève la question suivante : pourquoi les développeurs ne devraient-ils pas s'assurer que le code qu'ils créent respecte les mêmes SLO de production ? Ce concept de déplacement vers la gauche améliore la qualité du logiciel, aide à détecter les problèmes beaucoup plus tôt dans le cycle de vie et

empêche le code qui ne respecte pas les SLO de production de passer à l'étape suivante. Les résultats sont moins de violations de SLO en termes de production, de temps et d'argent économisé en raison de moins ou pas de salles de guerre, mais plus surtout, s'assurer que 100 % des accords de niveau de service (SLA) sont respectés.

Une façon d'automatiser ce processus de décalage vers la gauche consiste à utiliser des portes de qualité, qui vous permettent de comparer automatiquement les SLI de n'importe quel outil de pipeline (tels que la surveillance et les tests) par rapport aux SLO prédéfinis. Si le code ne passe pas la porte de qualité basée sur le SLO, il ne peut pas passer à l'étape suivante et le système notifie automatiquement l'équipe de développement pour résoudre le problème.



Fiabilité du décalage vers la droite

La livraison progressive (également appelée décalage vers la droite) se concentre sur l'expansion des pratiques CI/CD globales pour aider à fournir des applications et des services avec plus de contrôle.

Il permet aux organisations de gérer avec précision comment et quand les nouvelles fonctionnalités, mises à jour et correctifs sont fournis afin de minimiser l'impact négatif potentiel sur la base d'utilisateurs. Certaines pratiques courantes incluent les déploiements bleu-vert, les tests A/B, les déploiements Canary et les indicateurs de fonctionnalité.

• Déploiements bleu-vert

Ce modèle de version d'application fait passer progressivement les utilisateurs d'une version actuelle d'une application ou service (la version "bleue") à une nouvelle version (la version "verte") tandis que le bleu et le vert sont en cours de production. Ce changement devrait sembler transparent pour l'utilisateur, et le bleu peut se tenir prêt au cas où un problème imprévu avec le vert nécessiterait une restauration à la version antérieure, plus stable.

• Tests A/B

Également connu sous le nom de test fractionné, le test A/B fait référence à des processus d'expérimentation aléatoires dans lesquels deux versions ou plus d'une variable (par exemple, un service, une page Web ou un élément de page) sont présentées à différents utilisateurs finaux. À partir de là, vous pouvez surveiller les performances de l'application ainsi que le comportement et la satisfaction des utilisateurs pour déterminer quelle option est la meilleure pour les objectifs commerciaux.

• Indicateurs de fonctionnalités

Également connus sous le nom de bascules, les indicateurs de fonctionnalité sont une pratique de développement qui permet aux équipes logicielles et de développement d'activer et de désactiver des parties d'une base de code avec un simple commutateur (ou indicateur). Les indicateurs de fonctionnalité aident les organisations à dissocier les déploiements de code des versions de fonctionnalités, leur permettant d'apporter des modifications de code en production qui restent cachées aux utilisateurs jusqu'à ce qu'elles soient activées. Cela se traduit par des vitesses de déploiement accrues, une meilleure stabilité du système et une meilleure collaboration inter-équipes.

• Déploiements Canary

Tous les déploiements en production comportent des risques, même avec une surveillance et des tests complets. Une méthode permettant aux développeurs d'atténuer les perturbations graves consiste à déployer Canary. Le terme provient de l'époque où les canaris étaient utilisés pour détecter les gaz toxiques dans les mines de charbon. Si le canari mourait, les mineurs sauraient sortir avant que le gaz ne les atteigne. Un déploiement Canary est une version d'un logiciel qui est déployée sur un petit pourcentage (appelé Canary) de l'ensemble de la base d'utilisateurs. Si tout fonctionne bien dans votre Canary, vous pouvez ensuite déployer la version sur le reste de la base d'utilisateurs. Si les choses ne fonctionnent pas bien, au moins l'impact est beaucoup plus faible, moins perturbateur, et vous pouvez restaurer le logiciel.

Les déploiements Canary offrent la possibilité de tester des utilisateurs réels, qui peuvent fournir de véritables commentaires, tout en réduisant les risques en atténuant l'impact si des problèmes conduisent à un produit de meilleure qualité.



Sécurité Maj-gauche et Maj-droite

Les concepts de décalage vers la gauche et de décalage vers la droite s'appliquent également à la sécurité. Parlons d'abord du décalage vers la gauche.

Depuis que les équipes DevOps ont commencé à utiliser des conteneurs comme moyen d'emballer des applications et ont commencé à publier des logiciels à une cadence plus rapide, il y a eu un désir d'automatiser les tests de sécurité des applications et de fournir des résultats de test plus tôt dans le cycle de vie du développement logiciel. En fournissant les résultats des tests plus tôt, les développeurs de logiciels peuvent corriger les failles de sécurité plus rapidement et plus facilement. Ils n'ont pas à se souvenir d'une modification qu'ils ont apportée il y a des semaines et qui a accidentellement introduit une faille de sécurité, et à démêler tout ce qui a été fait depuis lors.

En plus des informations fournies précédemment, les décisions de libération automatisées peuvent être prises plus tôt, sur la base des résultats des tests de sécurité. Cela a été le Saint Graal pour DevSecOps - fournissant plus d'automatisation et moins de travail manuel. Le résultat est meilleur, des logiciels plus performants et plus sécurisés - avec moins de travail nécessaire pour les êtres humains.

Qu'en est-il de la sécurité shift-right ? C'est important aussi. Après plusieurs années de « virage à gauche », les entreprises réalisent qu'elles doivent également conserver une visibilité dans l'environnement de production. Nous avons vu de nombreuses attaques réussies contre les environnements Kubernetes - des images malveillantes qui ont été insérées dans le hub Docker en 2020, aux attaques contre Azure et Tesla par des "cryptojackers". C'est pourquoi [44 % des entreprises déclarent](#) ils prévoient d'adopter de nouveaux contrôles de sécurité d'exécution (shift-right) au cours des 12 à 24 prochains mois.

En résumé, voici les raisons pour lesquelles la sécurité automatisée (DevSecOps) peut et doit passer directement aux environnements de production :

- L'environnement de production est connecté à Internet, là où se produisent la plupart des attaques.

- Numérisation du code source dans le développement
environnement ne peut pas vous donner les mêmes informations riches que vous pouvez obtenir en observant une application lorsqu'elle s'exécute en production. Par exemple, les analyses de code source statiques ne peuvent pas vous montrer quelles bibliothèques sont réellement chargées, comment elles sont utilisées, si un processus est exposé à Internet ou si un processus interagit avec des données d'entreprise sensibles.
- Certaines applications exécutées en production, telles que celles que vous achetez auprès de tiers, peuvent ne pas avoir été exécutées dans votre environnement de développement, elles n'ont donc jamais eu la chance d'être analysées par des outils de sécurité en cours de développement.
- De nouvelles vulnérabilités zero-day sont souvent découvertes après le déploiement d'une application en production. En mettant en place une surveillance continue de la sécurité des applications en production, vous pouvez être au courant de ces risques.



Renforcer la résilience avec l'ingénierie du chaos

[Ingénierie du chaos](#) est une discipline de développement qui soumet les logiciels à des défaillances dans un environnement de production simulé afin de renforcer la résilience des systèmes logiciels de production distribués. Cette pratique renforce la confiance dans la capacité du logiciel à résister à des circonstances inattendues ou improbables, telles que des pannes, des ralentissements, des charges excessives, etc.

Tester les performances de votre application sous des circonstances aléatoires et extrêmes sont utiles exercice pour s'assurer que votre équipe offre des résultats durables, des systèmes fiables et hautement disponibles dans n'importe quel situation. La seule façon de le faire est en production environnements avec de vrais utilisateurs et des niveaux de charge réels.



Adopter une approche de plateforme pour votre flux de valeur DevOps

Il ne manque pas d'outils DevOps que les équipes informatiques peuvent utiliser aujourd'hui pour exécuter différentes parties du cycle de vie DevOps. Mais à mesure que votre approche DevOps mûrit et que vous cherchez à faire évoluer DevOps sur plusieurs applications, la prolifération de la chaîne d'outils devient une réalité. Ce qui fonctionnait bien autrefois devient manuel, lourd, coûteux et revient à une approche cloisonnée. Imaginez que plusieurs équipes essaient d'utiliser les mêmes outils chacune pour leurs propres applications.

La standardisation sur une approche de plate-forme qui fournit l'automatisation, l'intelligence et l'observabilité en plus des processus DevOps réguliers permet de réduire les frais généraux, de réduire le travail et d'améliorer l'efficacité. Une approche de plate-forme tout-en-un crée une source unique de vérité qui supprime les silos, intègre des chaînes d'outils et permet des modèles en libre-service. Cette approche permet d'automatiser l'ensemble du pipeline de développement et donne aux développeurs et aux équipes d'exploitation les bons outils et données pour chaque étape du cycle DevOps, du codage à la livraison et inversement.

Piloter le développement futuriste aujourd'hui

L'objectif de tout chef d'entreprise et de technologie est de faciliter le développement d'applications et de services.

Ces pratiques exemplaires émergentes comprennent l'efficacité façons de développer des applications critiques, du code et prestations de service. De plus, en tirant parti de ces outils émergents, vous pouvez fournir une approche plus proactive et prescriptive architecture de développement capable de répondre aux besoins d'aujourd'hui exigences numériques.



Les principes fondamentaux de DevOps sont un ensemble de processus et de technologies qui permettent l'automatisation, la surveillance et l'observabilité, ainsi que l'AIOps. Ces processus conduisent à un état d'esprit de décalage vers la gauche et vers la droite qui se traduit par une livraison continue de versions résilientes et de qualité.



CHAPITRE 5

Avantages de DevOps

Au fur et à mesure que les équipes adoptent des changements à la fois dans la culture et

processus, l'approche holistique de DevOps en matière de création de logiciels et d'infrastructures peut être payante, même au niveau organisationnel. Une fois que les équipes ont établi des meilleures pratiques et des mesures clés à surveiller et à gérer, les équipes peuvent s'attendre à voir certains avantages essentiels.

Accélération de la livraison

Alors que les entreprises se transforment numériquement, la pression est forte sur les équipes de développement pour créer et fournir des logiciels plus rapidement et plus souvent sans sacrifier la qualité. Lors d'une enquête, 63 % des praticiens DevOps déclarent que DevOps leur permet de [publier des logiciels plus fréquemment](#). En revanche, le développement prend 41%

plus de temps dans les organisations qui n'utilisent pas les pratiques DevOps. Ce temps pourrait être mieux utilisé pour créer de nouvelles fonctionnalités ou mettre en œuvre de nouveaux processus. La différence? DevOps encourage les équipes à développer coder en plus petits morceaux et démocratiser l'accès à code, ce qui signifie que les développeurs travaillent sur des versions plus petites et plus fréquentes et avec des boucles de rétroaction plus étroites pour itérer rapidement et publier des logiciels plus rapidement.

Les processus DevOps décomposent les silos et favorisent une meilleure collaboration et des boucles de rétroaction entre les équipes. Cette connexion et cette coordination interfonctionnelles réduit les délais de livraison en permettant aux équipes d'automatiser les processus tels que la surveillance, l'évaluation des tests et la correction qui étaient auparavant effectués

Les avantages du DevOps

Les pratiques DevOps sont un investissement dont les dividendes augmentent avec le temps et l'expérience. Certains de ces avantages incluent :



Accélération de la livraison grâce à des processus améliorés



Meilleure qualité versions de logiciels de meilleurs tests



Productivité améliorée et la collaboration de développeurs habilités



De meilleures affaires les résultats de des clients plus satisfaits

manuellement. Grâce à cette automatisation, les équipes peuvent développer des modèles en libre-service pour accélérer les pipelines de livraison et faire évoluer les processus DevOps dans toute l'organisation.

Augmentation de la fréquence des sorties

Une philosophie centrale de DevOps est de se concentrer sur des changements de codage plus petits - en optant pour l'agilité et des poussées de fonctionnalités rapides, par opposition à la maintenance d'énormes bases de code et à la réalisation de versions monolithiques peu fréquentes. Ces petites modifications sont plus faciles à valider dans des référentiels de code tels que GitHub ou BitBucket, et plus facile à tester.

Des versions plus fréquentes signifient que les utilisateurs peuvent accéder plus rapidement aux nouvelles fonctionnalités et fonctions. Cela signifie également que les développeurs obtiennent plus rapidement des commentaires concrets, ce qui signifie qu'ils peuvent répondre aux problèmes et effectuer des optimisations plus rapidement.

Les versions fréquentes favorisent également l'amélioration continue des processus et des flux de travail DevOps.

Certaines entreprises, telles que Google, classent leurs équipes en fonction de leurs performances de publication, ce qui favorise une attitude instable et met l'accent sur la rapidité temps de réponse.

Réduction des risques et augmentation de la fiabilité des versions grâce à un logiciel de meilleure qualité

Un autre principe central de DevOps est fréquent, tests automatisés à chaque étape du développement cycle de vie, qui révèle les problèmes bien avant qu'ils ne surviennent production. Grâce à des pratiques comme le décalage à gauche, où le code est évalué par rapport aux SLO de production, les équipes de développement trouvent automatiquement ce mauvais code est empêché de passer à l'étape suivante contribuant ainsi à améliorer la qualité globale de logiciels et en réduisant les taux d'échec et de défauts.

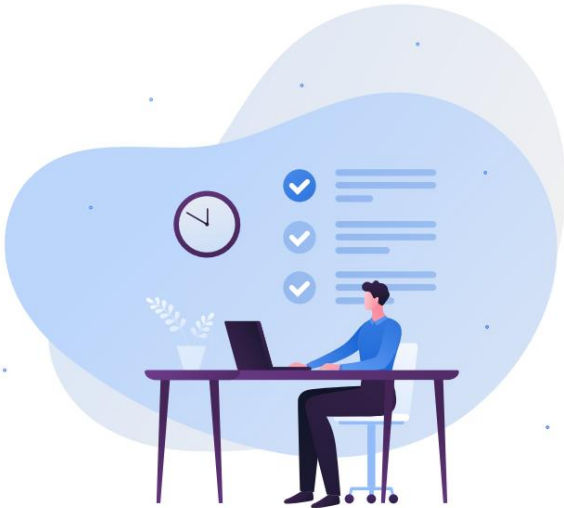
Les résultats des tests continus sont impressionnants : 63 % des organisations qui utilisent les pratiques DevOps ont [amélioré la qualité de leurs déploiements](#), tandis que 38 % signalent une amélioration globale de la qualité du code. Les équipes passent également 21 % de temps en moins à éteindre les incendies, ce qui laisse plus de temps pour innover et améliorer les processus.

Un risque réduit se traduit par une libération accrue confiance. Lorsque plus de tests et de processus, y compris les décisions de publication sont automatisées de manière fiable à l'aide de SLO portes de qualité pilotées, les équipes peuvent publier des logiciels sachant qu'il est bien testé et répond aux besoins des utilisateurs.

Autonomisation des développeurs

La planification, l'automatisation et les tests fréquents qui accompagnent les pratiques DevOps offrent aux développeurs une boucle de rétroaction rapide qui peut accroître leur confiance et leur permettre de travailler de manière indépendante.

En automatisant les tâches fastidieuses et les approbations de routine dans la mesure du possible, les équipes peuvent maximiser leur efficacité et travailler en toute confiance en sachant qu'elles suivent des flux de travail pré-approuvés. Ces processus permettent à une culture d'appartenance et de responsabilisation du groupe de se développer au sein des équipes. Lorsque des problèmes surviennent, la culture DevOps s'abstient de blâmer les erreurs. Au lieu de cela, l'accent est mis sur l'amélioration des processus, les leçons apprises et la croissance. Ce la culture peut remonter le moral et contribuer à une meilleure développement de logiciels.



Meilleure collaboration et productivité des développeurs

La culture DevOps encourage également un travail d'équipe solide, s'éloignant des silos au profit de flux de travail unifiés.

Alors que les équipes traitent des fragments de code plus petits et un développement rapide des fonctionnalités, les équipes peuvent s'engager dans une collaboration significative pour s'assurer que les pièces du puzzle s'emboîtent.

DevOps, et par extension DevSecOps, renforce les liens entre les développeurs, l'informatique, les opérations et les professionnels de la sécurité au sein d'une organisation, car les employés de chaque discipline sont plus conscients de chaque contribution. Avec une forte initiative DevOps, la transparence et la visibilité sont l'état naturel d'un projet. La collecte de métriques et une approche globale tiennent tout le monde au courant. Les outils DevOps centralisés et les plates-formes d'orchestration

équipes avec des informations et des ressources partagées.

Sécurité accrue

Depuis son introduction, DevOps s'est étendu au-delà

Les équipes de développement et d'exploitation incluront désormais les équipes de sécurité

— appelé DevSecOps. Cette méthodologie intègre de manière

transparente les tests de sécurité et la protection tout au long du

cycle de développement et de déploiement du logiciel. Tout comme

DevOps, DevSecOps consiste à adopter une approche

collaborative, et il est important que les organisations l'adoptent car la vitesse

de DevOps peut souvent rendre les applications du cycle de développement

vulnérables aux attaques de sécurité.

L'adoption de DevSecOps permet à votre organisation de maintenir

une approche collaborative tout au long du développement tout

en garantissant que la sécurité n'est pas compromise. Les évaluations

de sécurité ne peuvent pas attendre après le cycle de développement,

elles doivent plutôt avoir lieu simultanément. Ce faisant, les équipes

DevSecOps peuvent détecter et répondre aux failles logicielles en

production plus rapidement et plus efficacement. Cela se traduit par un

développement, une innovation et une livraison de logiciels plus

rapides.

Temps moyen de résolution réduit (MTTR)

Les systèmes complexes et à multiples facettes connaissent inévitablement

des défaillances, et les équipes doivent se préparer en conséquence.

Le MTTR réduit est une mesure du temps nécessaire pour réparer ces pannes

(généralement en heures ou en jours). Vous pouvez également envisager

cela comme un temps d'arrêt moyen sur une série de pannes. Naturellement,

des équipes très investies

dans les principes DevOps ont un MTTR inférieur. Il s'agit d'une mesure précise du respect des meilleures pratiques et des compétences collectives globales.

Dans de nombreux cas, la réduction du MTTR dépend simplement de la rapidité avec laquelle les équipes peuvent identifier et identifier les problèmes. Pour identifier rapidement les problèmes avec précision, les équipes doivent avoir une observabilité complète sur leur infrastructure dans un environnement DevOps. La réduction du MTTR repose également sur la surveillance, l'analyse de précision et la formulation de plans de remédiation. Les équipes passeront également moins de temps sur les appels d'assistance, non seulement parce que le code est de meilleure qualité, mais aussi parce que les équipes peuvent identifier plus rapidement les causes profondes.

Fiabilité et résilience logicielles accrues

Les pratiques DevOps consistant à gérer des blocs de code plus petits, à tester en continu et à automatiser les processus encouragent également les équipes à renforcer la fiabilité et la résilience des logiciels dès le début.

Lorsque le logiciel est conçu dans un souci de fiabilité et de résilience, il est beaucoup plus facile de restaurer le code si nécessaire, voire de supprimer le code en direct de la production lorsque des problèmes surviennent sans interrompre d'autres fonctionnalités.

Étant donné que les applications sont des collections de fonctions de base individuelles (en particulier à l'ère des plates-formes cloud et microservices), il est plus facile de désactiver ou de supprimer fonctionnalités défectueuses sans affecter l'ensemble du package.

Les équipes DevOps peuvent alors résoudre les problèmes et rééditer les services lorsqu'ils sont prêts, en maintenant la fiabilité du système.

Plusieurs outils permettent cela en un ou deux clics.

De plus, les données capturées pendant le fonctionnement du logiciel permettent aux équipes de concevoir de meilleurs tests et de prévenir de futurs problèmes.

Des clients plus satisfaits

DevOps encourage et permet aux équipes d'être plus réactives aux commentaires. Les processus tels que les restaurations ne sont pas relégués à de simples corrections de bogues - ils sont également utiles pour supprimer des fonctionnalités mal reçues

et les améliorer en créant un autre agile

méthode pour répondre aux demandes des utilisateurs. Les utilisateurs finaux sont également plus enclins à considérer les logiciels comme « innovants » ou "capable" lorsque la livraison des fonctionnalités est continue et sans couture.

De meilleurs résultats commerciaux

Nous parlons souvent des avantages que DevOps apporte directement aux équipes de développement et d'exploitation, mais les avantages de DevOps s'étendent également à l'ensemble de l'entreprise jusqu'au niveau C. Et Patrick Debois, connu comme le créateur du mouvement DevOps, note que le plus grand avantage de DevOps est la compréhension qu'il donne.

DevOps permet à une entreprise d'être plus polyvalente et axé sur l'information pour rencontrer le client et

les besoins de l'entreprise. Les avantages de l'adoption d'une approche DevOps se traduisent à leur tour par une efficacité accrue, une éthique de travail et, finalement, des bénéfices plus élevés qui peuvent être réinvestis dans l'entreprise pour une croissance future.

Prouver les progrès

Les dividendes des pratiques DevOps sont un avantage qui augmentent avec le temps et l'expérience. Établir une pratique DevOps est un investissement qui vient aussi avec ses défis.





CHAPITRE 6

Les défis du DevOps

Bien que les avantages de DevOps soient clairs, il peut être difficile d'établir les outils, les processus et les changements de culture nécessaires pour parvenir à une pratique DevOps réussie. En travaillant ensemble, les équipes de développement, informatiques et opérationnelles peuvent éliminer les obstacles et se concentrer sur l'amélioration de la façon dont elles créent, déploient et surveillent en continu les logiciels.

Bien que travailler ensemble soit un élément vital, ce n'est pas le seul ingrédient du succès. La transition vers DevOps nécessite une planification et préparation. Voici quelques pièges à éviter car les organisations planifient et mettent en œuvre des pratiques DevOps.

Pas d'adhésion par le haut

Pour qu'une organisation moderne réalise ses objectifs commerciaux, la direction doit faire confiance à son personnel technique et développer une compréhension des objectifs et des points faibles de l'informatique.

Étant donné que DevOps intègre des disciplines tout au long du cycle de vie du développement et des opérations, de la conception du produit aux appels d'assistance client, un DevOps solide ne peut se produire que si les organisations obtiennent l'adhésion des décideurs qui dirigent l'organisation ou les départements concernés. Les organisations les plus légères peuvent même avoir besoin de l'adhésion de tout le haut de la chaîne - le PDG. D'autres peuvent n'avoir besoin que d'un champion au niveau supérieur de la direction. Dans tous les cas, l'aîné

Les défis du DevOps

DevOps fait face à de nombreux défis sans stratégie appropriée et la discipline au sein de votre organisation. Ceux-ci peuvent inclure :



Pas d'adhésion des dirigeants



Pauvre observabilité stratégie



Manuel processus



Ne s'identifie pas/ obtenir les bonnes métriques



Sécurité vulnérabilités

la direction qui décide de la stratégie d'achat et de produit doit comprendre ce qu'il faut pour que les équipes informatiques et DevOps travaillent ensemble pour améliorer la façon dont l'organisation livre ses produits et services aux clients et promouvoir la culture changements nécessaires pour rendre l'organisation inter-organisationnelle DevOps réussi.

Ne pas avoir de stratégie d'observabilité

L'observabilité n'est pas la même chose que la surveillance. Dans un scénario de surveillance, des tableaux de bord préconfigurés sont destinés à alerter les équipes des problèmes de performances qu'elles attendez-vous à voir plus tard. Cependant, ces tableaux de bord reposez-vous sur une hypothèse clé : que les équipes peuvent prédire les types de problèmes qu'elles rencontreront avant qu'ils ne surviennent.

L'observabilité est basée sur les sorties d'un système et permet aux équipes de comprendre exactement ce qui est lent ou cassé. Avec une observabilité adéquate dans les applications et plates-formes cloud natives, les équipes de développement peuvent exploiter les données de télémétrie pour obtenir plus d'informations sur les applications et les systèmes, automatiser davantage de processus et publier plus rapidement un code de meilleure qualité. L'obtention d'une observabilité de bout en bout dans un environnement logiciel nécessite une combinaison d'examen minutieux et d'une technologie puissante et constitue un élément essentiel pour garantir l'évolutivité et le succès de DevOps.

Ne pas automatiser TOUS les processus manuels

Un objectif central de DevOps est d'automatiser autant de processus et de points de décision que possible pour améliorer le débit et la qualité des logiciels. Ici, les équipes doivent automatiser les tests, mais aussi les flux de travail, tels que la progression du logiciel du test à la publication ou la validation du code dans un référentiel.

Les équipes peuvent également trop s'appuyer sur les connaissances tribales qui vivent dans la tête de quelques-uns, nécessitant des approbations manuelles qui créent des goulots d'étranglement à l'automatisation.

Comme le nombre et le type de technologies que les environnements DevOps englobent augmentent et fluctuent constamment, la mise à l'échelle automatique en fonction de la demande devient impérative. L'automatisation de certains types de réponses, telles que l'alerte ou la correction automatique des problèmes de performances, est une autre capacité clé.

Ne pas apporter AIOps à DevOps

L'IA peut être considérée comme un tabouret à trois pieds, avec l'observabilité et l'automatisation. Prise de décision intelligente avec [l'IA basée sur la causalité](#) aide les équipes de développement à comprendre la cause profonde et à identifier précisément où les erreurs se produisent et ce qui a causé une défaillance de l'application.

Mais l'IA n'est pas seulement destinée aux mesures réactives, l'IA aide à prévoir les violations potentielles de SLO ou les défaillances d'application avant

ils entrent même en production, permettant aux équipes de corriger et de résoudre rapidement tout problème avant qu'il n'affecte les utilisateurs. L'exploitation de l'IA dans l'ensemble du pipeline aide le service informatique à faire évoluer son DevOps pour inclure des milliers d'applications et de microservices afin d'analyser des millions et des milliards de dépendances.

Ne pas adopter une approche en libre-service

Une partie importante de la transformation informatique est la capacité de donner aux technologues les outils dont ils ont besoin pour réussir au quotidien. Dans le cadre du développement d'applications et de services, il est important de considérer le libre-service comme un moyen de réduire les temps d'attente, de déployer de nouvelles fonctionnalités, de raccourcir les boucles de rétroaction entre les différentes équipes, d'aligner les outils et d'améliorer les pipelines CI/CD. Le libre-service est trop souvent ignoré jusqu'à la fin d'un effort de transformation ou complètement laissé de côté. Pour unifier les équipes DevOps, les organisations doivent intégrer un plan de libre-service dans la stratégie. Établir l'automatisation et l'observabilité dans votre stratégie DevOps simplifie la création de modèles en libre-service et la mise à l'échelle de DevOps.

Ne pas penser à la sécurité lors de la conception des processus

De nombreuses organisations traitent la sécurité comme une expertise distincte appliquée après le développement du code. Mais toute initiative DevOps doit avoir un plan pour intégrer la sécurité aussi étroitement que possible. Idéalement, l'équipe de sécurité devrait être un partenaire à part entière dans le cycle de vie du développement logiciel sur un pied d'égalité avec le développement et les opérations. C'est le sens de DevSecOps. En déplaçant la sécurité vers la gauche et en l'intégrant au produit à chaque étape du processus de développement et de livraison, les équipes rendent les applications et les services plus résilients contre un plus grand nombre de menaces, aujourd'hui et à l'avenir. DevSecOps offre une visibilité sur la vulnérabilité du code, la cartographie des dépendances, les révisions SDLC sécurisées, une compréhension approfondie de la façon dont une cible tolère une attaque réelle et jusqu'où un attaquant peut aller. Ne pas inclure la sécurité dans DevOps - ou au moins créer une feuille de route pour son inclusion à l'avenir

- est un faux pas critique qui sacrifiera d'innombrables informations importantes tout au long de la vie du produit.

Ne pas mesurer les bons indicateurs

Si les équipes parcourent les journaux d'erreurs ou essaient de manière réactive de reconstituer un problème, elles ne consultent peut-être pas les bonnes mesures. L'identification de ce que les organisations doivent mesurer est essentielle pour obtenir des informations précieuses à partir des systèmes de données produits.

Bien qu'il existe certains points de repère que toutes les organisations devraient surveiller, tels que le débit et la latence, les temps de réponse moyens, le temps d'attente, les erreurs et les impacts sur la mémoire, chaque organisation aura des métriques pour les systèmes et les processus qui sont uniques. Commencez par un ensemble de mesures clés qui évaluent la qualité du code et l'efficacité des tests, ainsi que l'efficacité du flux de travail et les temps de réponse aux incidents. Pour garantir l'adhésion de la direction, les équipes doivent comprendre les principaux points faibles qui préoccupent l'ensemble de l'organisation. Étant donné que les outils et processus DevOps touchent une empreinte numérique aussi large, ils offrent aux organisations la possibilité de découvrir, de mesurer et d'améliorer les tendances clés dans l'ensemble de l'organisation.

Un travail d'équipe

Aligner des équipes de différentes disciplines n'est jamais une tâche facile. Cependant, les leaders de l'espace technologique voient la frontière s'estomper entre l'informatique, les opérations et le développement, car les équipes dépendent les unes des autres pour assurer le succès de l'entreprise. Dans un écosystème sain où les opérations informatiques et de développement travaillent ensemble pour atteindre un objectif commun, les organisations peuvent mener de véritables efforts de transformation informatique avec une prise en charge complète du personnel et des initiatives DevOps.



CHAPITRE 7

Métriques DevOps

Au cœur de toutes les pratiques DevOps et SRE réussies se trouvent les métriques. La télémétrie de chaque étape du flux de travail DevOps - du développement et des tests au déploiement et aux opérations - fournit des indices critiques sur les performances de votre logiciel et sur l'efficacité de votre DevOps

les processus sont.

Des données fiables et mesurables sont nécessaires pour automatiser les tests, les validations et les versions. Pour établir les meilleures pratiques, les organisations peuvent commencer par le projet créé par l'équipe de recherche et d'évaluation DevOps (DORA) de Google, connue sous le nom de "[The Four Keys](#)", qui définit les quatre mesures de base qui indiquent la performance d'une équipe DevOps. Quels sont donc

Les quatre clés de DORA et quels autres indicateurs clés les organisations peuvent-elles suivre pour améliorer leurs pratiques DevOps et SRE ?

Fréquence de déploiement

La fréquence de déploiement mesure la fréquence à laquelle une organisation passe avec succès en production.

DevOps et intégration continue/livraison continue (CI/CD) vont de pair. Les équipes travaillent désormais avec des blocs de code plus petits, et les tests et validations continus permettent des validations plus rapides. Ce rythme de développement signifie que les équipes peuvent publier plus fréquemment, souvent plusieurs versions par jour.

Quatre mesures clés

DevOps est axé sur les données et son succès dépend de mesures fiables.

D'après [DORA](#) ceux-ci incluent (mais ne sont pas limités à):



Fréquence de
déploiement



Délai de mise en œuvre
pour les changements



Changement
taux d'échec



Entre temps
à la résolution (MTTR)

Une fréquence de déploiement élevée est essentielle pour répondre à la demande des clients. Les organisations plus rapides peuvent fournir des corrections de bogues, des améliorations et de nouvelles fonctionnalités, les développeurs les plus rapides peuvent recevoir de précieuses informations du monde réel commentaires et les utilisateurs peuvent réaliser une valeur qui renforce ta marque.

La fréquence de déploiement est une mesure à la fois à long terme et à court terme. Par exemple, vous pouvez mesurer le nombre de validations de code que vous effectuez chaque jour ou chaque semaine en réponse aux modifications de processus, peut-être comme indicateur d'efficacité. Sur de plus longues périodes, les équipes peuvent suivre si leur nombre de déploiements augmente au fil du temps. Les calendriers de publication lente peuvent indiquer des goulots d'étranglement ou des retards de service qui nécessitent une attention particulière.

Délai de modification

Le délai d'exécution mesure le temps qu'il faut pour que le code commis entre en production.

Le délai entre en jeu lors de la réponse à problèmes spécifiques liés à l'application et indique comment rapidement, votre équipe peut corriger un bogue ou un problème d'outillage. Comme la fréquence de développement, le délai d'exécution des changements aide les équipes à comprendre l'efficacité de leur processus sont.

Le délai d'exécution est facile à calculer et à quantifier, ce qui en fait une mesure accessible à toutes les parties prenantes de l'application.

Cependant, le délai de livraison n'est pas 100% noir et blanc.

Bien que des délais plus longs puissent indiquer des problèmes, ils peuvent également être le résultat de la concentration d'une équipe sur des projets complexes. Ces efforts prendront naturellement plus de temps.

Il est important d'étudier le contexte derrière les chiffres des délais et d'évaluer en conséquence. Alors que les organisations moyennes peuvent avoir des délais allant de

d'une semaine à un mois, certaines équipes DevOps peuvent repousser les changements de production en moins de 24 heures.

Deux moyens importants d'améliorer le délai de mise en œuvre des modifications consistent à mettre en œuvre des tests d'assurance qualité dans plusieurs environnements de développement et à automatiser les processus de test et DevOps.

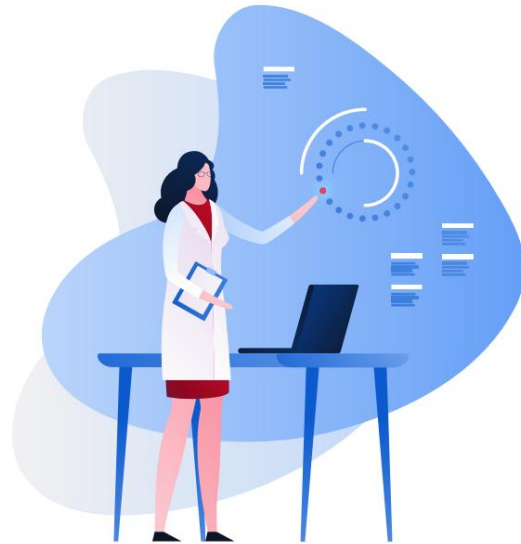
Modifier le taux d'échec

Le taux d'échec des changements mesure le pourcentage de déploiements qui entraînent un échec en production nécessitant une correction de bogue ou une restauration.

Les équipes peuvent valider et déployer des modifications avec une fréquence de déploiement et des délais de modification élevés taux, mais ces efforts sont diminués si des problèmes se faufilet dans la production. Un taux d'échec élevé (supérieur à 40 %) peut indiquer de mauvaises procédures de test et obliger les équipes à effectuer de petits changements fréquents, ce qui nuit à l'efficacité.

L'objectif derrière la mesure du taux d'échec des changements est d'évoluer vers un processus DevOps entièrement automatisé. En automatisant les tests et les processus, les logiciels publiés sont plus cohérents et fiables, et plus susceptibles de réussir en production.

Le calcul du taux d'échec des modifications nécessite la capacité de compter le nombre total de déploiements et de les lier à des incidents notables. Une solution AIOps automatisée peut trouver ces incidents dans GitHub ou d'autres rapports de référentiel de code, et surveiller les alertes système et les tickets utilisateur. Étant donné que tous les processus et systèmes de l'organisation sont uniques, la manière exacte de mesurer le taux d'échec des changements peut varier considérablement.



Temps moyen de résolution (MTTR)

Le temps moyen de résolution mesure le temps qu'il faut à une organisation pour se remettre d'une panne de production.

Les utilisateurs dépendent de la disponibilité des fonctionnalités, et 99,99+% de disponibilité est l'objectif convoité. Le délai de résolution est essentiel pour s'assurer que les équipes se remettent immédiatement et aussi efficacement que possible des pannes imprévues ou des défaillances de service, de peur que les organisations ne provoquent la frustration des utilisateurs et une perte de revenus.

Pour déterminer le MTTR, vous pouvez évaluer le temps entre quand un incident s'est produit par rapport au moment où c'était résolu. Quel déploiement a résolu cet incident ? L'observabilité des données de déploiement et des données d'expérience utilisateur est essentielle pour savoir si le service a été restauré efficacement. Des temps de restauration prolongés peuvent indiquer une mauvaise alerte ou une mauvaise surveillance et peuvent entraîner un plus grand nombre de systèmes affectés.

Une bonne pratique pour obtenir un MTTR rapide consiste à déployer logiciel par petits incréments pour réduire les risques et déployer des solutions de surveillance automatisées pour anticiper les pannes.

Le MTTR est une autre mesure qui varie considérablement d'un système à l'autre.

Au-delà de DORA : mesures supplémentaires importantes à suivre

Bien que les quatre clés de DORA constituent une base essentielle, il existe une myriade de mesures que vous pouvez utiliser pour suivre l'efficacité de vos processus DevOps et SRE.

Voici six autres mesures que vous pouvez suivre pour évaluer de manière globale l'efficacité de votre pipeline.

1. Taux d'échappement des défauts

Également connue sous le nom de vitesse d'échappement des défauts, cette métrique mesure le taux de problèmes qui « échappent » à la détection pendant le développement et sont découverts en production. Vous pouvez ensuite calculer le taux de défauts par période, par version ou par déploiement. Des taux d'échappement plus élevés peuvent indiquer des problèmes de test et des lacunes d'automatisation associées. Il est tout à fait possible que les outils soient défectueux lorsque les défauts se propagent.

2. Temps moyen de détection (MTTD)

Le MTTD mesure la rapidité avec laquelle les équipes découvrent les problèmes en moyenne. Les pannes, les vulnérabilités et les pannes du système à grande échelle peuvent causer des ravages sur les applications plus elles persistent. L'identification et la résolution de ces problèmes sont primordiales pour réduire l'impact global d'un problème sur vos applications, votre infrastructure et vos utilisateurs.

Une surveillance efficace est un principe central de DevOps.

Pour atteindre un faible MTTD, les équipes doivent mettre en œuvre une surveillance, une alerte et une observabilité de bout en bout efficaces pour détecter immédiatement une anomalie ou une dégradation du service. La source d'un problème peut être évidente, comme une panne de service central, mais il peut aussi être plus difficile à détecter, comme une défaillance du back-end, un problème avec un outil open source ou un extrait de code défectueux. La détection de problèmes comme ceux-ci nécessite une surveillance complète de la pile avec une visibilité au niveau du code.

3. Pourcentage de code couvert par des essais

Un autre objectif central de DevOps est d'automatiser autant que possible. En conséquence, les tests automatisés font partie intégrante de la détection d'un plus grand nombre d'erreurs de syntaxe, de sécurité et de compatibilité dans les versions.

Pour ce faire, les organisations peuvent mettre en place des environnements de test qui simulent automatiquement comment le code se comporte dans diverses circonstances.

L'augmentation du pourcentage global de code qui passe par des tests et une validation automatisés rend les tests plus rapides et plus faciles, ce qui accélère le pipeline DevOps et raccourcit la boucle de rétroaction de la détection du problème à sa résolution.

4. Disponibilité des applications

[Disponibilité des applications](#) est une mesure utilisée pour évaluer dans quelle mesure une application fonctionne pleinement et est disponible pour répondre aux besoins de l'entreprise et de l'utilisateur final. Un système hautement disponible est conçu pour répondre à l'indicateur de performance clé de référence de cinq 9 (99,999 %). Garantir une plus grande disponibilité des applications maintient les clients engagés et connectés à vos services.

Bien que les temps d'arrêt ne soient pas toujours prévus, ils sont souvent planifiés à la suite d'une maintenance. La communication entre les membres de l'équipe DevOps et SRE est cruciale pour résoudre les pannes imprévues et garantir que le frontend et le backend fonctionnent de manière transparente.

5. Utilisation et trafic des applications

L'utilisation et le trafic des applications surveillent le nombre d'utilisateurs accédant à votre système et informent de nombreuses autres mesures, y compris la disponibilité du système. Les statistiques d'utilisation sont utiles pour les équipes, car il n'est pas rare que les mises à jour d'applications aient un impact sur l'activité des utilisateurs, qu'elle soit bonne ou mauvaise. Cela peut se produire lorsque des problèmes surviennent ou lorsque des fonctionnalités attendues depuis longtemps disparaissent, entraînant une augmentation du trafic. Ces métriques permettent aux membres de l'équipe DevOps de réagir et de gérer efficacement ces pics.

À l'autre extrémité du spectre, lorsque l'activité de l'utilisateur ralentit, cela pourrait suggérer qu'un service est interrompu sous une forme ou une autre. Bien que les changements soudains soient beaucoup plus révélateurs pour les équipes, il est également important d'évaluer les tendances à long terme des problèmes qui se développent au fil du temps.

DevOps est axé sur les données

Où que vous en soyez dans votre cheminement vers la maturité DevOps, la capacité de mesurer et de donner un sens aux données provenant de chaque étape de votre flux de travail vous aidera à perfectionner vos stratégies. Dans tous les domaines, des performances, de la fiabilité et de la stabilité des applications à l'efficacité et à l'efficience de DevOps, les métriques sont la lentille de vos pratiques DevOps pour vous aider à conduire une amélioration continue. Tirer parti d'une solution d'observabilité qui fournit des données et des analyses haute fidélité sur lesquelles vous pouvez compter pour alimenter ces métriques permet aux équipes d'augmenter la réactivité, d'affiner les processus et de fournir de meilleurs logiciels plus rapidement.



CHAPITRE 8

DevOps et l'observabilité intelligente permettent la transformation numérique

La transformation numérique est désormais essentielle pour que les entreprises atteignent leurs objectifs commerciaux. En créant une boucle de rétroaction qui se renforce continuellement, DevOps offre aux organisations un moyen de faire évoluer la culture d'entreprise et d'habiller la prochaine génération de création, de gestion et de sécurité de logiciels.

DevOps combine le développement et les opérations dans un cadre unifié qui brise les silos et favorise la collaboration tout au long du cycle de vie. Dans cet environnement, les SRE peuvent mettre en œuvre des opérations qui garantissent la disponibilité, la latence, les performances et la résilience des systèmes logiciels. De même, les pratiques CI/CD peuvent fournir

un développement, des tests, une livraison et un déploiement bien alignés et automatisés.

Des pratiques DevOps efficaces et efficaces dépendent également de la surveillance de mesures clés, telles que les DORA Four Keys qui mesurent la fréquence de déploiement, le délai de modification, le taux d'échec des modifications, la disponibilité des applications et le temps moyen de restauration du service (MTTR), entre autres. Une fois les bases de DevOps posées, les organisations peuvent développer les meilleures pratiques, telles que l'automatisation, la surveillance et l'observabilité, ainsi que l'AIOps, qui permettent la livraison continue de logiciels à grande échelle.

DevOps n'améliore pas seulement les flux de travail, il offre également des avantages mesurables de bout en bout, notamment une vitesse de livraison accrue, une fréquence de publication accrue, un risque réduit et un MTTR réduit.

Pour faciliter des pratiques DevOps, SRE et CI/CD fluides, la plateforme d'intelligence logicielle optimisée par l'IA de Dynatrace s'intègre de manière transparente à la chaîne d'outils DevOps d'une organisation et automatise les tâches tout au long du cycle de vie DevOps. Avec une automatisation continue et une détermination précise des causes profondes,

Dynatrace permet aux organisations de réaliser le potentiel de DevOps et de simplifier la complexité du cloud. En permettant une innovation plus rapide, une collaboration plus efficace et la capacité d'intégrer la sécurité des applications dans le cadre des solutions DevSecOps émergentes, Dynatrace fournit des réponses précises pour chaque phase du développement logiciel et du cycle de vie de la livraison.

Prêt à exploiter le potentiel de transformation de la livraison numérique ? Découvrez comment Dynatrace peut vous aider.

[Apprendre encore plus](#)



À propos de Dynatrace

Dynatrace fournit une intelligence logicielle pour simplifier la complexité du cloud et accélérer la transformation numérique. Avec une observabilité automatique et intelligente à grande échelle, notre plateforme tout-en-un fournit des réponses précises sur les performances et la sécurité des applications, l'infrastructure sous-jacente et l'expérience de tous les utilisateurs pour permettre aux organisations d'innover plus rapidement, de collaborer plus efficacement et de fournir plus de valeur avec beaucoup moins d'effort. C'est pourquoi bon nombre des plus grandes entreprises du monde font confiance à Dynatrace® pour moderniser et automatiser les opérations cloud, publier de meilleurs logiciels plus rapidement et offrir des expériences numériques inégalées.