



Guide de démarrage

DevOps

Sommaire

03 INTRODUCTION

04 QU'EST-CE QUE LE DEVOPS ET COMMENT PEUT-IL AIDER VOTRE ENTREPRISE ?

05 TECHNOLOGIES ET PROCESSUS DE BASE

09 AUTRES TECHNOLOGIES DEVOPS CLÉS QUE VOUS DEVEZ COMPRENDRE

12 À QUOI RESSEMBLE LE DEVOPS EN PRATIQUE ?

13 LA RAISON POUR LAQUELLE APPRENDRE LE DEVOPS EST UN ATOUT POUR VOTRE CARRIÈRE

14 RESSOURCES

16 À PROPOS DE GITLAB

Introduction

Le nombre d'entreprises qui utilisent le DevOps augmente considérablement. Par conséquent, plusieurs membres des équipes DevOps découvrent pour la première fois cette technologie et ses pratiques. Selon le rapport intitulé [2021 Upskilling Enterprise DevOps Skills Report](#) du DevOps Institute, 24 % des plus de 2 000 professionnels interrogés ont déclaré faire partie de leur équipe DevOps depuis moins d'un an, tandis que 30 % ont répondu avoir entre un et deux ans d'expérience. Par conséquent, si vous venez de vous lancer dans le DevOps, vous n'êtes certainement pas le ou la seul(e) dans le cas.

Ce guide contient tout ce que vous devez savoir pour être opérationnel. Nous vous expliquerons ce qu'est (et n'est pas) le DevOps, quelles technologies et quels termes clés vous devez connaître et pourquoi la collaboration est si importante. Nous vous présenterons ensuite un exemple pratique du DevOps, puis vous indiquerons pourquoi le DevOps peut booster votre carrière et faire augmenter votre salaire. Enfin, nous mettrons à votre disposition une longue liste de ressources ainsi qu'un questionnaire qui vous permettra de tester vos connaissances.

Plongeons maintenant dans le vif du sujet.



Qu'est-ce que le DevOps et comment peut-il aider votre entreprise ?

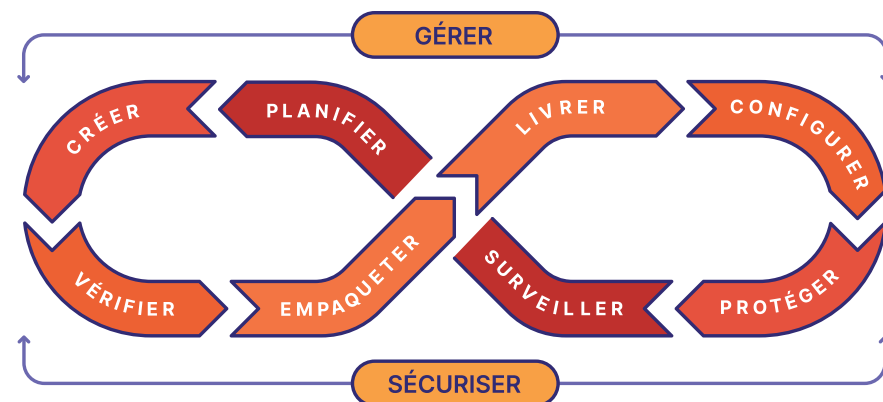
La première chose que vous devez savoir [à propos du DevOps](#) est qu'il implique de responsabiliser les équipes afin d'instaurer une culture de travail collaborative dans toute l'entreprise de manière à créer et fournir des logiciels sécurisés plus rapidement et plus efficacement. Pendant des années, le développement de logiciels était tout sauf fluide et efficace. Les processus étaient cloisonnés et créaient des goulots d'étranglement ainsi que des retards coûteux, tandis que la sécurité intervenait bien trop tardivement. Le DevOps est né de la profonde frustration ressentie à l'égard des anciennes méthodes de travail et a apporté avec lui la promesse de processus plus simples et plus rapides.

Nous pensons que le DevOps est mieux mis en œuvre sur une seule plateforme de bout en bout. Avec une plateforme DevOps, les équipes peuvent abandonner ou éviter d'utiliser une multitude d'outils souvent complexes et déconcertants en faveur d'un écosystème de développement logiciel unique et complet. Les membres de l'équipe n'ont ainsi plus à passer d'un outil à l'autre, ce qui permet de gagner du temps et d'économiser de l'argent. Cet écosystème peut servir à concevoir, créer et, en fin de compte, déployer des logiciels de meilleure qualité, plus sécurisés et plus conformes, de manière plus efficace, continue et rapide. Il aidera ainsi non seulement les équipes DevOps, mais aussi l'ensemble de l'organisation à être [plus agiles](#) afin de permettre aux entreprises de répondre plus rapidement aux besoins des clients, de rester

conformes, de garder une longueur d'avance sur leurs concurrents et de s'adapter rapidement à l'évolution du contexte économique. Le DevOps stimule donc l'agilité tant dans le cadre du développement et du déploiement de logiciels que dans celui des opérations des entreprises.

En recourant à l'automatisation, en impliquant la sécurité plus en amont et en rendant les processus reproductibles et mesurables, une plateforme DevOps permet d'améliorer les logiciels et de réduire le temps entre la conception de nouvelles fonctionnalités de meilleure qualité et leur déploiement dans l'environnement de production. Et cela maximise le rendement global du développement logiciel.

Sachez également que [la culture](#) ou l'état d'esprit qui sous-tend le DevOps ne doivent pas être laissés de côté. Il ne s'agit pas d'un processus de développement habituel. La culture DevOps repose sur [une collaboration et une responsabilité partagée](#), ainsi que sur un cycle constant d'itération, de mesure, d'évaluation et de réévaluation. Encore une fois, tout est question d'agilité et de capacité à apprendre et à mettre en œuvre rapidement pour apporter des améliorations continues et itératives, et procéder au déploiement des fonctionnalités.



Technologies et processus de base

Les étapes du processus DevOps

Votre apprentissage en matière de DevOps n'est pas terminé tant que vous n'avez pas compris les étapes du cycle de développement, c'est-à-dire le processus allant de la planification au lancement de nouvelles fonctionnalités, en passant par l'analyse et la collecte des retours.

Lorsque vous travaillez dans le DevOps, vous devez impérativement connaître ces étapes, car chacune d'entre elles fait partie intégrante du processus. Dans l'ensemble, il comprend trois étapes globales qui s'exécutent dans un ordre logique : le build, le test et le déploiement. Il s'agit là du cours naturel des opérations. Vous créez le code, puis vous le testez et, si tout fonctionne, vous le déployez.

Il faut y regarder de plus près pour déceler les couches plus complexes de ces étapes. Chacune est déterminante pour générer de la valeur logicielle et métier. Une fois que vous comprendrez et utiliserez ce processus, vous gagnerez en efficacité, en fiabilité, en rapidité et en agilité. Voici un aperçu plus détaillé des neuf étapes clés :

- **Planification** – Cette étape du DevOps englobe tout ce qui se passe avant que la première ligne de code ne soit écrite. Elle consiste à créer une feuille de route du produit qui guidera le développement à venir et aidera l'équipe à organiser les ressources et les priorités, et à aligner et suivre les projets.
- **Création** – Il s'agit de la première étape du pipeline CI/CD. C'est à ce moment-là que le code est conçu et développé. Les développeurs utilisent le contrôle de version pour coordonner toutes les modifications qu'ils apportent à la même base de code. C'est l'un des moyens d'accélérer le processus.
- **Vérification** – Cette étape consiste à confirmer la qualité du code. Elle s'appuie sur les tests de sécurité, l'analyse de la qualité du code, l'exécution parallèle et l'automatisation pour garantir que les retours parviennent rapidement aux développeurs et aux testeurs, et que chaque commit peut être visualisé immédiatement. Si les développeurs peuvent trouver et corriger les erreurs lors du codage, c'est moins cher et plus efficace.
- **Empaquetage** – Cette étape est lancée une fois que le code a été écrit et testé. L'empaquetage des applications et des dépendances, la gestion des conteneurs et la création d'artefacts contribuent à créer une chaîne logistique fiable.
- **Livraison** ou déploiement – Cette phase consiste à transférer les mises à jour du code dans l'environnement de production. Avec le DevOps, les versions peuvent être déployées lorsque les itérations sont créées, testées et prêtes, et non à une date de livraison groupée statique et pré-planifiée.
- **Configuration** – Elle comprend la mise en place, la gestion et la maintenance des environnements d'application. La gestion automatisée de la configuration est conçue pour gérer ces environnements complexes sur les serveurs, les réseaux et les systèmes de stockage.
- **Surveillance** – Cette partie proactive et automatisée du processus se concentre sur le suivi des logiciels, de l'infrastructure et du réseau afin de garder un œil sur leur statut et d'émettre des alertes en cas de problèmes. Elle permet d'augmenter la sécurité, la fiabilité et l'agilité.



- **Protection** – Cette étape vise à protéger les applications et leur environnement d'exécution contre les intrusions et les nouvelles vulnérabilités.
- **Supervision** – Elle implique d'avoir une visibilité et un contrôle sur l'ensemble du cycle de développement logiciel par le biais de la gestion des autorisations, de la standardisation des processus de création et de déploiement DevOps et de l'automatisation des garde-fous afin de garantir que les directives de sécurité et de conformité sont respectées.

Qu'en est-il de la sécurité ? C'est une bonne question. Et c'est la beauté du DevOps : la sécurité n'est pas une considération secondaire. Elle intervient dans CHAQUE étape du processus, de la documentation des exigences aux tests automatisés en passant par la validation de ces exigences. Cela garantit que le nouveau code et les nouvelles fonctionnalités fonctionnent exactement comme ils ont été conçus et qu'aucun bug, qu'aucune menace pour la sécurité ou qu'aucun problème de conformité ne puisse apparaître.

Ces étapes font toutes partie d'un cycle continu. Toutes les informations créées au cours du processus sont instantanément disponibles via la plateforme pour toutes les personnes impliquées à toutes les étapes. Elles constituent ainsi une source unique d'informations qui favorise la visibilité et la collaboration. L'autre avantage intéressant d'une plateforme commune est la possibilité de gérer et de contrôler l'ensemble du cycle de développement logiciel à partir d'une seule interface.

Outils qui optimisent ces étapes du cycle DevOps

GESTION DU CODE SOURCE

Un outil de gestion du code source (Source Code Management ou SCM en anglais) offre à de nombreux développeurs la possibilité de

partager un repository de code sans que les modifications d'une personne ne remplacent celles d'une autre. Le code est divisé et organisé en projets et groupes de projets. Lorsqu'un développeur teste le code existant ou ajoute du code, l'outil SCM identifie les modifications conflictuelles apportées au même code et signale qu'elles doivent être corrigées. Ce processus permet à plusieurs développeurs de travailler simultanément sur un projet, un facteur essentiel pour déployer plus rapidement les mises à jour logicielles. Le DevOps prospère sur les repositories Git ([une différence importante par rapport aux anciens systèmes de contrôle de version](#)), car leur architecture moderne offre des fonctions puissantes.

INTÉGRATION CONTINUE

L'étape de l'intégration continue (Continuous Integration ou CI en anglais) rend l'itération possible en validant les modifications à temps et fréquemment (plusieurs fois par jour) dans un repository de code source commun, en testant automatiquement chaque modification et en lançant un build.

L'intégration continue est étroitement liée à l'efficacité. En automatisant le travail manuel et en testant le code plus fréquemment, les équipes peuvent itérer plus rapidement et déployer de nouvelles fonctionnalités plus souvent avec moins de bugs. Parmi ses [autres avantages](#), la CI permet également d'identifier et de résoudre les problèmes plus facilement, de réduire le nombre des changements de contexte pour l'équipe et d'améliorer la satisfaction des utilisateurs et des clients.

Si vous souhaitez tirer le meilleur parti de l'intégration continue, vous devez vous assurer que votre configuration comprend ces éléments clés :

- un repository de code source avec tous les fichiers et scripts nécessaires pour créer les builds ;



- des builds automatisés grâce à des scripts qui incluent tout le nécessaire pour compiler en une seule commande ;
- des builds avec autotest qui automatisent vos politiques (par exemple, mise en échec si un test échoue) ;
- des itérations et des commits fréquents afin de réduire le nombre de conflits ;
- des environnements de test stables qui reflètent fidèlement l'environnement de production ;
- de la visibilité pour que chaque développeur puisse accéder aux derniers exécutables et visualiser les modifications apportées au repository.

LIVRAISON CONTINUE

La livraison continue (Continuous Delivery ou CD en anglais) est un processus de développement logiciel utilisé conjointement avec l'intégration continue pour automatiser le processus de livraison d'une application. Une fois que le code a été testé et compilé dans le cadre du processus d'intégration continue, les étapes finales sont prises en charge par la livraison continue pour garantir que les paquets de code contiennent tout le nécessaire pour se déployer dans n'importe quel environnement, à tout moment. La livraison continue peut englober une multitude d'opérations, du provisionnement de l'environnement de l'infrastructure au déploiement de l'application testée en environnement de test/pré-production ou de production.

Grâce à la livraison continue, le logiciel est compilé de manière à pouvoir être mis en production à tout moment. Vous pouvez ensuite déclencher les déploiements manuellement ou automatiser le processus.

Lorsque la livraison continue est bien configurée, les processus de livraison de logiciels deviennent monotones, car ils présentent un faible risque, sont cohérents et reproductibles. Vous pouvez alors planifier en toute confiance les processus et les calendriers de livraison, automatiser l'infrastructure et les déploiements, et gérer plus efficacement vos ressources cloud.

TESTS AUTOMATISÉS

Les tests automatisés sont essentiels pour adopter pleinement le DevOps et l'intégration continue, ainsi que pour livrer plus fréquemment du code de meilleure qualité. Lorsque les tests sont intégrés au pipeline CI, chaque modification de code validée déclenche la création d'un build. Le build exécute ensuite des tests pour s'assurer que les modifications réussissent tous les tests et respectent les politiques et les normes de conformité de code définies pour votre application. De cette façon, les bugs sont identifiés plus tôt et avec plus de contexte pour simplifier la correction, vos équipes peuvent effectuer des déploiements plus fréquemment et en toute confiance, et vous réduisez au strict minimum les tests manuels et les retouches en fin de processus.



Sécurité en amont

Une opération clé pour garantir la réussite du DevOps consiste à intégrer la sécurité dans l'automatisation de bout en bout. Cette approche est plus connue sous le nom de DevSecOps. Si vous intégrez les tests et le processus de vérification de la sécurité au début du cycle de vie du développement logiciel, il est plus facile de traiter correctement les problèmes liés à la sécurité. D'autre part, si les tests de sécurité passent au second plan ou sont uniquement effectués lorsque le code est prêt pour la production, il peut être difficile de revenir en arrière et de résoudre les problèmes, et il est même parfois trop tard pour les corriger rapidement et efficacement. Cela peut ensuite entraîner des retards dans les déploiements, des vulnérabilités dans l'environnement de production, une dette technique plus importante et des cloisonnements inefficaces entre la sécurité et le reste des équipes DevOps.

Pour intégrer la sécurité en amont, vous devez ajouter des tests de sécurité à vos pipelines CI. De cette façon, le code est continuellement testé en termes de sécurité globale, et non pas uniquement pour les autres commits du repository partagé. Voici quelques types de tests de sécurité que vous devriez inclure au début du cycle de développement :

- Tests statiques de sécurité des applications (SAST)
- Tests dynamiques de sécurité des applications (DAST)
- Analyse des images des conteneurs et des clusters
- Analyse des dépendances
- Détection des secrets
- Analyse de l'Infrastructure-as-code (IaC)
- Tests API

Documentation

Bien que parfois négligée, la documentation est inestimable pour la mise en œuvre des pratiques DevOps. La création et la maintenance d'une documentation interne pour les services et les applications sur lesquels votre équipe travaille et pour votre processus DevOps peuvent contribuer à améliorer les performances de votre équipe et des logiciels que vous créez. Le rapport [Accelerate State of DevOps](#) de 2021 a révélé que les équipes disposant d'une documentation de meilleure qualité sont 2,4 fois plus susceptibles d'améliorer la livraison des logiciels et les performances opérationnelles.

Retours

Les retours (ou feedback) constituent une pièce essentielle du puzzle, car les entreprises doivent toujours rechercher des moyens d'améliorer l'expérience utilisateur ainsi que le processus DevOps en général. Dans le cadre du développement traditionnel de logiciels, il peut s'avérer difficile d'obtenir des retours. Avec le DevOps, une collaboration plus étroite et des itérations rapides signifient que les équipes ont un accès constant à des données gérables de sorte qu'elles puissent exploiter le feedback, ajuster leurs efforts et apporter des améliorations efficacement et rapidement. L'automatisation du processus est donc une étape primordiale pour s'assurer que les informations sont collectées et distribuées aux bonnes personnes dans l'équipe, qui pourront ensuite adapter rapidement les nouvelles mises à jour du code.

Vous débutez dans le DevOps ?

Répondez à quelques questions afin de nous permettre de mieux comprendre ce qui vous intéresse.

Répondre à l'enquête 



Commencez votre essai gratuit de GitLab

Autres technologies DevOps clés que vous devez comprendre

Au fur et à mesure que vous vous familiarisez avec le DevOps, vous devez vous renseigner sur plusieurs domaines et vous assurer de rester à jour sur ces sujets. Nous avons rassemblé certains de ces domaines ici :

Le cloud

Si vous travaillez dans le DevOps, il est préférable de comprendre comment fonctionne [le cloud](#). La plupart des logiciels modernes s'appuient sur une infrastructure cloud et cloud-native, y compris les conteneurs et les orchestrateurs qui aident à automatiser le processus de développement et de livraison des logiciels. Les applications ainsi développées offrent aux spécialistes du DevOps la possibilité d'effectuer un déploiement n'importe où et de tirer le meilleur parti des [différentes plateformes cloud](#), voire des clouds privés de leur datacenter.

Une approche cloud-native est plus évolutive, car le code ne dépend plus du matériel utilisé. Afin d'aider leur entreprise et de booster leur carrière, les professionnels doivent avoir une bonne connaissance des fournisseurs, des services et des plateformes cloud, c'est même un point crucial pour nombreux spécialistes du DevOps. Selon l'enquête intitulée [2021 Global DevSecOps Survey](#), les personnes interrogées ont déclaré vouloir en savoir plus sur les compétences cloud et l'architecture sans serveur dans le cadre de l'amélioration

des compétences. De plus, 56 % d'entre elles affirment que leur priorité absolue au travail est la gestion des services cloud.



Une culture de collaboration

La collaboration est essentielle, c'est même l'un des principes de la pratique et de la philosophie du DevOps. En invitant les membres de l'équipe, qu'ils soient expérimentés ou récemment arrivés, à discuter, à apporter leur contribution et à offrir leur aide, il est possible de créer une culture d'apprentissage et de confiance basée sur l'expertise des autres. Pour faire partie d'une véritable culture DevOps, vous devez également être capable d'écouter, de rester calme sous pression, d'instaurer une relation de confiance entre les membres de l'équipe et de vous approprier une situation ou un problème donné.

Cependant, la collaboration ne concerne pas seulement les collaborateurs DevOps qui travaillent ensemble. Elle implique également d'autres composantes de l'entreprise comme les équipes de sécurité, du marketing, de la finance, du service client et même

la direction. La collaboration entre le DevOps et la sécurité, par exemple, est un moyen d'intégrer la sécurité dans l'ensemble du processus de développement.

Ne commettez pas l'erreur de considérer cet aspect comme une compétence non technique qui a moins d'importance que les capacités techniques. [Il vous faut développer des compétences clés](#) telles que la communication pour savoir parler des besoins de l'entreprise et collaborer de manière à résoudre des problèmes. Dans l'enquête 2021 Global DevSecOps Survey, la communication et la collaboration ont souvent été classées comme des compétences importantes. D'ailleurs, un nombre égal de participants ont cité l'expertise en la matière et ces compétences comme étant les plus utiles pour leur avenir professionnel.

Langages de programmation essentiels

Les ingénieurs DevOps doivent être capables d'écrire du code, mais ils doivent surtout tenir compte des processus, des outils et des méthodologies qui sont utilisés tout au long des étapes du cycle DevOps décrites ci-dessus. Certains langages se prêtent mieux à ce processus de bout en bout que d'autres, mais il existe de nombreux langages de programmation et il peut être très difficile de savoir par où commencer. Vous devez d'abord déterminer ce dont vos équipes DevOps ont besoin. Sur quels projets travaillez-vous ? De quels langages avez-vous actuellement besoin ? Quels langages seront nécessaires pour les projets futurs ?

Certains des langages de programmation les plus populaires sont Python, Golang, Ruby, JavaScript, Perl, Java, Bash et PHP. Selon l'enquête intitulée [Stack Overflow 2021 Developer Survey](#), le langage de programmation le plus utilisé pour la neuvième année consécutive est JavaScript, tandis que Python est le langage que la plupart des professionnels souhaitent apprendre pour la cinquième année consécutive. Quant aux autres langages, 56 % des personnes

interrogées ont déclaré utiliser HTML/CSS, 48 % ont répondu qu'ils utilisent Python, 47 % ont recours à SQL et plus de 35 % ont affirmé employer Java. Il est également primordial de développer vos propres compétences. Selon le rapport 2021 Upskilling Enterprise DevOps Skills Report du DevOps Institute, il est préférable de ne pas se spécialiser dans un seul langage, mais plutôt d'en maîtriser plusieurs.

Vous avez besoin de pratique ? Pourquoi ne pas [offrir votre temps en tant que bénévole](#) ou contribuer à des projets open source pour enrichir votre CV ?

Automatisation

Les équipes DevOps ont de plus en plus tendance à automatiser les processus tout au long du cycle de développement et de déploiement. Par conséquent, il va sans dire que vous devez savoir comment fonctionne l'automatisation et comment l'utiliser. L'automatisation permet de réduire le temps et l'argent consacrés aux tâches répétitives et d'éliminer les erreurs humaines en rationalisant l'ensemble du processus DevOps.

Grâce à l'automatisation, chaque tâche est effectuée de manière identique et avec cohérence, fiabilité et précision. Cela accélère le processus et augmente le nombre de livraisons (et donc de déploiements). L'automatisation n'exclut pas les humains, elle minimise seulement la nécessité de faire appel à eux pour la gestion des tâches récurrentes, comme la surveillance de la disponibilité, des performances ou des problèmes de sécurité, la configuration homogène des environnements logiciels, le test des nouvelles versions des applications selon des normes de qualité prédéfinies, l'intégration de code, l'accélération des déploiements, la prise en charge des logiciels de test CI/CD pendant le processus de développement et la gestion des journaux et de la documentation. Oui, ça fait beaucoup.



Selon l'enquête 2021 Global DevSecOps Survey, plus de 55 % des équipes ont déclaré que leur cycle de développement logiciel est « entièrement » ou « presque entièrement » automatisé. 27 % ont quant à elles affirmé qu'il est « partiellement automatisé ». Cela signifie que si vous travaillez dans le DevOps, vous devriez normalement déjà posséder quelques connaissances en automatisation.

Surveillance

À mesure que la pile d'applications de votre entreprise et le nombre d'équipes DevOps qui y travaillent augmentent, le nombre d'éléments mobiles ne fera que se multiplier, et les suivre à la trace peut être une tâche compliquée. Par conséquent, une surveillance continue est une condition sine qua non pour pouvoir garder un œil sur la situation de votre écosystème de bout en bout et en temps réel. Grâce au DevOps, les applications les plus complexes peuvent être mises à jour et déployées quotidiennement, voire plusieurs fois par jour. Une surveillance poussée et automatisée est un moyen proactif de réduire les bugs, d'améliorer la vitesse et l'efficacité des déploiements, d'identifier les menaces de sécurité et les problèmes de conformité, d'éliminer les modifications indésirables et de conserver la documentation. Elle doit être appliquée de la planification au développement, y compris lors de l'intégration, des tests, du déploiement et même des opérations.

En d'autres termes, la surveillance est un processus indispensable non seulement pour les développeurs, mais aussi pour les chefs de projet et les équipes sécurité. Elle ne se résume pas à suivre les processus, elle sert également à générer des alertes en cas de problèmes de performance et de menaces dans le pipeline. Selon le rapport Accelerate State of DevOps de 2021, les meilleurs professionnels DevOps qui atteignent leurs objectifs de fiabilité sont 4,1 fois plus susceptibles d'utiliser des solutions qui incluent une surveillance complète du système. Les équipes DevOps qui

cherchent à exceller dans le domaine travailleront de plus en plus avec la surveillance, vous devez donc bien connaître cet outil.

Conteneurs

Le domaine du DevOps a pleinement adopté les conteneurs, que l'on pourrait comparer à des paquets de code logiciel, avec leur configuration, leurs bibliothèques système, leur environnement d'exécution et le reste. Les conteneurs contiennent tout ce dont l'application a besoin pour s'exécuter. Ils garantissent que les applications créées dans un environnement s'exécutent de manière cohérente et fluide dans d'autres environnements. Ils permettent ainsi d'exécuter un logiciel de manière fiable lorsqu'il est déplacé d'un environnement à l'autre. Ces unités modulaires (ou éléments de base) sont conçues pour permettre aux équipes DevOps de réaliser le build, tester, déployer et maintenir des applications efficacement, avec moins de ressources, extrêmement rapidement et en toute sécurité.

Puisque les conteneurs peuvent être facilement partagés entre plusieurs équipes, non seulement ils accélèrent le développement et le déploiement, mais ils favorisent également une culture de collaboration, qui est le fondement même du DevOps. Si vous savez comment fonctionnent les conteneurs, vous devez également connaître Docker, une technologie populaire de conteneurs pour les applications, et Kubernetes, un système d'orchestration de conteneurs open source qui régit comment et où les conteneurs s'exécutent. Le rapport Accelerate State of DevOps de 2021 a révélé qu'une bonne connaissance des conteneurs est un aspect essentiel du travail dans le DevOps, puisque 64 % des professionnels ont déclaré les utiliser, une tendance qui émerge dans les entreprises de toutes tailles.



Comment le DevOps résout les problèmes concrets

Si vous souhaitez comprendre à quel point l'adoption du DevOps peut être importante et révolutionnaire pour votre entreprise, jetez un œil à ce qu'une plateforme DevOps a pu faire pour [HackerOne](#). En tant que plateforme de sécurité la plus fiable au monde (qui est également créée par les hackers), HackerOne permet aux entreprises d'accéder à la plus grande communauté de hackers. Avec plus de 70 sites dans le monde, la collaboration transversale constituait un véritable défi pour l'entreprise. Par exemple, lorsque des développeurs de différents continents devaient poursuivre un projet de codage que d'autres avaient commencé, les longs délais d'exécution du pipeline interféraient avec le transfert. De plus, après avoir triplé la taille de son équipe d'ingénieurs, HackerOne avait besoin d'accélérer le développement et le déploiement, de réduire la complexité de sa chaîne d'outils et de permettre aux équipes de gérer efficacement plusieurs projets. [L'entreprise a donc opté pour une plateforme DevOps](#). Grâce à une plateforme unique et unifiée, les équipes de HackerOne ont pu détecter les problèmes du code plus tôt dans le pipeline, corriger de manière itérative les failles de sécurité et simplifier les audits. Elles ont également augmenté le nombre des déploiements d'une ou deux fois par jour à cinq fois par jour, économisant ainsi à chaque développeur quatre heures de développement par semaine.



La raison pour laquelle apprendre le DevOps est un atout pour votre carrière

Nous savons que cela représente beaucoup d'informations à assimiler, alors laissez-nous vous expliquer quelles sont les implications pour vous et votre carrière. Si vous avez récemment rejoint une équipe DevOps, nous avons une bonne nouvelle à vous annoncer : vous venez de prendre une excellente décision pour votre carrière. Si vous travaillez déjà dans l'informatique, vous êtes également sur la bonne voie. Des études ont montré que l'industrie technologique dans son ensemble est parvenue à rester forte face aux difficultés économiques, mais le DevOps en particulier est sur une pente ascendante. Selon les rapports d'organisations comme Glassdoor ou la société de conseil en ressources humaines Robert Half International, Inc., de nombreuses entreprises cherchent à embaucher des spécialistes du DevOps.

De plus, [les sociétés offrent une bonne rémunération](#) aux professionnels du DevOps qu'elles embauchent ou promeuvent à ces postes en interne. Amanda Stansell, Data Scientist chez Glassdoor, a classé les ingénieurs DevOps parmi les [10 meilleurs emplois en Amérique pour 2021](#) sur la base des revenus potentiels, de la satisfaction globale au travail et du nombre d'offres d'emploi répertoriées sur Glassdoor.

Mais quel est le salaire des spécialistes du DevOps ?

Les [salaires](#) varient considérablement en fonction du lieu et du niveau de compétence. Néanmoins, nous avons une bonne nouvelle pour les nouveaux venus dans le domaine du DevOps : si vous n'avez qu'un an d'expérience, vous pouvez toujours vous attendre à un salaire de 112 785 \$, et avec au moins 10 ans d'expérience, vous pouvez atteindre 165 980 \$. Ces chiffres sont globalement cohérents avec ceux de sites comme ZipRecruiter et Glassdoor. Alors, où vous situez-vous avec ce salaire par rapport à d'autres fonctions dans l'informatique ? Aux États-Unis, par exemple, les ingénieurs DevOps figurent parmi les 10 professionnels de l'informatique les mieux payés en 2021, selon Glassdoor.

Donc, si vous décrochez un emploi dans le DevOps, vous bénéficierez d'un bon salaire. Toutefois, quelle est la probabilité que vous puissiez [gravir les échelons](#) au sein de votre équipe DevOps ou trouver un emploi mieux rémunéré et offrant une meilleure culture et de meilleurs avantages dans une autre entreprise ? Selon les chiffres du [guide des salaires 2021 de Randstad](#), vos chances sont plutôt bonnes, car de nombreuses entreprises affirment que des postes dans leurs équipes DevOps restent vacants pendant de longues périodes (plus de 50 jours en moyenne).

Vérifier vos connaissances en DevOps 

Testez comment votre équipe utilise sa plateforme DevOps

Répondez à ce [questionnaire de 5 minutes](#) pour découvrir où en est votre équipe DevOps au niveau des efforts qu'elle déploie non seulement pour utiliser le DevOps, mais aussi pour exploiter efficacement la plateforme. Vous serez ensuite redirigé(e) vers des ressources qui peuvent aider votre équipe à pratiquer le meilleur DevOps possible.



Commencez votre essai gratuit de GitLab

Ressources

Voici quelques-unes des ressources à votre disposition (en anglais) :

Podcasts qui vous aideront à découvrir le monde du DevOps

- [The Humans of DevOps Podcast Series](#) offre un aperçu de sujets tels que l'amélioration des compétences, l'art du DevOps et les femmes travaillant dans ce domaine.
- [Arrested DevOps](#) sont des entretiens avec des technophiles qualifiés sur l'état actuel du DevOps.
- [Real World DevOps](#) sont des entretiens avec des organisateurs de conférences DevOps, des auteurs de livres sur le sujet et des développeurs technologiques.
- [The Cloudcast](#) porte bien son nom : tout tourne autour du cloud.
- [Greater Than Code](#) se concentre à la fois sur les problématiques humaines et technologiques dans le DevOps, mais aborde également le domaine des technologies en général.
- [Code Newbie Podcast](#) est destiné aux personnes qui débutent dans le développement de logiciels.
- [DevOps Paradox](#) fait appel à des sommités du secteur pour vous expliquer ce qu'est le DevOps.

Livres et eBooks utiles

- [Seven Tips to Get the Most out of Your DevOps Platform](#) est un eBook de GitLab qui a pour but de permettre aux équipes de tirer le meilleur parti d'une plateforme DevOps.
- [Continuous Delivery](#) a été qualifié d'ouvrage « incontournable » pour quiconque souhaite relier l'ensemble du processus de développement et de livraison.
- [Practical DevOps](#) explique comment fonctionne le DevOps, puis aborde le stockage du code, les tests et le déploiement.
- [The DevOps Handbook](#) est considéré comme un ouvrage de référence pour tous ceux qui travaillent dans ce domaine. Il évoque non seulement les avantages du DevOps, mais explique également comment il peut offrir aux entreprises un avantage concurrentiel.
- [GitLab Quick Start Guide](#) est un eBook et un excellent guide pour passer à la plateforme GitLab.
- [Big Little Book on Git](#) est un eBook qui s'adresse aussi bien aux professionnels qu'aux novices du DevOps.
- [Ten Steps Every CISO Should Take to Secure Next Generation Software](#) est une introduction destinée aux professionnels de la sécurité qui explique comment les changements dans le développement de logiciels affectent les programmes de sécurité.

Certifications

- [Le DevOps Institute propose des certifications](#) dans des domaines tels que le développement, l'ingénierie DevOps, les tests DevOps et l'ingénierie de la sécurité.
- [GitLab possède ses propres certifications](#) dans des domaines tels que la CI/CD, la gestion de projet et la sécurité DevOps.
- Consultez toujours la source. Par exemple, si vous souhaitez apprendre à utiliser Google Cloud, recherchez [les certifications possibles sur le site de l'entreprise](#).



Ateliers et cours

- La plateforme de formation en ligne [A Cloud Guru](#) propose des certifications cloud moyennant des frais mensuels. Elle est structurée pour fournir aux utilisateurs du contenu vidéo, des exercices pratiques, des outils d'apprentissage, des questionnaires et des examens.
- [DevOps Foundations de LinkedIn Learning](#) offre une base de connaissances solide pour ceux qui abordent le DevOps pour la première fois, ou presque. Disponibles gratuitement pour toute personne disposant d'un abonnement LinkedIn, les vidéos donnent un aperçu de l'industrie ainsi que des principes et technologies clés comme l'automatisation, la collaboration, la surveillance et la culture.
- [Le DevOps Implementation Boot Camp](#), organisé par la société de conseil Cprime, est un cours de trois jours à partir de 1 695 \$. Il est disponible en présentiel ou en direct en ligne. Des formations privées en équipe sont également possibles.
- [DevOps Culture and Mindset](#) est proposé par l'Université de Californie à Davis via Coursera, un fournisseur de cours en ligne. Le cours d'environ 15 heures est entièrement en ligne et porte sur les principes fondamentaux du DevOps. Il est gratuit pour ceux qui disposent d'un abonnement à Coursera.
- [Continuous Delivery & DevOps](#) est un cours en ligne de 8 heures pour débutants proposé par l'Université de Virginie via Coursera. Il se concentre sur les thèmes de la livraison continue, des tests et de l'Infrastructure-as-code. Il est gratuit pour ceux qui disposent d'un abonnement à Coursera.

À propos de GitLab

GitLab propose une plateforme DevOps qui permet aux entreprises d'optimiser leurs investissements dans le développement logiciel en livrant les logiciels plus vite et efficacement, tout en renforçant la sécurité et la conformité. L'application unique de GitLab est facile à utiliser, accélère les cycles et vous offre une visibilité et un contrôle sur toutes les étapes du cycle de vie DevOps. Grâce à GitLab, toutes les équipes de votre entreprise ont la possibilité de planifier, créer, sécuriser et déployer des logiciels de manière collaborative afin d'atteindre plus rapidement les objectifs métier en toute transparence, cohérence et traçabilité.

GitLab est une plateforme open source qui s'appuie sur sa communauté grandissante, constituée de milliers de développeurs et de millions d'utilisateurs, pour proposer en permanence des innovations dans le domaine du DevOps. Plus de 30 millions d'utilisateurs enregistrés (comptes payants et gratuits confondus), allant de start-ups aux multinationales, comme Ticketmaster, Jaguar Land Rover, Nasdaq, Dish Network et Comcast, font confiance à GitLab pour créer de formidables logiciels plus rapidement. Avec son modèle 100 % télétravail depuis sa création, GitLab compte aujourd'hui plus de 1 350 collaborateurs répartis dans plus de 65 pays.





GitLab