

Comment simplifier les processus de déploiement avec DevOps

DevOps

Dans ce guide

- CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications
 - Les tests en production : et si cela était bénéfique
 - DevOps : 10 catégories d'outils approuvés par les entreprises
 - Comment la technique Blue/Green garantit un déploiement cohérent et continu
-

Introduction.

Pourquoi donc accélérer les processus de développement si l'on n'accélère pas la mise en production du code ?

C'est à cette question que doit répondre le déploiement continu et l'ensemble d'un processus DevOps. Avec l'émergence de nouvelles architectures applicatives, comme les microservices, les containers et le cloud, il est indispensable de revoir la façon dont sont déployées ces applications.

A infrastructure moderne, processus modernes.

Reste que l'accélération des déploiements ne doit pas omettre les étapes clés qui garantissent un code propre et un déploiement sûr et efficace. Avec DevOps, les équipes fonctionnent par courtes itérations : le code est développé de façon incrémentale et poussé en production lorsque le composant est validé et conforme.

Dans ce guide

- CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications
 - Les tests en production : et si cela était bénéfique
 - DevOps : 10 catégories d'outils approuvés par les entreprises
 - Comment la technique Blue/Green garantit un déploiement cohérent et continu
-

Une démarche continue qui tranche avec l'approche plus lourde du cycle en V où l'ensemble des versions de code est déployé dans une unique et lourde mise à jour.

Il faut donc s'adapter. Si cela vise aussi à simplifier les phases de déploiement, un changement culturel et de mentalité doit être appliqué. Ce guide essentiel donne les bases pour cela.



Dans ce guide

- CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications
 - Les tests en production : et si cela était bénéfique
 - DevOps : 10 catégories d'outils approuvés par les entreprises
 - Comment la technique Blue/Green garantit un déploiement cohérent et continu
-

■ CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications

Tom Nolle, CIMI Corporation

Les entreprises qui s'attaquent à un vaste projet de modernisation d'applications doivent intégrer 4 éléments dans leur stratégie : la livraison continue (CD), DevOps, les conteneurs et le cloud. La réussite dépend du sens dans lequel ces composants seront mis en place.

Le choix entre [conteneurisation](#), [DevOps](#) et migration vers le cloud a de quoi refroidir nombre de directions informatiques, dans leurs projets de modernisation d'applications. La raison : ils évaluent d'abord les options technologiques avant même de déterminer l'objectif.

Si évidemment, les applications doivent subir une forme de modernisation, on sait que cela est notoirement difficile à exécuter. Les entreprises se plaignent par exemple que les objectifs sont trop techniques, et que les coûts dépassent généralement ce qui était prévu.

Dans ce guide

- CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications
 - Les tests en production : et si cela était bénéfique
 - DevOps : 10 catégories d'outils approuvés par les entreprises
 - Comment la technique Blue/Green garantit un déploiement cohérent et continu
-

Les projets de modernisation d'applications sont alors souvent au centre d'une vision technique en tunnel, déconnectés des potentiels gains et ne sont plus pensés comme une façon d'unifier et de d'orienter l'architecture applicative.

Les bons projets de modernisation naissent en général de la volonté d'améliorer la productivité, de réduire les coûts et les erreurs. L'intégration et la livraison en continu (CI/CD), DevOps et le [cloud ou les conteneurs - ou conteneurs dans le cloud](#) – dépendent avant tout des objectifs du projet, de la structure de l'entreprise et de l'architecture des applications.

CI/CD pour la modernisation de toutes les applications

Nombre d'entreprises estiment que la modernisation d'applications qui s'appuient sur des processus de CI/CD est utile car cela réduit le temps nécessaire pour injecter de nouvelles fonctions, et au final améliore le support IT des opérations commerciales. L'amélioration de la productivité passe inévitablement par un meilleur support IT.

L'automatisation des mises à jour via CI/CD permet d'intégrer efficacement aux projets les autres composantes clés : conteneurisation (ou containerisation en anglais), DevOps et la migration vers le cloud. Cependant, CI/CD change les

Dans ce guide

- CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications
 - Les tests en production : et si cela était bénéfique
 - DevOps : 10 catégories d'outils approuvés par les entreprises
 - Comment la technique Blue/Green garantit un déploiement cohérent et continu
-

pratiques de développement pour nécessiter aussi une refonte de l'application. Ces processus en continu peuvent également déboucher sur des problèmes d'exploitation.

Quand les coûts entraînent la modernisation

La modernisation d'applications, quand elle est pilotée par les coûts, a provoqué nombre de débats dans les DSI. L'adoption de DevOps, des conteneurs (ou containers) et du cloud public contribuent certes à réduire les coûts IT, mais peu d'entreprises ont toutefois été en mesure d'élaborer une stratégie qui prend en compte toutes ces options. Beaucoup finissent par se concentrer uniquement sur les mauvais points.

Si l'on choisit d'amorcer une modernisation des applications par la question des coûts, [il convient de commencer par DevOps](#) - déploiement et redéploiement automatisé. Si l'entreprise a déjà des méthodologies DevOps en place, elle risque de les conserver. Dans le cas contraire, il existe beaucoup d'outils de gestion de configuration et d'automatisation, comme [Chef](#) et [Puppet](#).

La réussite de DevOps est de penser en termes d'états et d'événements applicatifs, que l'outil choisi soit prescriptif ou déclaratif. Les applications doivent avoir un petit nombre d'états fonctionnels spécifiques - comme une charge normale, une charge lourde ou une altération. Et un ensemble de conditions – qui associent les composants entre eux, créent des connexions et les

Dans ce guide

- CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications
 - Les tests en production : et si cela était bénéfique
 - DevOps : 10 catégories d'outils approuvés par les entreprises
 - Comment la technique Blue/Green garantit un déploiement cohérent et continu
-

ressources d'hébergement – doit y être relié. L'outil DevOps doit définir les conditions souhaitées ou les étapes nécessaires à atteindre, selon que l'outil est déclaratif ou prescriptif, respectivement. Les événements signalent un changement d'état de fonctionnement et déclenchent donc une activité DevOps.

Les états et les événements sont également clés pour les processus de CI/CD. Dans les workflows de CD, DevOps doit refléter la manière dont les nouvelles fonctionnalités sont développées et déployées, ce qui implique de se concentrer sur la manière dont les fonctionnalités s'adaptent aux composants logiciels.

Qu'il s'agisse d'une modernisation axée sur la productivité ou sur les coûts, une évaluation de DevOps identifiera les éléments réels qui seront déployés et redéployés. Et c'est là que se pose la question cruciale de la décomposition de l'application.

Bien que cela doive être un élément clés dans une stratégie de CD, il est aussi possible d'appliquer ces processus CI/CD sans avoir à restructurer les applications. Mais Il n'est pas possible de passer à des processus DevOps sans décomposition des applications.

Dans ce guide

- CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications
 - Les tests en production : et si cela était bénéfique
 - DevOps : 10 catégories d'outils approuvés par les entreprises
 - Comment la technique Blue/Green garantit un déploiement cohérent et continu
-

DevOps, cloud et déploiement de conteneurs

La façon dont on utilise DevOps pour le déploiement et le redéploiement de composants, [et le recours aux microservices](#), est symbolique des orientations futures en matière de modernisation applicative.

Les applications avec des structures front et back end distinctes qui requièrent donc agilité et résilience ont tout à gagner d'une configuration distincte : des services cloud publics pour la partie frontale et des conteneurs pour le back-end.

La DSI doit ensuite trouver des outils de gestion du cloud qui intègrent les éléments frontaux du cloud public au back-end du datacenter. Ils doivent également être compatibles avec les processus DevOps en place. En back-end, les [options de déploiement des conteneurs](#) doivent quant à elles s'intégrer facilement aux opérations IT en cours.

Quel que soit l'ordre dans lequel vous considérez les conteneurs, le cloud public et DevOps, il est probable qu'ils joueront tous un rôle dans le projet de modernisation. Mais chaque choix technologique doit s'adapter aux autres.

Dans ce guide

- CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications
 - Les tests en production : et si cela était bénéfique
 - DevOps : 10 catégories d'outils approuvés par les entreprises
 - Comment la technique Blue/Green garantit un déploiement cohérent et continu
-

■ Les tests en production : et si cela était bénéfique

Gerie Owen, QualiTest

À une époque où les tests manuels se retrouvent limités en temps, il est l'heure de penser à des alternatives. L'experte Gerie Owen explique pourquoi les tests en production sont essentiels.

Tester en production ? Vraiment ? Les tests en production - parfois appelés tests synthétiques - s'appliquent presque toujours aux applications Web, afin de garantir un comportement, des performances et un niveau de disponibilité adaptés en fonction de l'utilisation en production d'une application.

Les tests synthétiques ne sont pas très différents des tests automatisés. [Vous enregistrez des séquences de tests](#) en accédant à des pages Web et en entrant des données pour simuler des transactions courantes, puis vous les exécutez de façon régulière. L'objectif est de s'assurer que ces applications sont disponibles quand il le faut et qu'elles répondent aux normes du fournisseur. Ces tests servent également à vérifier que les résultats de chaque étape sont conformes aux attentes.

Dans ce guide

- CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications
 - Les tests en production : et si cela était bénéfique
 - DevOps : 10 catégories d'outils approuvés par les entreprises
 - Comment la technique Blue/Green garantit un déploiement cohérent et continu
-

Une des différences clés est que ces types de tests sont exécutés alors que l'application est en cours d'exécution, face aux clients. Dynatrace, Catchpoint Systems et New Relic font partie de ce marché. Vous pouvez utiliser ces services de test à partir de points de présence mondiale, en simulant des clients réels ou des utilisateurs à partir de leur localisation géographique.

Ces tests peuvent également être utilisés pour surcharger une application et déterminer si cela affecte les performances. En créant de nombreuses instances de test, reposant sur divers types de tests géographiquement répartis, on peut simuler le comportement de l'utilisateur avec beaucoup plus de précision que dans un laboratoire.

Les résultats de ces tests dits synthétiques sont généralement affichés sur le tableau de bord du centre de contrôle des opérations réseau de l'entreprise ou dans un outil de monitoring. Les testeurs effectuent leurs tests dans le monde entier et tout au long de la journée, surveillent les signes d'un comportement inhabituel.

Dans ce guide

- CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications
 - Les tests en production : et si cela était bénéfique
 - DevOps : 10 catégories d'outils approuvés par les entreprises
 - Comment la technique Blue/Green garantit un déploiement cohérent et continu
-

Les tests en production ne disparaissent pas

Comme vous pouvez l'imaginer, les tests en production sont controversés chez les testeurs traditionnels. Il existe une forte culture selon laquelle il est préférable d'effectuer les tests avant qu'une application ne soit placée en production. Une fois ces essais terminés, les testeurs sont plus ou moins assurés que l'application ne contient pas de défauts critiques et qu'elle se comportera comme prévu en production.

Cependant, il est impossible de maîtriser l'environnement de production pendant les essais de préproduction. Cela entraîne des défaillances qui se manifestent une fois l'interaction avec les utilisateurs réels effectuée. Si le déploiement se fait sur une ou plusieurs instances cloud, par exemple, [le comportement des services cloud utilisés peut aussi affecter la qualité](#) – lors d'une dégradation des performances dans un datacenter par exemple.

Les ingénieurs réseau ont besoin d'alertes, afin qu'ils puissent prendre des mesures. Les alertes peuvent être par exemple des notifications si un test ou un chargement de pages individuelles prend beaucoup plus de temps que d'habitude. Ces ingénieurs ont également accès à des graphiques sur les performances et les niveaux de disponibilité. Une seule défaillance n'est généralement pas une priorité, mais une série de défaillances ou une

Dans ce guide

- CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications
 - Les tests en production : et si cela était bénéfique
 - DevOps : 10 catégories d'outils approuvés par les entreprises
 - Comment la technique Blue/Green garantit un déploiement cohérent et continu
-

dégradation régulière des performances sont généralement la première indication d'un problème au niveau de l'application.

Les tests en production offrent aussi la possibilité d'identifier des problèmes sporadiques au niveau du réseau. Ces problèmes de disponibilité et de performance peuvent être réglés pendant la production (équilibre de charge, routage, etc.).

Ceux qui effectuent ces tests sont généralement des ingénieurs réseau possédant les compétences nécessaires pour comprendre le fonctionnement réseau et serveurs. Ce qui leur manque, c'est la compréhension de la valeur d'usage de l'application et du comportement des utilisateurs.

Les testeurs traditionnels peuvent travailler à partir d'exigences et de retours utilisateurs pour concevoir des batteries de tests en production. Ils peuvent également développer des tests conçus pour sonder uniquement les points faibles d'une application.

Dans ce guide

- CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications
 - Les tests en production : et si cela était bénéfique
 - DevOps : 10 catégories d'outils approuvés par les entreprises
 - Comment la technique Blue/Green garantit un déploiement cohérent et continu
-

■ DevOps : 10 catégories d'outils approuvés par les entreprises

Beth Pariseau, journaliste

Les solutions comportant une étiquette DevOps sont partout, et la liste des options s'allonge chaque jour. Toutefois, les meilleurs outils DevOps sont déjà bien connus des professionnels de l'IT. Cet article liste 10 outils déjà passés au crible et validés par le secteur.

Depuis la formulation du concept [DevOps](#) par le Phoenix Project, les outils ont beaucoup mûri, et les professionnels de l'IT ont atteint un consensus et se sont entendus sur un point : les 10 catégories d'outils véritablement essentielles dans la mise en place d'un processus DevOps.

Dans certaines de ces 10 grandes catégories, un seul outil est utilisé, comme Docker pour les containers d'application ou Jenkins pour CI/CD (Continuous Integration / Continuous Delivery). Dans d'autres, certains sont considérés comme interchangeables par les spécialistes DevOps, et sont choisis en fonction des préférences personnelles et des besoins spécifiques.

Dans ce guide

- CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications
 - Les tests en production : et si cela était bénéfique
 - DevOps : 10 catégories d'outils approuvés par les entreprises
 - Comment la technique Blue/Green garantit un déploiement cohérent et continu
-

1. Repository de code source

La première étape d'un processus DevOps est le référentiel de code source et le contrôle de versions où les développeurs posent, vérifient et gèrent le code. La plupart des outils de déploiement et d'intégration continue réagissent automatiquement aux commits de code dans de tels référentiels. Pour les spécialistes du genre, un processus DevOps qui ne commence pas par un contrôle du code source ne mène nulle part.

"Si vous ne pouvez pas codifier ce que vous faites, inutile de le faire", explique Brad Linder, évangéliste de DevOps et données chez Dish Technologies, la branche ingénierie de Dish Network à Englewood, Colo.

Git et le service SaaS, GitHub, sont les outils de dépôt de code source les plus utilisés. GitLab, un autre gestionnaire de référentiel Git, propose également un paquet d'outils de pipeline CI/CD et un registre Docker pour les containers.

2. Dépôt d'artefacts

Le code source est la matière première d'une application. Ce qui ressort de ce code est un artefact. Les artefacts doivent également être gérés et leurs versions contrôlées. Pour cela, la plupart des équipes DevOps préfèrent JFrog Artifactory ou Nexus Repository.

« Pour la rapidité des releases, Il est important que les choses soient simples », assure Gary McKay, Scrum Master chez Somos, un organisme à but non

Dans ce guide

- CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications
 - Les tests en production : et si cela était bénéfique
 - DevOps : 10 catégories d'outils approuvés par les entreprises
 - Comment la technique Blue/Green garantit un déploiement cohérent et continu
-

lucratif qui gère les dossiers de numéros de téléphone gratuits. « Si j'ai 70 à 80 builds par jour, il est facile d'aller chez Artifactory et d'en extraire une. Comparer les builds est plus facile, et ça vous donne de la flexibilité.

3. CI/CD

Il existe une pléthore [d'outils de CI/CD sur le marché](#) qui automatisent les tests des modifications de code source et assemblent les applications à partir d'artefacts. Pour la plupart des spécialistes DevOps, cependant, Jenkins reste la norme. Jenkins n'est pas connu pour sa facilité d'administration, et beaucoup d'utilisateurs admettent avoir une relation « je t'aime, moi non plus » avec lui, surtout dans les versions plus anciennes.

« Jenkins est comme une paire de vieilles chaussures confortables dont on ne peut pas se débarrasser, et elle ne donne une ampoule qu'à un seul endroit », ironise Brad Linder. « Son écosystème est très vaste. Les nouveaux modèles sont compliqués, mais ça fonctionne. »

Aucun autre outil pour DevOps ne peut égaler Jenkins, soutient à son tour Richard Fong, directeur de l'ingénierie logicielle chez Mitchell International, une société spécialisée dans les logiciels pour l'assurance automobile. Il a mis en place l'un des premiers environnements Jenkins en production chez Yahoo en 2003, et utilise toujours l'outil actuellement.

4. Containers

Dans ce guide

- CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications
 - Les tests en production : et si cela était bénéfique
 - DevOps : 10 catégories d'outils approuvés par les entreprises
 - Comment la technique Blue/Green garantit un déploiement cohérent et continu
-

Les containers font fureur chez les entreprises qui veulent mettre en place DevOps. Ils font la promesse de la portabilité du code applicatif, depuis les portables des développeurs vers les environnements de développement et de test et, théoriquement, jusqu' à la production. Ils constituent également un moyen efficace pour déployer des micro-services. Les containers peuvent être utilisés pour déployer rapidement mais également de façon sûre des applications.

« Sans containers, vous mettez en place des serveurs pour effectuer ce travail et vous configurez des load balancers pour acheminer un pourcentage du trafic », explique de son côté Andy Domeier, directeur des opérations chez SPS Commerce, spécialisée dans les communications pour la supply chain. « Les containers facilitent la mise en production grâce à la portabilité entre environnements. »

Ici, le choix d'outils est en grande partie évident : Docker domine le segment des containers depuis des années déjà. Les équipes DevOps apprécient aussi les outils de la société Docker comme Docker Compose, qui définit et exécute des applications Docker multi-container. Selon eux, cela facilite la productivité des développeurs car ils n'ont pas à comprendre toutes les spécificités de l'environnement du container avant d'écrire le code.

5. Orchestration de containers

Dans ce guide

- CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications
 - Les tests en production : et si cela était bénéfique
 - DevOps : 10 catégories d'outils approuvés par les entreprises
 - Comment la technique Blue/Green garantit un déploiement cohérent et continu
-

Les containers ont besoin d'un outil pour les contrôler et automatiser leur déploiement et leur redéploiement. Ici, Kubernetes est apparu comme le standard de facto pour l'orchestration de containers, et ce, bien que la plupart des entreprises ne déploient pas encore de containers en production. Elles n'ont également pas exclu le logiciel Docker Enterprise Edition de Docker comme alternative.

« Certains pensent que Docker [Enterprise Edition] n'est pas aussi avancé [que Kubernetes], mais Docker a beaucoup travaillé ces 18 derniers mois en matière de fonctions de découverte automatique, d'équilibrage de charge et d'autoscaling », commente Richard Fong. « A notre échelle, ça fonctionne avec Docker. »

6. Infrastructure-as-code

Exécuter des clusters de containers à l'échelle nécessite également un moyen d'automatiser l'infrastructure sous-jacente. Ici, les équipes IT se tournent vers des outils d'infrastructure-as-code. Les plus cités sont Terraform de HashiCorp et Ansible de Red Hat.

Ansible est mieux connu pour ses fonctionnalités de gestion de configuration, mais les entreprises l'utilisent également pour le provisioning d'infrastructures.

« Ansible est flexible, une sorte de couteau suisse », lance Thomas Davis, directeur de la sécurité chez ServiceMaster Global Holdings. ServiceMaster, qui

Dans ce guide

- CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications
 - Les tests en production : et si cela était bénéfique
 - DevOps : 10 catégories d'outils approuvés par les entreprises
 - Comment la technique Blue/Green garantit un déploiement cohérent et continu
-

possède des marques grand public telles que Terminix, Merry Maids, Furniture Medic et ServiceMaster Clean and Restore, utilise Ansible pour suivre les modifications des adresses IP dans son environnement AWS, lors des mises à jour des applications de l'entreprise.

Ansible peut également créer des versions de templates d'images serveur, les mettre à jour à la volée et en contrôler les versions, ajoute Richard Fong.

7. Gestion de la configuration

Une fois les serveurs provisionnés, l'équipe DevOps doit configurer l'infrastructure pour exécuter l'application. Dans ce domaine, les professionnels ont un large choix. Ils ont d'ailleurs tendance à utiliser un mixe d'outils, comme [Puppet](#), [Chef](#), Ansible, [SaltStack](#), parmi tant d'autres.

« Avec Ansible, on peut créer des templates d'infrastructure, mais ce n'est pas son vrai rôle », affirme Gary McKay. « Son vrai rôle vient après que vous ayez créé un environnement avec Terraform, et que vous utilisiez Ansible pour configurer un serveur web pour exécuter JBoss et la gateway API de WSO2, par exemple. »

8. Monitoring de logs

Dans ce guide

- CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications
 - Les tests en production : et si cela était bénéfique
 - DevOps : 10 catégories d'outils approuvés par les entreprises
 - Comment la technique Blue/Green garantit un déploiement cohérent et continu
-

Certains outils de monitoring DevOps compilent logs et métriques en une seule interface, mais la plupart des équipes DevOps utilisent encore des outils distincts.

La stack ELK (Elasticsearch, Logstash et Kibana) est ici souvent utilisée, car cela est gratuit – même s'il nécessite beaucoup de main d'œuvre et d'expertise. Les outils propriétaires, comme Splunk, valent donc l'investissement pour certaines entreprises.

« Nous sommes prêts à payer pour avoir Splunk », illustre Kevin Burnett, responsable DevOps chez Rosetta Stone, une société spécialisée dans les logiciels éducatifs.

9. Monitoring et indicateurs

Certains spécialistes DevOps considèrent que le monitoring d'indicateurs et d'événements ne sont pas aussi essentiels que le monitoring des logs. Toutefois, la plupart d'entre eux en possède. Le monitoring active la boucle d'interaction propre à DevOps.

« Pour vous distinguer tant qu'ingénieur DevOps, il faut améliorer l'information transmise à l'équipe produit », déclare Nathaniel Felsen, ingénieur DevOps chez Medium.com, la plateforme de blogs. Donner des indicateurs sur la performance des applications en production permet à tous les membres de l'équipe DevOps d'améliorer leur travail.

Dans ce guide

- CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications
 - Les tests en production : et si cela était bénéfique
 - DevOps : 10 catégories d'outils approuvés par les entreprises
 - Comment la technique Blue/Green garantit un déploiement cohérent et continu
-

Certains se tournent vers des outils connus tels qu'AppDynamics App IQ et New Relic pour rapprocher les équipes de développement et les opérationnels et mieux répondre aux incidents DevOps.

L'application Open Source Prometheus est quant à elle un favori pour monitorer les clusters Kubernetes. Les outils de surveillance d'indicateurs jettent également les bases de l'AIOps, une approche de gestion automatisée de l'infrastructure qui repose sur le Machine Learning et des techniques d'intelligence artificielle.

10. ChatOps

Les équipes DevOps doivent [pouvoir communiquer efficacement pour résoudre les problèmes de production](#). Ici, les outils ChatOps de Slack et HipChat d'Atlassian sont les plus populaires.

Slack a une version gratuite, tandis que HipChat offre des fonctionnalités d'audit d'entreprise et des intégrations avec les systèmes de ticketing et de gestion de projets tels que Jira d'Atlassian. Les ChatOps se reposent sur des chatbots qui envoient des alertes aux utilisateurs, et exécutent les modifications à effectuer à partir des commandes émises dans l'interface de chat.

Pour les très grandes entreprises, les canaux ChatOps ne sont pas suffisants - elles doivent également organiser et acheminer des alertes de performance

Dans ce guide

- CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications
 - Les tests en production : et si cela était bénéfique
 - DevOps : 10 catégories d'outils approuvés par les entreprises
 - Comment la technique Blue/Green garantit un déploiement cohérent et continu
-

entre plusieurs équipes, parfois réparties dans plusieurs pays. PagerDuty est par exemple devenu un complément des ChatOps.

« La culture DevOps conduit à donner aux ingénieurs une plus grande responsabilité. Mais lorsque ceux qui développent les applications doivent également les maintenir, cela augmente le nombre de personnes disponibles pour répondre aux besoins », explique encore Andy Domeier (SPS).

Dans ce guide

- CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications

- Les tests en production : et si cela était bénéfique

- DevOps : 10 catégories d'outils approuvés par les entreprises

- Comment la technique Blue/Green garantit un déploiement cohérent et continu

■ Comment la technique Blue/Green garantit un déploiement cohérent et continu

Kurt Marko, Consultant MarkoInsights

Dans le cloud, le déploiement d'applications fonctionne différemment que sur l'infrastructure physique. Le modèle de déploiement Blue Green comporte quelques variantes, chacune d'entre elles permettant de gagner du temps et d'éviter les erreurs.

L'essor Avec l'isolation du système, la facilité de réplication et le déploiement rapide, l'infrastructure virtuelle est le moyen idéal pour effectuer des tests de préproduction, l'intégration du système et la mise en production. Elle est devenue la base des modèles de déploiement Canary et Blue/Green. Ces techniques exigent un effort important de la part de [DevOps](#) et des équipes réseau qui gèrent les serveurs virtuels en interne. Les services cloud atténuent ces problèmes.

L'[IaaS](#) amplifie les avantages de l'infrastructure virtuelle, car il supprime les opérations liées à l'infrastructure et place une couche d'abstraction au niveau du service entre l'utilisateur et les ressources physiques.

Dans ce guide

- CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications
 - Les tests en production : et si cela était bénéfique
 - DevOps : 10 catégories d'outils approuvés par les entreprises
 - Comment la technique Blue/Green garantit un déploiement cohérent et continu
-

Les services cloud - depuis Amazon Elastic Compute Cloud (EC2) et Elastic Block Store jusqu'à Microsoft Azure Cosmos DB, Google Cloud Storage et d'autres - éliminent les opérations de provisioning et fournissent une couche d'API qui facilite l'automatisation et la réutilisation des tâches.

Les services cloud sont une plate-forme idéale pour définir des modèles de déploiement d'applications à haute disponibilité. En outre, leurs services de développement peuvent aussi automatiser entièrement les processus de déploiement dans de multiples environnements de test et de production.

Déploiement dans le cloud : quelques modèles

L'essor Les stratégies de déploiement sont certes spécifiques à chaque entreprise ou application, mais cinq catégories se dégagent – [comme l'a d'ailleurs identifié un expert du secteur](#) :

- **Déploiement Rip-and-Replace** : Les équipes opérationnelles connaissent bien l'approche Rip-and-Replace (on jette et on remplace), remplacer le vieux par le neuf. Cette méthode de déploiement fonctionne mieux avec une infrastructure stable dans le temps.
- **Mise à jour en continu (« Rolling upgrade »)** : Le nouveau code est inséré dans l'ancien, dans un environnement unique. Étant donné la

Dans ce guide

■ CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications

■ Les tests en production : et si cela était bénéfique

■ DevOps : 10 catégories d'outils approuvés par les entreprises

■ Comment la technique Blue/Green garantit un déploiement cohérent et continu

nature graduelle d'une mise à niveau continue, les utilisateurs passent progressivement, plutôt qu'en une seule fois, au nouveau code.

- **Déploiement Blue/Green** : Ce modèle de déploiement implique la mise en place de deux infrastructures parallèles - l'ancien et le nouvel environnement. Les équipes opérationnelles assurent une transition du nombre d'utilisateurs de l'ancien au nouveau, après une période de test bêta.
- **Déploiement « Canary »** : Identique au modèle de déploiement Blue/Green, le test Canary exécute en même temps le nouveau et l'ancien code. Cependant, seul un petit groupe d'utilisateurs ont accès à la nouvelle version pour effectuer des tests en direct.
- **Déploiement par versions** : Cette variante du modèle de déploiement Canary maintient différentes versions de code indéfiniment, les utilisateurs ayant la possibilité de choisir une version.

À l'exception des mises à jour Rip-and-Replace, ces stratégies impliquent un déploiement parallèle, en ce sens qu'elles nécessitent d'héberger simultanément différentes versions d'applications dans un environnement de production. Le clonage de ressources est particulièrement facile avec les services cloud.

Dans ce guide

- CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications
 - Les tests en production : et si cela était bénéfique
 - DevOps : 10 catégories d'outils approuvés par les entreprises
 - Comment la technique Blue/Green garantit un déploiement cohérent et continu
-

Quels sont les services cloud pour automatiser ces déploiements en parallèle ?

Un modèle de déploiement Blue/Green est peut-être l'option la plus utilisée pour les applications virtualisées. Les modèles Canary et par versions sont des variantes de cette approche.

La configuration Blue/Green ne nécessite presque pas de temps d'arrêt, et avec les services cloud, il existe un moyen simple de passer d'une version à l'autre, à la fois pour la mise à niveau et le roll back. Il est également simple de répliquer complètement un environnement de production pendant le test. Le déploiement Blue/Green sur le cloud n'exige aucun investissement en capital.

[AWS](#) est représentatif des services cloud.

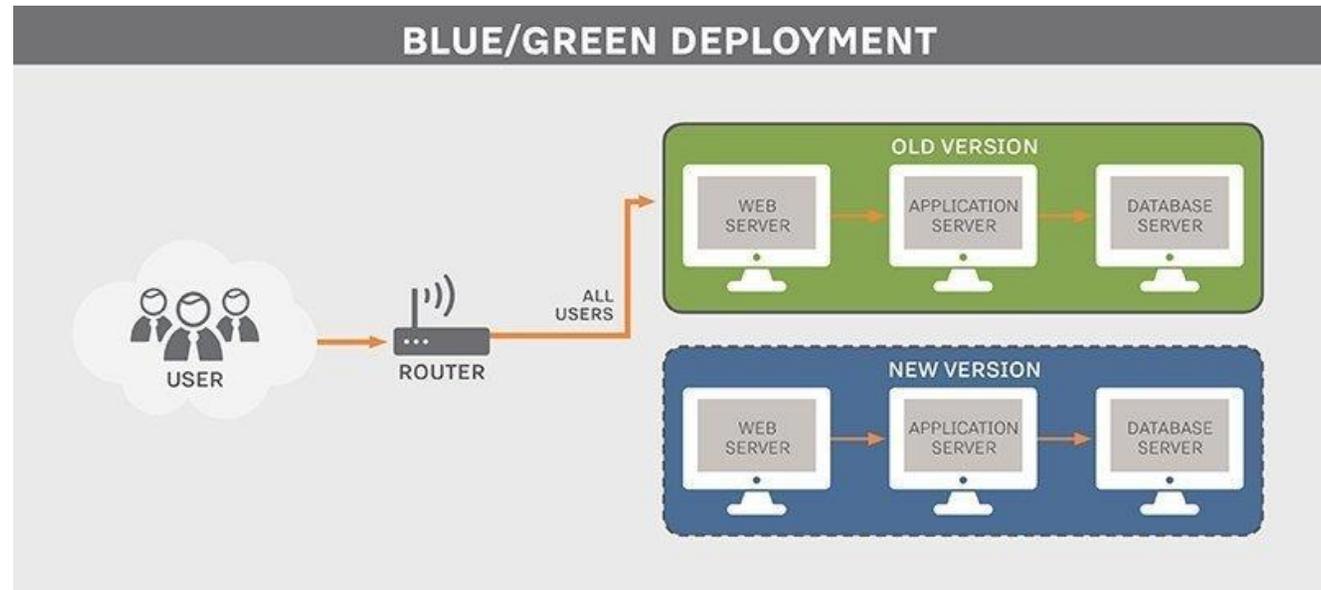
Dans ce guide

- CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications

- Les tests en production : et si cela était bénéfique

- DevOps : 10 catégories d'outils approuvés par les entreprises

- Comment la technique Blue/Green garantit un déploiement cohérent et continu



AWS fournit quelques détails pour un déploiement Blue/Green. Il se repose sur plusieurs services cloud :

- DNS : Le DNS dirige le trafic utilisateur vers chaque environnement - par exemple, vers beta.myapp.io ou prod.myapp.io. Chez AWS, le service se nomme Route 53.
- Load balancing et mise à l'échelle : Le modèle de déploiement doit inclure une capacité de dimensionnement de l'infrastructure, surtout lorsque les testeurs augmentent la charge. AWS a deux outils de load balancing, Application Load Balancer et Network Load Balancer, et propose un service de dimensionnement automatique.

Dans ce guide

- CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications
 - Les tests en production : et si cela était bénéfique
 - DevOps : 10 catégories d'outils approuvés par les entreprises
 - Comment la technique Blue/Green garantit un déploiement cohérent et continu
-

- **Templates de ressources** : Les templates facilitent les déploiements automatisés, qui vont de pair avec des approches types, comme le modèle Blue/Green – CloudFormation chez AWS.
- **Orchestration des ressources** : Les administrateurs de cloud ont la possibilité d'utiliser des déploiements programmables, notamment pour cloner des environnements complets. Par exemple, AWS propose OpsWorks et Elastic Beanstalk. Elastic Beanstalk, qui supporte Java, PHP, .NET, Node.js, Python et Ruby, peut cloner un environnement en un seul clic à partir de l'AWS Management Console.
- **Monitoring de service** : Le déploiement n'est qu'un début. Le code actif doit être contrôlé pour s'assurer que les indicateurs restent dans les limites établies. En cas de pannes de service ou d'anomalies au niveau des performances, des alertes doivent se déclencher. AWS propose CloudWatch pour cela.

Une fois les environnements Blue/Green configurés, il convient de changer celui qui est en production en direct (green) via un changement de DNS. Il s'agit d'une opération simple sur la console AWS via Route 53.

En raison de la mise en cache DNS et des paramètres TTL, les transitions DNS sont graduelles, à mesure que l'ancienne adresse (Blue) est remplacée par la nouvelle adresse IP sur les serveurs. Pour un transfert plus rapide, l'administrateur peut utiliser Elastic Load Balancing (ELB) et Auto Scaling en

Dans ce guide

- CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications
 - Les tests en production : et si cela était bénéfique
 - DevOps : 10 catégories d'outils approuvés par les entreprises
 - Comment la technique Blue/Green garantit un déploiement cohérent et continu
-

attachant le groupe Green au pool ELB de production et en le dimensionnant, tout en décommissionnant les instances du groupe Blue.

Automatiser le cycle de déploiement Blue/Green

Les équipes DevOps qui ont adopté les pipelines CI/CD peuvent automatiser le modèle de déploiement Blue/Green via un service de déploiement programmable. Sur AWS, ce produit s'appelle CodeDeploy (pour EC2) et l'option cloud de type serverless Lambda. Il intègre le type Blue/Green. Un service de déploiement programmable repose sur le load balancing et le dimensionnement du fournisseur de cloud pour répliquer les environnements et assurer la transition Blue et Green.

Chez AWS, on peut configurer CodeDeploy pour un mode Blue/Green à l'aide de la console AWS. Le service clone automatiquement l'environnement de production dans un autre groupe Auto Scaling. Les développeurs taggent le logiciel poussé vers un référentiel de code, comme GitHub, ou vers S3, avec une étiquette qui correspond à un environnement Blue ou Green. CodeDeploy détermine ensuite la cible et déploie le code dans l'environnement approprié.

Lorsque [le nouveau code est déployé dans un pipeline CI/CD](#), CodeDeploy peut immédiatement acheminer le trafic vers le nouvel environnement.

Dans ce guide

- CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications
 - Les tests en production : et si cela était bénéfique
 - DevOps : 10 catégories d'outils approuvés par les entreprises
 - Comment la technique Blue/Green garantit un déploiement cohérent et continu
-

Le processus fonctionne un peu différemment avec les fonctions Lambda qu'avec EC2. Avec Lambda, CodeDeploy crée des environnements parallèles Blue/Green, et les utilisateurs doivent sélectionner l'une des trois options pour déplacer le trafic de l'ancien vers le nouveau : Canary, linéaire ou d'un seul coup. Ces approches sont bien connues des spécialistes du déploiement et ne sont pas spécifiques au serverless :

- Canary déplace le trafic en deux paliers, avec une part et un intervalle configurables. Par exemple, 15 % de la circulation passe au Green pendant une journée pour test, après quoi les 85 % restants changent.
- Linéaire amène le trafic vers le nouveau déploiement par paliers égaux, avec un temps défini entre chaque déploiement. Par exemple, en cinq étapes, 20 % du trafic passe à l'environnement de production Green toutes les huit heures.
- Le déploiement massif et unique coupe tout le trafic du Blue au Green d'un seul coup.

Déploiement dans le cloud : quelles sont les autres alternatives

Mais il n'y pas qu'AWS : d'autres fournisseurs de cloud proposent également un modèle de déploiement Blue/Green. Par exemple, Microsoft propose Azure

Dans ce guide

- CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications
 - Les tests en production : et si cela était bénéfique
 - DevOps : 10 catégories d'outils approuvés par les entreprises
 - Comment la technique Blue/Green garantit un déploiement cohérent et continu
-

Service Fabric, avec, par défaut, des possibilités de déploiements en continu (rolling), de multiples versions d'applications. On peut également utiliser PowerShell pour automatiser le passage de l'environnement Blue vers le Green. Azure App Service permet également de réaliser cette transition.

De son côté, App Engine, le PaaS de Google offre également des fonctions qui permettent de gérer ce type de déploiement, comme la répartition du trafic qui dirige des connexions entrantes vers un environnement ou un autre. Google Kubernetes Engine (GKE), son implémentation de l'orchestrateur Kubernetes, met à disposition une option intéressante pour les déploiements Blue/Green. L'outil open source Spinnaker CI/CD peut déclencher la création d'un pipeline automatisé qui inclut un environnement de test Canary pour le nouveau code avant le déploiement en production. D'autres modèles de déploiement, avec Spinnaker ou Jenkins, peuvent être adaptés pour un déploiement Blue/Green.

■ Accéder à plus de contenu exclusif PRO+

Vous avez accès à cet e-Handbook en tant que membre via notre offre PRO+ : une collection de publications gratuites et offres spéciales rassemblées pour vous par nos partenaires et sur tout notre réseau de sites internet.

Dans ce guide

- CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications
 - Les tests en production : et si cela était bénéfique
 - DevOps : 10 catégories d'outils approuvés par les entreprises
 - Comment la technique Blue/Green garantit un déploiement cohérent et continu
-

L'offre PRO+ est gratuite et réservée aux membres du réseau de sites internet TechTarget.

Profitez de tous les avantages liés à votre abonnement sur: <http://www.lemagit.fr/eproducts>

Images; Fotolia

©2018 TechTarget. Tout ou partie de cette publication ne peut être transmise ou reproduite dans quelque forme ou de quelque manière que ce soit sans autorisation écrite de la part de l'éditeur.

Dans ce guide

- CI/CD, DevOps et cloud : les éléments clés de la modernisation des applications
 - Les tests en production : et si cela était bénéfique
 - DevOps : 10 catégories d'outils approuvés par les entreprises
 - Comment la technique Blue/Green garantit un déploiement cohérent et continu
-



Le document consulté provient du site www.lemagit.fr

Cyrille Chausson | *Rédacteur en Chef*
TechTarget
22 rue Léon Jouhaux, 75010 Paris
www.techtarget.com

©2018 TechTarget Inc. Aucun des contenus ne peut être transmis ou reproduit quelle que soit la forme sans l'autorisation écrite de l'éditeur. Les réimpressions de TechTarget sont disponibles à travers The YGS Group.

TechTarget édite des publications pour les professionnels de l'IT. Plus de 100 sites qui proposent un accès rapide à un stock important d'informations, de conseils, d'analyses concernant les technologies, les produits et les process déterminants dans vos fonctions. Nos événements réels et nos séminaires virtuels vous donnent accès à des commentaires et recommandations neutres par des experts sur les problèmes et défis que vous rencontrez quotidiennement. Notre communauté en ligne "IT Knowledge Exchange" (Echange de connaissances IT) vous permet de partager des questionnements et informations de tous les jours avec vos pairs et des experts du secteur.