

Présentation de DevOps sur AWS

David Chapman

Décembre 2014



Table des matières

Table des matières	2
Résumé	3
Introduction	3
Évolution agile vers DevOps	4
Infrastructure en tant que code	5
AWS CloudFormation	6
AMI AWS	9
Déploiement continu	10
AWS CodeDeploy	10
AWS CodePipeline	12
AWS CodeCommit	12
Documentation d'AWS Elastic Beanstalk et d'AWS OpsWorks	13
Déploiement « blue-green »	13
Automatisation	15
AWS Elastic Beanstalk	16
AWS OpsWorks	17
Supervision	18
Amazon CloudWatch	18
AWS CloudTrail	20
Sécurité	20
Identity Access Management (IAM)	20
Conclusion	21

Résumé

Avec l'accélération de l'innovation et l'évolution rapide des besoins client, les entreprises se doivent de devenir toujours plus agiles. Le délai de commercialisation est essentiel et pour faciliter les objectifs professionnels globaux, les services informatiques doivent être agiles. Au fil des années, le cycle de vie du développement des logiciels est passé d'un modèle en cascade à des modèles de développement agiles. Ces améliorations se déplacent en aval vers les opérations informatiques avec l'évolution de DevOps.

Pour répondre aux exigences d'une entreprise agile, les opérations informatiques doivent déployer des applications de manière cohérente, reproductible et fiable. Ceci peut uniquement être complètement réalisé avec l'adoption de l'automatisation.

Amazon Web Services (AWS) prend en charge nombre de principes et de pratiques DevOps dont les services informatiques peuvent tirer parti pour améliorer l'agilité des entreprises.

Ce document se concentre sur les principes et pratiques DevOps pris en charge sur la plateforme AWS. Une courte présentation des origines de DevOps plante le décor et explique comment et pourquoi DevOps a évolué.

Introduction

DevOps (ou opérations de développement) est un nouveau terme qui se concentre principalement sur une collaboration, une communication et une intégration améliorées entre les développeurs de logiciels et les opérations informatiques. Il s'agit d'un terme générique qui décrit une philosophie, un changement culturel et un changement d'orientation.

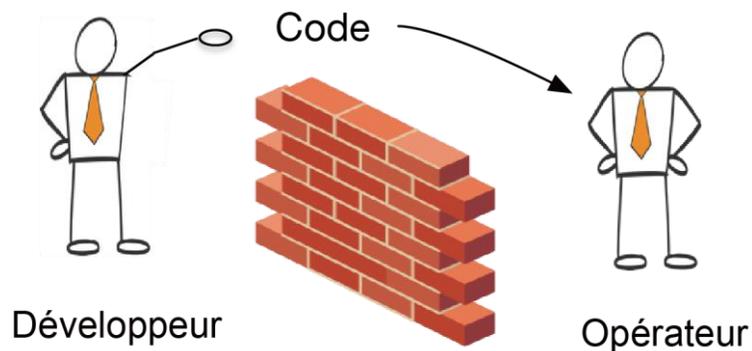


Figure 1 : Développeur lançant du code « par-dessus le mur »

Historiquement, de nombreuses organisations ont été structurées verticalement avec une intégration médiocre entre les équipes de développement, d'infrastructure, de sécurité et de support. Souvent, les groupes dépendent de structures organisationnelles différentes avec des philosophies et des objectifs d'entreprise différents.

Le déploiement des logiciels a été essentiellement le rôle du groupe des opérations informatiques. Fondamentalement, les développeurs aiment développer et changer les choses rapidement, alors que les opérations informatiques se concentrent sur la stabilité et la fiabilité. Ces divergences d'objectifs peuvent entraîner des conflits pouvant nuire à l'entreprise.

Aujourd'hui, ces anciennes divisions disparaissent avec une fusion des rôles informatiques et de développeur et en suivant une série de principes systématiques :

- Infrastructure en tant que code
- Déploiement continu
- Automatisation
- Supervision
- Sécurité

Un examen de chacun de ces principes révèle un lien étroit avec les offres disponibles auprès d'Amazon Web Services.

Évolution agile vers DevOps

Pour pleinement apprécier les principes DevOps, il est utile de comprendre le contexte dans lequel ceux-ci ont évolué. L'histoire commence avec le développement logiciel agile, qui s'est répandu il y a plus de dix ans et était considéré comme une meilleure approche pour développer des logiciels. Avant le modèle agile, la méthodologie dominante de développement en cascade était basée sur une séquence commençant par une phase des exigences lors de laquelle 100 % du système en développement était défini initialement. Cette approche s'est révélée inflexible et monolithique.

Le modèle agile a introduit le concept d'une collaboration nouvelle et améliorée entre les utilisateurs d'entreprise et les développeurs. Le développement logiciel a commencé à se concentrer sur des itérations de logiciels opérationnels devant évoluer au fil du temps et offrant une valeur ajoutée en cours de route. Le modèle agile est un processus d'ingénierie disciplinée et de nombreux outils le prennent maintenant en charge. Pour les développeurs, de tels outils incluent des IDE, des infrastructures de test d'unité et des optimiseurs de code. Au fur et à mesure que les développeurs deviennent plus productifs, l'entreprise gagne en productivité et peut répondre aux demandes des clients plus rapidement et efficacement.

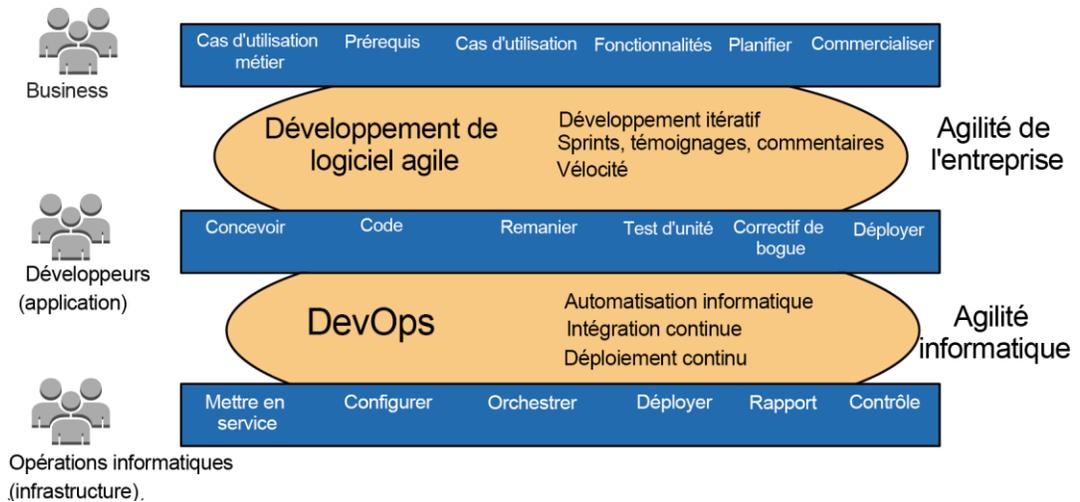


Figure 2 : La coévolution du développement logiciel agile et de DevOps

Au cours des quelques dernières années, l'évolution du développement logiciel agile a commencé à se déplacer en aval vers l'infrastructure sous l'intitulé *DevOps*. Alors que le développement logiciel agile se concentre principalement sur la collaboration entre l'entreprise et ses développeurs, le modèle DevOps se concentre sur la collaboration entre les développeurs, les opérations informatiques et les équipes chargées de la sécurité. Les opérations informatiques comprennent les administrateurs système, les administrateurs de base de données, les ingénieurs réseau, les architectes d'infrastructure et le personnel de support. Alors que le développement logiciel agile offre une agilité au niveau de l'entreprise, le modèle DevOps fournit une agilité informatique en autorisant le déploiement d'applications qui sont plus fiables, prévisibles et efficaces.

Les pratiques DevOps varient selon la tâche : Avec le développement d'applications, le modèle DevOps se concentre sur la création du code, la couverture du code, les tests d'unité, le packaging et le déploiement. Avec l'infrastructure, par contre, le modèle DevOps se concentre sur la mise en service, la configuration, l'orchestration et le déploiement. Dans chaque domaine, les principes sous-jacents de la gestion des versions, du déploiement, de la restauration, de la restauration par progression et des tests restent les mêmes.

Infrastructure en tant que code

Un principe fondamental du modèle DevOps consiste à traiter l'infrastructure de la même manière que les développeurs traitent le code. Le code d'application a un format et une syntaxe définis. Si le code n'est pas écrit selon les règles du langage de programmation, les applications ne peuvent pas être créées. Le code est stocké dans un système de gestion des versions qui consigne un historique de développement du code, des changements et des correctifs de bogues. Lorsque le code est compilé (intégré) dans des applications, nous nous attendons à ce qu'une application cohérente soit créée. En d'autres termes, la build est reproductible et fiable.

Gérer l'« infrastructure en tant que code » signifie appliquer la rigueur du développement de code d'application à la mise en service de l'infrastructure. Toutes les configurations doivent être définies de manière déclarative et stockées dans un système de gestion des versions, tout comme le code d'application. La mise en service, l'orchestration et le déploiement de l'infrastructure doivent prendre en charge l'utilisation du « code d'infrastructure ».

Jusqu'à récemment, la rigueur appliquée au développement de code d'application n'était pas forcément appliquée à l'infrastructure. Souvent, l'infrastructure est mise en service à l'aide de processus manuels. Les scripts développés lors de la mise en service peuvent ne pas être stockés dans des systèmes de gestion des versions, et la création d'environnements n'est pas toujours reproductible, fiable et cohérente.

En revanche, AWS fournit une méthode de création et de maintenance de l'infrastructure orientée DevOps. Tout comme les développeurs de logiciels écrivent du code d'application, AWS fournit des services qui permettent de créer, déployer et assurer la maintenance de l'infrastructure par programmation, et de manière descriptive et déclarative. Ces services offrent de la rigueur, de la clarté et de la fiabilité. Les services AWS présentés dans ce document sont au cœur de la stratégie DevOps et constituent les fondements de nombre de principes et pratiques AWS DevOps de plus haut niveau.

AWS CloudFormation

[AWS CloudFormation](#) constitue un bon exemple de la façon dont les principes DevOps sont utilisés dans la pratique.¹ En utilisant des templates AWS CloudFormation, vous pouvez définir et modéliser des ressources AWS pouvant être créées et mises à jour. Ces templates sont écrits dans un format appelé JavaScript Object Notation (JSON). Les templates nécessitent une syntaxe et une structure spécifiques qui dépendent des types de ressources créés et gérés. En utilisant ces templates, vous pouvez mettre en service une infrastructure de manière reproductible et fiable.

Vous pouvez créer des templates AWS CloudFormation personnalisés ou utiliser des exemples de templates disponibles publiquement. Une fois les templates déployés ou mis à jour dans l'environnement AWS, l'ensemble de ressources est appelé « stack » (ou pile). Vous pouvez gérer les stacks via la console AWS Management Console, l'interface de ligne de commande AWS ou des API AWS CloudFormation. Des actions courantes incluent create-stack, describe-stacks, list-stacks et update-stack.

Lorsque vous créez ou mettez à jour une stack dans la console, des événements sont affichés montrant le statut de la configuration. Si une erreur se produit, la stack est restaurée à son état précédent. Amazon Simple Notification Service (Amazon SNS) vous aide à gérer ces événements. Par exemple, vous pouvez utiliser Amazon SNS pour suivre la progression de création et de suppression de stack par e-mail, et intégrer ce processus à d'autres processus par programmation.

¹ <http://aws.amazon.com/cloudformation>

Avec les templates, vous pouvez utiliser un vaste ensemble d'offres AWS, notamment Amazon Simple Storage Service (Amazon S3), Auto Scaling, Amazon CloudFront, Amazon DynamoDB, Amazon Elastic Compute Cloud (EC2), Amazon ElastiCache, AWS Elastic Beanstalk, Elastic Load Balancing, AWS Identity and Access Management, AWS OpsWorks et Amazon Virtual Private Cloud.

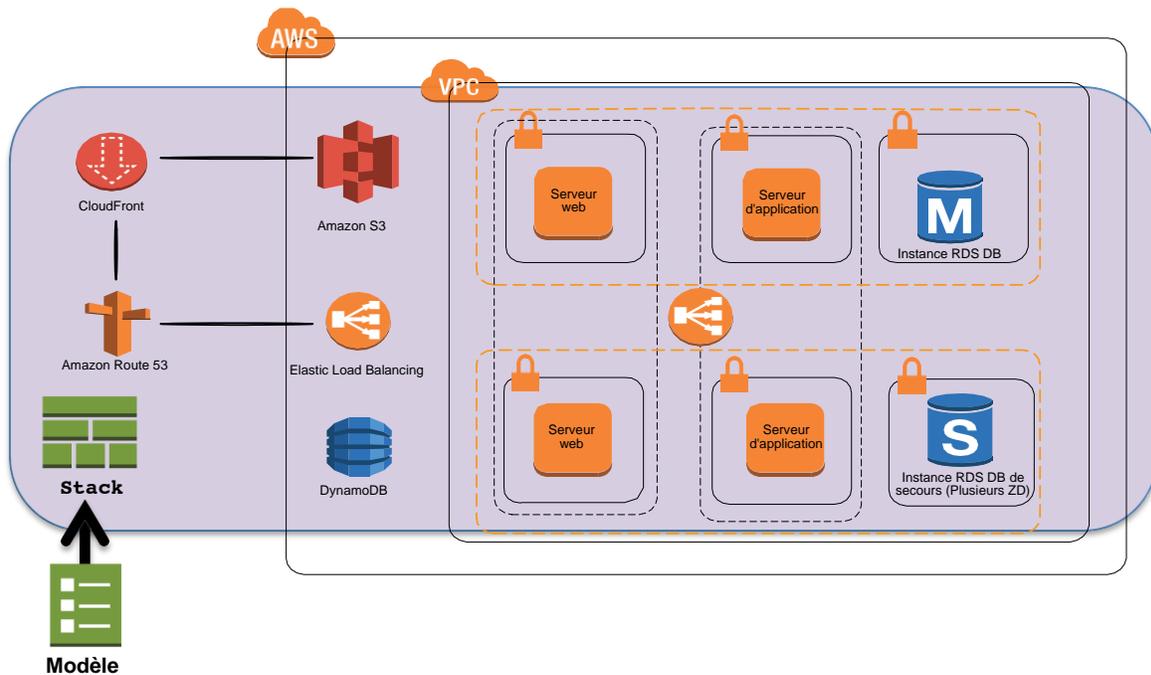


Figure 3 : Exemple AWS CloudFormation 1 — création d'un environnement complet (stack) à partir d'un template

Vous pouvez utiliser un template unique pour créer et mettre à jour un environnement complet, ou des templates distincts pour gérer des couches au sein d'un environnement. Cela permet de modulariser des templates et cela fournit également une couche de gouvernance qui est importante pour de nombreuses organisations.

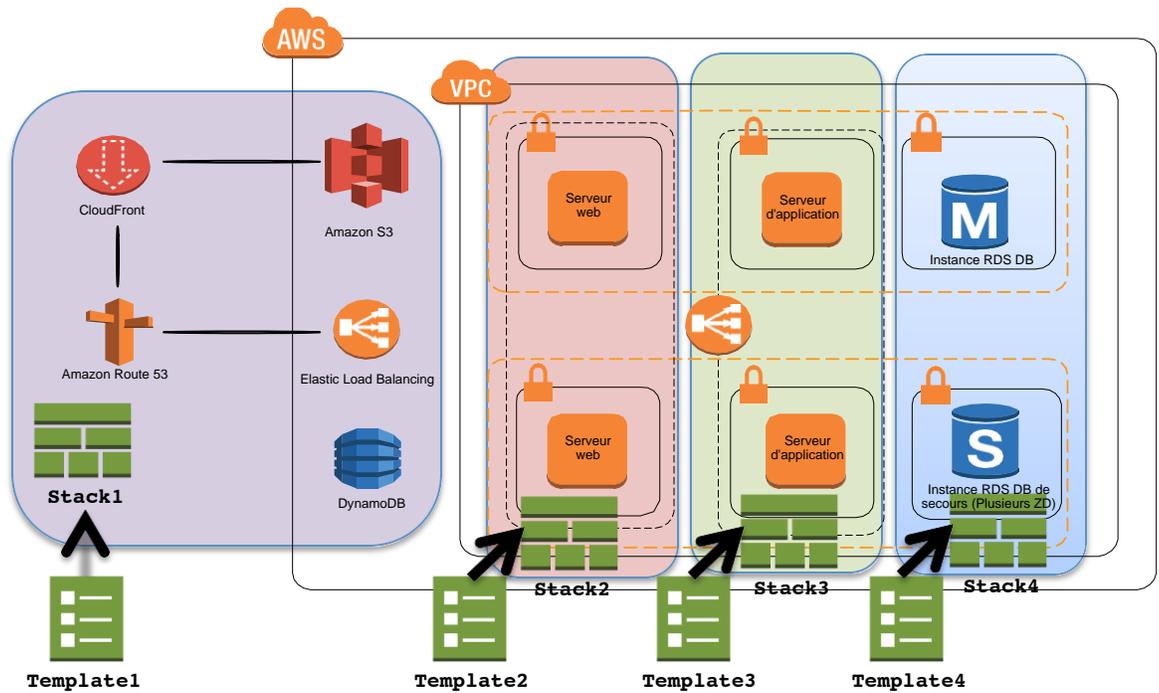


Figure 4 : Exemple AWS CloudFormation 2 — création d'un environnement complet (stacks) à partir de plusieurs templates dans une approche multicouche

AWS CloudFormation simplifie l'organisation et le déploiement d'un ensemble de ressources AWS et vous permet de décrire des dépendances ou de transmettre des paramètres spéciaux lorsque la stack est configurée.

Pour exploiter le potentiel d'AWS CloudFormation des « informations en tant que code », vous devez stocker les templates dans le contrôle de versions des systèmes de gestion du code source avant de les déployer ou mettre à jour dans AWS. Amazon S3 fournit un bon emplacement pour stocker et gérer les versions (versioning) des templates. Vous pouvez intégrer AWS CloudFormation aux outils de développement et de gestion de votre choix.

Aucun frais ne s'applique à la définition de votre « infrastructure en tant que code » dans le service AWS CloudFormation. Vous ne payez que les tarifs standards des ressources AWS créées par AWS CloudFormation et utilisées par votre application.

```
{
  "Description": "Create an EC2 instance running the Amazon Linux 32 bit AMI.",
  "Parameters": {
    "KeyPair": {
      "Description": "The EC2 Key Pair to allow SSH access to the instance",
      "Type": "String"
    }
  },
  "Resources": {
    "Ec2Instance": {
      "Type": "AWS::EC2::Instance",
      "Properties": {
        "KeyName": { "Ref": "KeyPair" },
        "ImageId": "ami-75g0061f",
        "InstanceType": "m1.medium"
      }
    }
  },
  "Outputs": {
    "InstancedId": {
      "Description": "The InstancedId of the newly created EC2 instance",
      "Value": { "Ref": "Ec2Instance" }
    }
  }
}
```

Figure 5 : Exemple de template AWS CloudFormation pour le lancement d'une instance EC2

Dans l'exemple ci-dessus, un template AWS CloudFormation a été défini en notation JSON pour créer une instance Amazon EC2. Cet exemple met en service un type m1.medium d'instance EC2. Le mot `Ref` fait référence aux paramètres accédés lorsque la stack est créée. Un paramètre appelé `KeyPair` peut être fourni lorsque la stack est créée. Il s'agit du nom de la paire de clés utilisée lors de l'accès à l'instance avec SSH.

AMI AWS

Une Amazon Machine Image (AMI) constitue un autre exemple d'« infrastructure en tant que code ». Ce composant de base de l'informatiques AWS est une sorte de template numérique qui peut lancer (mettre en service) des instances Amazon EC2 (l'environnement de calcul AWS fondamental dans le cloud). L'image contient une configuration logicielle telle qu'un serveur web, un serveur d'application et une base de données.

Vous avez le choix entre trois types d'AMI :

- Publiées par AWS
- Tierces
- Créées et personnalisées

AWS publie des AMI contenant des configurations logicielles courantes basées sur des systèmes d'exploitation populaires comme Linux et Microsoft Windows. Des AMI peuvent également être obtenues auprès de fournisseurs tiers, certaines d'entre elles étant disponibles sur [AWS Marketplace](#).² Vous ou votre organisation peut également créer et publier ses propres AMI personnalisées. Les AMI personnalisées d'organisation incluent généralement des logiciels distribués à l'échelle de l'entreprise comme des systèmes d'exploitation renforcés, des anti-virus et des suites de productivité bureautiques.

Les AMI peuvent également inclure des logiciels d'application regroupés avec l'AMI, ou des scripts et des logiciels permettant à l'instance d'installer des logiciels d'application au lancement (« démarrage »). Le préchargement de logiciels de niveau application dans une AMI présente des avantages et des inconvénients. Pour ce qui est des avantages, les instances peuvent être lancées très rapidement car elles n'ont pas besoin d'installer des logiciels supplémentaires au démarrage. Par contre, vous pouvez avoir besoin de créer une nouvelle AMI chaque fois que vos logiciels de niveau application changent.

Déploiement continu

Le déploiement continu est un autre concept de base d'une stratégie DevOps. Son objectif principal est de permettre le déploiement automatisé d'un code d'application prêt à la production.

Parfois, le déploiement continu est appelé diffusion en continu. La seule différence est que le déploiement continu fait généralement référence à des déploiements de production.

À l'aide des pratiques et outils de diffusion en continu, les logiciels peuvent être déployés rapidement, et de façon répétitive et fiable. Si un déploiement échoue, la version précédente peut être restaurée automatiquement.

AWS CodeDeploy

Un excellent exemple de ce principe dans AWS est le service de déploiement de code [AWS CodeDeploy](#).³ Ses fonctions de base donnent la possibilité de déployer des applications sur une flotte Amazon E2C avec une interruption minimale, en centralisant le contrôle et en s'intégrant à votre processus de lancement ou de diffusion en continu de logiciels existants.

² <https://aws.amazon.com/marketplace>

³ <http://aws.amazon.com/codedeploy>

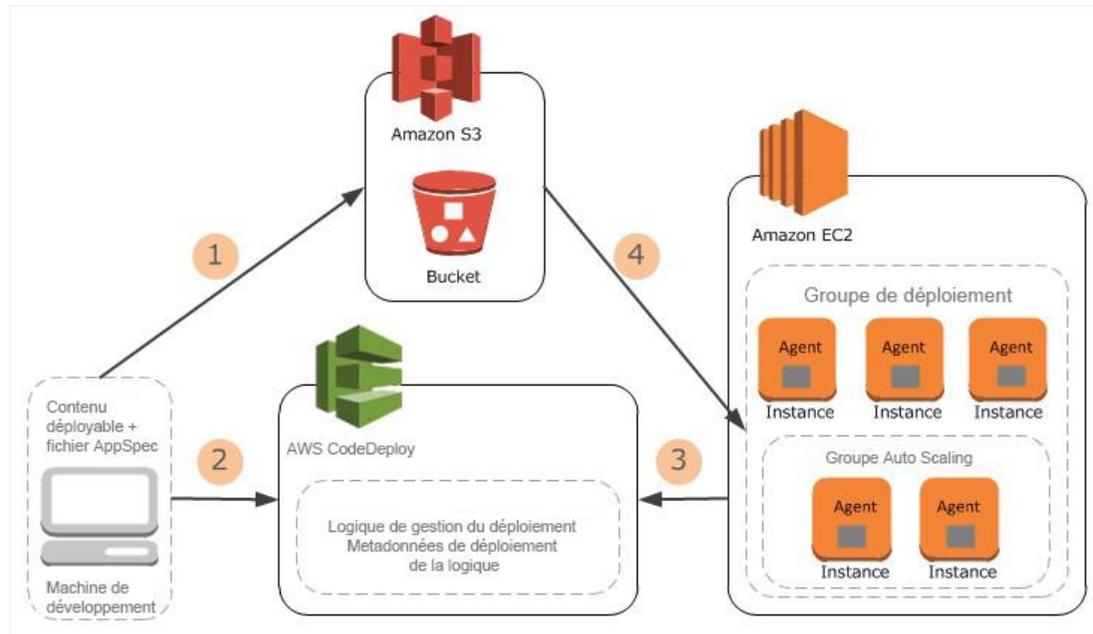


Figure 6 : Processus AWS CodeDeploy

Voici comment cela fonctionne :

1. Le contenu de l'application est packagé et déposé dans Amazon S3 avec un fichier AppSpec (Application Specific) qui définit une série d'étapes de déploiement devant être exécutées par AWS CodeDeploy. Ce package est appelé « révision » CodeDeploy.
2. Vous créez une application dans AWS CodeDeploy et définissez les instances sur lesquelles l'application doit être déployée (DeploymentGroup). L'application définit également le compartiment Amazon S3 dans lequel le package de déploiement réside.
3. Un agent AWS CodeDeploy est déployé dans chaque instance Amazon EC2 participante. L'agent interroge AWS CodeDeploy pour déterminer quoi extraire et quand extraire une révision à partir du compartiment Amazon S3.
4. L'agent AWS CodeDeploy extrait le code d'application packagé et le déploie sur l'instance. Le fichier AppSpec contenant les instructions de déploiement est également téléchargé.

Ainsi, AWS CodeDeploy illustre bien le déploiement automatisé continu qui est au cœur du concept DevOps.

AWS CodePipeline

Comme AWS CodeDeploy, [AWS CodePipeline](http://aws.amazon.com/codepipeline) (disponible en 2015) est un service d'automatisation de diffusion et de lancement en continu qui favorise la fluidité des déploiements.⁴ Vous pouvez concevoir votre flux de travail de développement pour la vérification du code, la création de l'application, son déploiement pour activation, les tests et sa mise en production. Vous pouvez intégrer des outils tiers à n'importe quelle étape de votre processus de lancement, ou utiliser AWS CodePipeline en tant que solution de bout en bout. Avec AWS CodePipeline, vous pouvez offrir rapidement des fonctions et des mises à jour de qualité grâce à l'automatisation de votre processus de build, de test et de lancement.

AWS CodePipeline présente plusieurs avantages conformes au principe DevOps de déploiement continu :

- Diffusion rapide
- Qualité améliorée
- Workflow configurable
- Facile à intégrer

AWS CodeCommit

Également prévu pour 2015, [AWS CodeCommit](http://aws.amazon.com/codecommit) est un service de contrôle de code source géré, sécurisé et extrêmement évolutif qui héberge des référentiels Git privés.⁵ Grâce à CodeCommit, vous n'avez plus besoin de gérer votre propre système de contrôle de code source, ni à vous inquiéter quant à l'adaptation de son infrastructure. Vous pouvez utiliser CodeCommit pour tout stocker, des éléments de code aux fichiers binaires. CodeCommit prend en charge la fonctionnalité standard de Git, et s'exécute ainsi sans problème avec vos outils Git existants. Votre équipe peut également utiliser les outils de code en ligne de CodeCommit pour collaborer sur des projets, les parcourir et les modifier.

AWS CodeCommit offre plusieurs avantages :

- Entièrement géré
- Capable de tout stocker
- Haut niveau de disponibilité
- Offre des cycles de développement plus rapides
- Fonctionne avec vos outils existants
- Sécurité

⁴ <http://aws.amazon.com/codepipeline>

⁵ <http://aws.amazon.com/codecommit>

Documentation d'AWS Elastic Beanstalk et d'AWS OpsWorks

[AWS Elastic Beanstalk](#)⁶ et [AWS OpsWorks](#)⁷ prennent charge le déploiement continu des changements du code d'application et des modifications d'infrastructure. Dans AWS Elastic Beanstalk, les déploiements de changements de code sont stockés en tant que « versions de l'application », et les modifications d'infrastructure sont déployées en tant que « configurations enregistrées ». AWS OpsWorks dispose de son propre processus pour le déploiement d'applications et peut définir des commandes de lancement et des recettes de Chef supplémentaires lors de l'exécution.

Un exemple de version d'application serait une nouvelle application Java que vous chargez sous la forme d'un fichier `.zip` ou `.war`. Un exemple de configuration enregistrée serait une configuration AWS Elastic Beanstalk qui utilise Elastic Load Balancing et Auto Scaling plutôt qu'une instance unique. Lorsque vous avez fini d'effectuer des modifications, vous pouvez enregistrer votre nouvelle configuration.

AWS Elastic Beanstalk prend en charge la pratique DevOps appelée « propagation des déploiements ». Lorsque celle-ci est activée, vos déploiements de configuration fonctionnent conjointement avec Auto Scaling pour veiller à ce qu'un nombre d'instances défini soit toujours disponible lorsque les changements de configuration sont effectués. Vous prenez ainsi le contrôle alors que les instances Amazon EC2 sont mises à jour. Par exemple, si le type d'instance EC2 est modifié, vous pouvez déterminer si AWS Elastic Beanstalk met à jour simultanément toutes les instances ou laisse certaines instances s'exécuter pour répondre aux demandes pendant que les autres instances sont mises à jour.

De même, AWS OpsWorks vous offre la possibilité de définir quelles instances dans quelles couches doivent être mises à jour lorsque des déploiements sont exécutés.

Des fonctions supplémentaires d'AWS Elastic Beanstalk et AWS OpsWorks sont décrites dans la section [Automatisation](#).

Déploiement « blue-green »

Le déploiement « blue-green » est une pratique de déploiement DevOps qui utilise des services de noms de domaine (DNS) pour effectuer des déploiements d'application. La stratégie implique de démarrer avec un environnement existant (blue, bleu) tout en testant un nouvel environnement (green, vert). Quand le nouvel environnement a réussi tous les tests nécessaires et est prêt à être activé, il suffit de rediriger le trafic de l'ancien environnement vers le nouveau via DNS.

⁶ <http://aws.amazon.com/elasticbeanstalk>

⁷ <http://aws.amazon.com/opsworks>

AWS offre tous les outils dont vous avez besoin pour implémenter une stratégie de déploiement « blue-green ». Vous configurez votre nouvel environnement d'infrastructure idéal à l'aide d'un service comme AWS CloudFormation ou AWS Elastic Beanstalk. Avec des templates AWS CloudFormation, vous pouvez facilement créer un nouvel environnement identique à l'environnement de production existant.

Si vous utilisez le service DNS AWS Amazon Route 53, vous pouvez diriger le trafic à l'aide d'ensembles d'enregistrements de ressource pondérés. En utilisant ces ensembles d'enregistrements, vous pouvez définir plusieurs services ou équilibres de charge avec la résolution DNS.

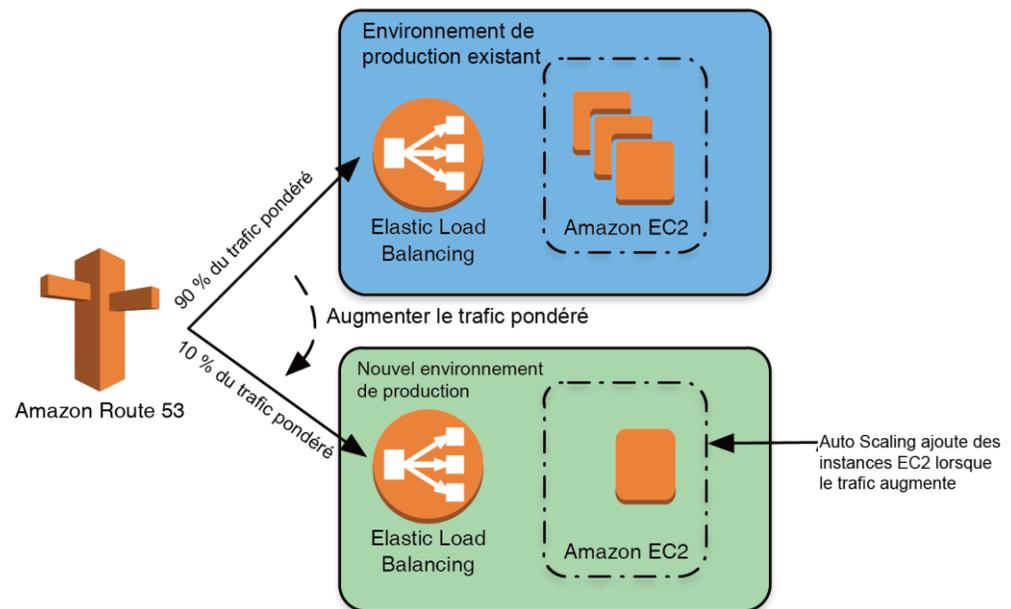


Figure 7 : Déploiement « blue-green » à l'aide d'ensembles d'enregistrements de ressource pondérés

La résolution du service DNS (conversion d'un nom de domaine en adresse IP) est pondérée, ce qui signifie que vous pouvez définir la quantité de trafic dirigée vers votre environnement de production nouvellement déployé. À l'aide de cette fonction, vous pouvez tester l'environnement et lorsque vous êtes sûr que celui-ci est correct, augmenter la pondération. Lorsque que l'ancien environnement de production reçoit 0 % du trafic, vous pouvez le conserver à des fins de sauvegarde ou le mettre hors service. Au fur et à mesure que la quantité de trafic dans le nouvel environnement augmente, vous pouvez utiliser Auto Scaling pour procéder à une mise à l'échelle ascendante avec des instances Amazon EC2 supplémentaires.

C'est cette capacité à créer et disposer facilement d'environnements identiques dans le cloud AWS qui rend possible des pratiques DevOps telles que le déploiement « blue-green ».

Vous pouvez également utiliser le déploiement « blue-green » pour des services principaux comme le déploiement de base de données ou le basculement.

Automatisation

L'automatisation représente une autre philosophie et une autre pratique de base de DevOps. L'automatisation se concentre sur la configuration, le déploiement et le support de l'infrastructure et des applications qui s'exécutent sur cette dernière. À l'aide de l'automatisation, vous pouvez configurer plus rapidement des environnements de façon standardisée et reproductible. L'élimination des processus manuels est essentielle pour une stratégie DevOps réussie. Traditionnellement, la configuration de serveur et le déploiement d'application était avant tout un processus manuel. Les environnements deviennent non standards et la reproduction d'un environnement en cas de problème s'avère difficile.

L'utilisation de l'automatisation est essentielle pour profiter pleinement des avantages du cloud. En interne, AWS s'appuie fortement sur l'automatisation pour offrir les fonctions de base d'élasticité et d'évolutivité. Les processus manuels sont source d'erreurs, peu fiables et inadaptés pour la prise en charge d'une entreprise agile. Souvent, une organisation doit immobiliser des ressources hautement qualifiées pour effectuer une configuration manuelle. Il serait préférable de consacrer ce temps précieux à des activités plus stratégiques et à plus forte valeur ajoutée dans l'entreprise.

Les environnements d'exploitation modernes s'appuient généralement sur une automatisation complète pour éliminer les interventions ou les accès manuels aux environnements de production. Cela inclut toutes les tâches de lancement de logiciels, de configuration de machines, d'application de correctifs de système d'exploitation, de dépannage ou de correction de bogues. De nombreux niveaux de pratiques d'automatisation peuvent être utilisés ensemble pour offrir un processus automatisé de bout en bout de haut niveau.

L'automatisation offre de nombreux avantages :

- Changements rapides
- Productivité améliorée
- Configurations reproductibles
- Environnements reproductibles
- Élasticité optimisée
- Dimensionnement automatique optimisé
- Tests automatisés

L'automatisation est une pièce maîtresse des services AWS et est prise en charge en interne dans tous les services, fonctions et offres.

AWS Elastic Beanstalk

AWS Elastic Beanstalk est un parfait exemple d'automatisation dans AWS. AWS Elastic Beanstalk est un service simple et productif qui permet aux développeurs de déployer des applications dans des stacks de technologies couramment utilisées. Il s'agit d'une interface simple d'utilisation qui aide les développeurs à déployer rapidement et facilement des applications multinationales. AWS Elastic Beanstalk prend en charge l'automatisation et de nombreuses autres bonnes pratiques DevOps, notamment le déploiement d'application automatisé, la supervision, la configuration d'infrastructure et la gestion des versions. Les changements d'application et d'infrastructure peuvent être facilement restaurés et restaurés par progression.

La création d'environnements est un bon exemple d'automatisation AWS Elastic Beanstalk. Il vous suffit de spécifier les détails pour votre environnement. AWS Beanstalk exécute alors toutes les tâches de configuration et de mise en service de façon autonome. Par exemple, voici juste quelques-unes des options que vous pouvez spécifier dans l'assistant d'application :

- Si vous souhaitez un niveau de serveur web (contenant un serveur web et un serveur d'application) ou un niveau d'exécutant (utilisant Amazon Simple Queue Service).
- Quelle plateforme utiliser pour votre application. Les options incluent IIS, Node.js, PHP, Python, Ruby, Tomcat ou Docker.
- S'il faut lancer une instance unique ou créer un environnement à dimensionnement automatique et équilibrage de charge.
- Quelle URL affecter automatiquement à votre environnement.
- Si l'environnement inclut une instance Amazon Relational Database.
- S'il faut créer votre environnement dans un Amazon Virtual Private Cloud.
- Quelle URL (le cas échéant) utiliser pour les vérifications de l'état de votre application.
- Quelles balises (le cas échéant) appliquer pour identifier votre environnement.

AWS Elastic Beanstalk utilise également l'automatisation pour déployer des applications. Selon la plateforme, tout ce que vous avez à faire est de déployer des applications est de charger des packages sous la forme de fichiers `.war` ou `.zip` directement depuis votre ordinateur ou d'Amazon S3.

Pendant la création de l'environnement, AWS Elastic Beanstalk consigne automatiquement des événements sur la console de gestion en fournissant des commentaires sur la progression et le statut du lancement. Une fois la création terminée, vous pouvez accéder à votre application à l'aide de l'URL définie.

AWS Elastic Beanstalk peut être personnalisé si vous souhaitez prendre le contrôle de certains aspects de l'application et de la stack de technologies.

AWS OpsWorks

AWS OpsWorks pousse les principes de DevOps encore plus loin qu'AWS Elastic Beanstalk. AWS OpsWorks offre encore plus de niveaux d'automatisation avec des fonctions supplémentaires comme l'intégration d'un logiciel de gestion de la configuration (Chef) et la gestion du cycle de vie des applications. Vous pouvez utiliser la gestion du cycle de vie des applications pour définir à quel moment des ressources sont configurées, déployées ou résiliées, ou quand leur déploiement est annulé.

Pour une flexibilité accrue, AWS OpsWorks vous fait définir votre application en stacks configurables. Vous pouvez également sélectionner des stacks d'applications prédéfinies. Les stacks d'applications contiennent toute la mise en service pour les ressources AWS requises par votre application, y compris les serveurs web, les serveurs d'application, les bases de données et les équilibreurs de charge.

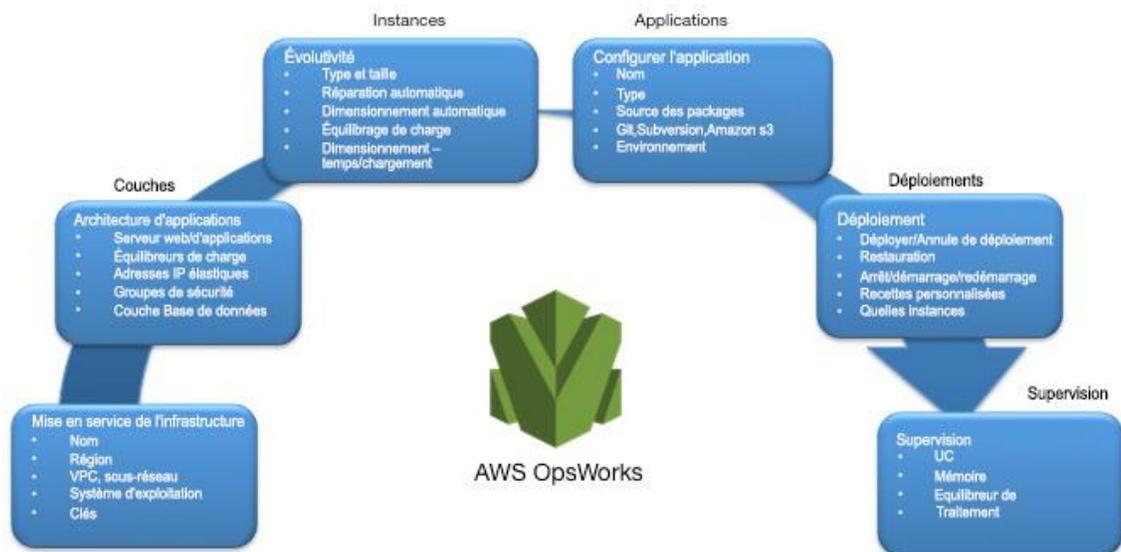


Figure 8 : AWS OpsWorks illustrant les fonctions et l'architecture DevOps

Les stacks d'applications sont organisées en couches d'architecture pour pouvoir être gérées de façon indépendante. Par exemple, elles peuvent inclure une couche web, une couche applicative et une couche de base de données. Prêt à l'emploi, AWS OpsWorks simplifie également la configuration de groupes Auto Scaling et d'équilibreurs de charge Elastic Load Balancing, en illustrant plus en détails le principe d'automatisation de DevOps. Tout comme AWS Elastic Beanstalk, AWS OpsWorks prend en charge la gestion des versions d'application (versioning), le déploiement continu et la gestion de la configuration de l'infrastructure.

AWS OpsWorks prend également en charge les pratiques DevOps de supervision et de consignation (présentées dans la section suivante). La prise en charge de la supervision est assurée par Amazon CloudWatch. Tous les événements du cycle de vie sont consignés et un journal Chef distinct documente les recettes Chef exécutées, avec les éventuelles exceptions.⁸

Supervision

La communication et la collaboration sont fondamentales dans une stratégie DevOps. Pour les faciliter, les commentaires sont essentiels. Dans AWS, les commentaires sont fournis par deux services de base : Amazon CloudWatch et AWS CloudTrail. Ensemble, ils offrent une infrastructure de supervision, d'alerte et d'audit solides. Les développeurs et les équipes opérationnelles peuvent donc collaborer étroitement et en toute transparence.

Amazon CloudWatch

[Amazon CloudWatch](#) supervise en temps réel toutes les ressources AWS et les applications que vous exécutez sur celles-ci.⁹ Les ressources et les applications peuvent produire des métriques regroupées et suivies par Amazon CloudWatch. Vous pouvez configurer des alarmes et envoyer des notifications quand des événements ont lieu. Vous pouvez configurer des notifications dans plusieurs formats, notamment via des e-mails, Amazon SNS et Amazon Simple Queue Service. Les notifications peuvent être fournies à des personnes, des équipes ou d'autres ressources AWS.

En plus de fournir des commentaires, Amazon CloudWatch prend également en charge le concept d'automatisation DevOps. Des services AWS comme Auto Scaling s'appuient sur CloudWatch pour activer des notifications qui déclenchent une action automatisée comme un dimensionnement vers le haut ou vers le bas des instances Amazon EC2, et des augmentations et diminutions de charge.

⁸ Une recette Chef est une application Ruby qui définit toutes les tâches qui doivent être exécutées pour configurer un système, comme l'installation de packages et de correctifs.

⁹ <http://aws.amazon.com/cloudwatch>

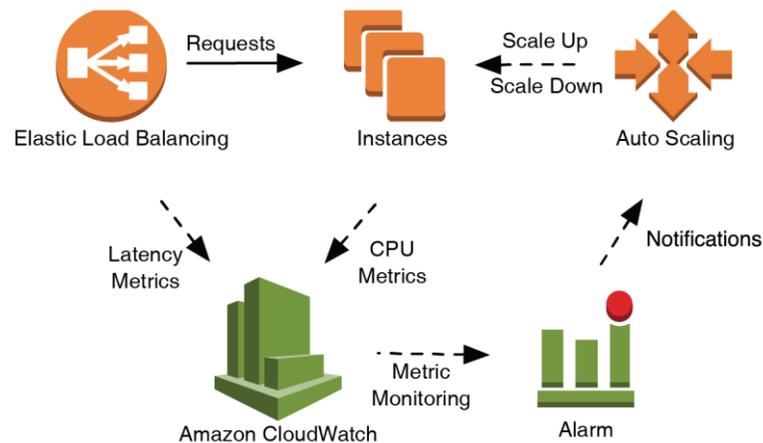


Figure 9 : Exemple d'automatisation DevOps à l'aide d'Amazon CloudWatch et d'Auto Scaling

Dans l'exemple ci-dessus, Amazon CloudWatch supervise des métriques de latence issues d'Elastic Load Balancing et des métriques d'utilisation d'UC moyenne provenant des instances Amazon EC2 en cours d'exécution. Les métriques de latence mesurent le temps nécessaire pour que des réponses soient renvoyées après que des demandes ont été faites sur des instances Amazon EC2. Vous pouvez créer des stratégies de dimensionnement pour agir sur des alarmes qui sont déclenchées quand des seuils définis sont dépassés. Une telle stratégie peut se traduire par une augmentation ou une diminution du nombre d'instances Amazon EC2, en fonction de la situation. Vous pouvez définir des notifications supplémentaires pour fournir des messages via Amazon SNS. Cela peut s'avérer utile pour avertir les parties concernées, comme des équipes de support, que des événements ont eu lieu.

L'exemple d'Auto Scaling illustre comment des services AWS fonctionnent ensemble pour fournir les services transparents et automatisés qui sont essentiels pour adopter une stratégie DevOps. Les administrateurs système et les équipes de support peuvent se concentrer sur d'autres besoins métiers à valeur ajoutée et être assurés que l'infrastructure AWS gère les exigences de dimensionnement des applications. Notez que ce scénario suppose que l'application concernée est optimisée pour le cloud et conçue d'une façon évolutive horizontalement pour tirer profit des avantages offerts par Auto Scaling.

AWS CloudTrail

Pour adopter les principes DevOps de collaboration, communication et transparence, il est important de comprendre qui apporte des modifications à votre infrastructure. Dans AWS, cette transparence est fournie par le service [AWS CloudTrail](#).¹⁰ Toutes les interactions AWS sont traitées via des appels d'API AWS qui sont supervisées et consignées par AWS CloudTrail. Tous les fichiers journaux générés sont stockés dans un compartiment Amazon S3 que vous définissez. Les fichiers journaux sont chiffrés à l'aide du chiffrement côté serveur (SSE) d'Amazon S3. Tous les appels d'API sont consignés, qu'ils proviennent directement d'un utilisateur ou qu'ils soient exécutés pour le compte d'un utilisateur par un service AWS. Plusieurs groupes peuvent tirer parti des journaux CloudTrail, notamment les équipes opérationnelles en soutien des équipes de sécurité pour la gouvernance, et les équipes financières pour la facturation.

Sécurité

Dans un environnement mettant en œuvre DevOps, l'accent sur la sécurité est d'une importance primordiale. L'infrastructure et les ressources de l'entreprise doivent être protégées et lorsque des problèmes surviennent, ceux-ci doivent être résolus rapidement et de manière efficace.

Identity and Access Management (IAM)

Le service AWS Identity and Access Management (IAM) est un composant de l'infrastructure de sécurité AWS. Avec IAM, vous pouvez gérer de façon centralisée les utilisateurs et les informations d'identification de sécurité comme les mots de passe, les clés d'accès et les stratégies d'autorisation qui contrôlent les services et les ressources AWS auxquels les utilisateurs peuvent accéder. Vous pouvez également faire appel à IAM pour créer des rôles qui sont largement utilisés au sein d'une stratégie DevOps. Avec un rôle IAM, vous pouvez définir un ensemble d'autorisations d'accès aux ressources dont a besoin un utilisateur ou un service. Mais au lieu d'attacher les autorisations à un utilisateur ou un groupe spécifique, vous les attachez à un rôle nommé. Des ressources peuvent être associées à des rôles et des services pouvant alors être définis par programme pour assumer un rôle.

Des exigences et des contrôles de sécurité doivent être respectés lors des processus d'automatisation, et un soin tout particulier doit être apporté aux mots de passe et aux clés. De bonnes pratiques en matière de sécurité doivent être suivies en permanence. Pour plus de détails sur l'importance de la sécurité pour AWS, visitez le [Centre de sécurité AWS](#).¹¹

¹⁰ <http://aws.amazon.com/cloudtrail>

¹¹ <http://aws.amazon.com/security/>

Conclusion

Pour rendre la transition vers la cloud fluide et efficace, les entreprises technologiques peuvent adopter les principes et pratiques DevOps. Ces principes sont intégrés à la plateforme AWS. En effet, ils constituent la pièce maîtresse de nombreux services AWS, surtout ceux faisant partie des offres de déploiement et de supervision.

Commencez par définir votre infrastructure en tant que code à l'aide du service AWS CloudFormation. Ensuite, définissez la manière dont vos applications vont utiliser le déploiement continu à l'aide de services comme AWS CodeDeploy, AWS CodePipeline et AWS CodeCommit. Au niveau des applications, utilisez des services comme AWS Elastic Beanstalk et AWS OpsWorks pour simplifier la configuration des architectures les plus courantes. L'utilisation de ces services facilite également l'inclusion d'autres services importants comme Auto Scaling et Elastic Load Balancing. Enfin, utilisez la stratégie de supervision DevOps (AWS CloudWatch) et de solides pratiques de sécurité (AWS IAM).

Avec AWS comme partenaire, vos principes DevOps conféreront de l'agilité à votre entreprise et votre organisation informatique, et accéléreront votre transition vers le cloud.

© 2014, Amazon Web Services, Inc. ou ses affiliés. Tous droits réservés.