Livre blanc AWS

Présentation de DevOps sur AWS

Présentation de DevOps sur AWS: Livre blanc AWS

Table of Contents

Présentation de DevOps sur AWS

Date de publication : 16 octobre 2020 (Révisions du document)

Résumé

Aujourd'hui plus que jamais, les entreprises entreprennent leur transformation numérique pour établir des liens plus étroits avec leurs clients afin d'obtenir une valeur opérationnelle durable et résistante. Les organisations, quelles que soient leur forme et leur taille, laissent leurs concurrents loin derrière et pénètrent de nouveaux marchés en innovant plus rapidement. Pour ces organisations, il est important de se concentrer sur l'innovation et les perturbations liées aux logiciels, ce qui rend essentielle la rationalisation de la fourniture de leurs logiciels. Les organisations accélèrent les délais entre l'idée et la production en donnant priorité à la rapidité et à l'agilité pourraient devenir les acteurs du changement.

Plusieurs facteurs doivent être pris en compte pour espérer devenir le prochain acteur numérique. Ce livre blanc se concentre sur le DevOps, ainsi que sur les services et fonctions de la plateforme AWS qui aideront une organisation à fournir rapidement des applications et des services.

Introduction

Le DevOps est une combinaison de pratiques et de modèles culturels et d'ingénierie, et d'outils qui améliorent la capacité d'une organisation à livrer des applications et des services à un rythme plus soutenu et de meilleure qualité. Au fil du temps, plusieurs pratiques essentielles sont apparues lors de l'adoption de DevOps : intégration continue, livraison continue, Infrastructure as Code, surveillance et journalisation.

Ce livre blanc met en lumière les fonctionnalités AWS qui vous aident à accélérer l'adoption de DevOps. Il explique également comment les services AWS peuvent vous aider à éliminer les tâches complexes non différenciées associées à l'adaptation à DevOps. Nous expliquons également comment créer une fonctionnalité d'intégration et de livraison continues sans gérer de serveurs ou de nœuds de génération, et comment tirer parti d'Infrastructure as Code pour allouer et gérer vos ressources cloud de manière cohérente et reproductible.

- L'intégration continue est une méthode de développement de logiciels dans laquelle les développeurs intègrent régulièrement leurs modifications de code à un référentiel centralisé, suite à quoi des opérations de création et de test sont automatiquement menées.
- La livraison continue est une méthode de développement de logiciels avec laquelle les changements de code sont automatiquement générés, testés et préparés pour une publication dans un environnement de production.
- Infrastructure as Code est une pratique qui implique la mise en service et la gestion de l'infrastructure à l'aide de code et de techniques de développement de logiciels, notamment le contrôle des versions et l'intégration continue.
- La surveillance et la journalisation permet aux organisations de découvrir l'impact des performances de l'application et de l'infrastructure sur l'expérience de l'utilisateur final du produit.
- Les pratiques de communication et collaboration sont établies pour rapprocher les équipes. Elles permettent de créer des flux de travail et de répartir les responsabilités pour DevOps.
- La sécurité doit être une préoccupation transversale. Vos pipelines d'intégration continue et de livraison continue (CI/CD), ainsi que les services associés, doivent être protégés et des autorisations de contrôle d'accès appropriées doivent être configurées.

L'examen de chacun de ces principes montre un lien étroit avec les offres disponibles auprès d'Amazon Web Services (AWS).

Intégration continue

Intégration continue (CI) : l'intégration continue est une pratique de développement logiciel où les développeurs fusionnent régulièrement leurs modifications de code dans un référentiel de code central, après quoi des créations et des tests automatisés sont exécutés. L'intégration continue aide à trouver et à corriger plus rapidement les bogues, à améliorer la qualité des logiciels et à réduire le temps nécessaire pour valider et publier de nouvelles mises à jour de logiciels.

AWS propose les services suivants pour l'intégration continue :

Rubriques

- AWS CodeCommit
- AWS CodeBuild
- AWS CodeArtifact

AWS CodeCommit

AWS CodeCommit est un service de contrôle de code source géré, sécurisé, extrêmement évolutif qui héberge les référentiels Git privés. CodeCommit vous évite de devoir utiliser votre propre système de contrôle de source. Vous n'avez plus besoin de mettre en service et de dimensionner du matériel, ni d'installer, de configurer ou d'exploiter des logiciels. Vous pouvez utiliser CodeCommit pour stocker tout ce que vous voulez, du code aux fichiers binaires. Il prend en charge la fonctionnalité standard de Git, ce qui lui permet de fonctionner sans problème avec vos outils basés sur Git. 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 :

Collaboration : AWS CodeCommit est conçu pour le développement collaboratif de logiciels. Vous pouvez facilement valider, diviser et fusionner vos codes, ce qui vous permet de garder facilement le contrôle sur les projets de votre équipe. CodeCommit prend également en charge les demandes d'extraction, ce qui permet à un mécanisme de demander des vérifications de code de discuter du code avec les collaborateurs.

Chiffrement : vous pouvez transférer vos fichiers vers et depuis AWS CodeCommit en utilisant HTTPS ou SSH, selon vos préférences. Vos référentiels sont également chiffrés de manière automatique au repos via AWS Key Management Service (AWS KMS) à l'aide des clés propres à chaque client.

Contrôle d'accès : AWS CodeCommit utilise AWS Identity and Access Management (IAM) pour contrôler et surveiller qui peut accéder à vos données, ainsi que comment, quand et où ils peuvent y accéder. CodeCommit vous aide également à surveiller vos référentiels via AWS CloudTrail et Amazon CloudWatch.

Haute disponibilité et durabilité : AWS CodeCommit stocke vos référentiels dans Amazon Simple Storage Service (Amazon S3) et Amazon DynamoDB. Vos données chiffrées sont stockées de manière redondante dans plusieurs installations. Cette architecture améliore la disponibilité et la durabilité des données de votre référentiel.

Notifications et scripts personnalisés : vous pouvez désormais recevoir des notifications pour les événements ayant un impact sur vos référentiels. Les notifications seront envoyées sous forme de notifications Amazon Simple Notification Service (Amazon SNS). Chaque notification inclura un message d'état ainsi qu'un lien vers les ressources dont l'événement a généré la notification. En outre, avec les déclencheurs de référentiel AWS CodeCommit, vous pouvez envoyer des notifications et créer des webhooks HTTP avec Amazon SNS, ou appeler des fonctions AWS Lambda en réponse aux événements de référentiel de votre choix.

AWS CodeBuild

AWS CodeBuild est un service d'intégration continue entièrement géré qui compile votre code source, exécute des tests et produit des packages logiciels prêts à être déployés. Vous n'avez pas besoin de mettre en service, de gérer et de dimensionner vos serveurs de développement. CodeBuild peut utiliser GitHub, GitHub Enterprise, BitBucket, AWS CodeCommit ou Amazon S3 en tant que fournisseur de source.

CodeBuild est mis à l'échelle en continu et peut traiter plusieurs builds simultanément. CodeBuild propose divers environnements préconfigurés pour différentes versions de Microsoft Windows et Linux. Les clients peuvent également utiliser leurs environnements de construction personnalisés sous forme de conteneurs Docker. CodeBuild s'intègre également à des outils open source tels que Jenkins et Spinnaker.

CodeBuild peut également créer des rapports pour des tests unitaires, fonctionnels ou d'intégration. Ces rapports fournissent une représentation visuelle du nombre de cas de test exécutés et du nombre de cas ayant réussi ou échoué. Le processus de génération peut également être exécuté au sein d'un Amazon Virtual Private Cloud (Amazon VPC), ce qui peut s'avérer utile si vos services d'intégration ou vos bases de données sont déployés au sein d'un VPC.

AWS CodeArtifact

AWS CodeArtifact est un service de référentiels d'artefacts entièrement géré qui permet aux organisations de stocker, de publier et de partager facilement et en toute sécurité les packages logiciels utilisés dans leur processus de développement de logiciels. CodeArtifact peut être configuré pour récupérer automatiquement les logiciels et les dépendances des référentiels d'artefacts publics afin que les développeurs aient accès aux dernières versions.

Les équipes de développement de logiciels s'appuient de plus en plus sur des packages open source pour effectuer les tâches courantes de leur package d'application. Il est désormais essentiel pour les équipes de développement de logiciels de garder le contrôle d'une version particulière du logiciel open source afin de garantir qu'il n'est pas vulnérable. Avec CodeArtifact, vous pouvez configurer des contrôles pour appliquer cela.

CodeArtifact fonctionne avec les gestionnaires de packages couramment utilisés et crée des outils comme Maven, Gradle, npm, yarn, twine et pip, ce qui facilite l'intégration dans les flux de développement existants.

Livraison continue

La livraison continue est une méthode de développement de logiciels dans le cadre de laquelle les modifications de code sont automatiquement préparées en vue de leur publication dans un environnement de production. Véritable pilier du développement d'applications modernes, la livraison continue étend le principe de l'intégration continue en déployant tous les changements de code dans un environnement de test et/ou de production après l'étape de création. Lorsque la livraison continue est correctement implémentée, les développeurs disposent en permanence d'un artefact de génération prêt pour le déploiement qui a été soumis avec succès à un processus de test standardisé.

La livraison continue permet aux développeurs d'automatiser les tests au-delà des simples tests d'unité, afin de vérifier différents aspects d'une mise à jour d'application avant de la déployer auprès des clients. Il peut s'agir de tests d'interface, de charge, d'intégration, de fiabilité de l'API, etc. De cette manière, les développeurs peuvent vérifier de façon plus complète les mises à jour et détecter les éventuels problèmes à corriger avant le déploiement. Avec le cloud, l'automatisation de la création et de la réplication de plusieurs environnements de test est facile et économique, alors qu'une telle opération serait difficile à mettre en œuvre avec une infrastructure sur site.

AWS propose les services suivants pour la livraison continue :

- AWS CodeBuild
- AWS CodeDeploy
- AWS CodePipeline

Rubriques

- AWS CodeDeploy
- AWS CodePipeline

AWS CodeDeploy

AWS CodeDeploy est un service de déploiement entièrement géré qui automatise les déploiements de logiciels vers divers services de calcul tels qu'Amazon Elastic Compute Cloud (Amazon EC2), AWS Fargate, AWS Lambda et vos serveurs sur site. AWS CodeDeploy vous permet de rapidement lancer de nouvelles fonctions, vous aide à éviter les temps d'inactivité durant le déploiement

d'application et gère les tâches complexes de mise à jour de vos applications. Vous pouvez utiliser CodeDeploy pour automatiser les déploiements de logiciels, tout en éliminant le besoin des opérations manuelles susceptibles d'engendrer des erreurs. Le service se met à l'échelle pour correspondre à vos besoins de déploiement.

CodeDeploy présente plusieurs avantages qui s'alignent sur le principe DevOps de déploiement continu :

Déploiements automatisés : CodeDeploy automatise entièrement les déploiements des logiciels, vous permettant d'effectuer les déploiements de manière fiable et rapide.

Contrôle centralisé : CodeDeploy vous permet de lancer et de suivre facilement le statut de vos déploiements d'applications via la console de gestion AWS ou l'AWS CLI. CodeDeploy vous fournit un rapport détaillé qui vous permet de voir quand et où chaque révision d'application a été déployée. Vous pouvez également créer des notifications Push et recevoir des mises à jour en direct à propos de vos déploiements.

Réduire les temps d'arrêt : CodeDeploy vous permet de maximiser la disponibilité de votre application lors du processus de déploiement logiciel. Ce service introduit les modifications de manière incrémentielle et suit la santé de l'application en fonction des règles configurables. Les déploiements de logiciels peuvent être facilement arrêtés et restaurés si des erreurs se produisent.

Facile à adopter : CodeDeploy fonctionne avec n'importe quelle application et offre la même expérience quelle que soit la plateforme et la langue. Vous pouvez réutiliser facilement votre code de configuration existant. CodeDeploy peut s'intégrer également avec votre processus de lancement de logiciel existant ou votre chaîne de développement continue (par exemple AWS CodePipeline, GitHub, Jenkins).

AWS CodeDeploy prend en charge plusieurs options de déploiement. Pour de plus amples informations, veuillez consulter Stratégies de déploiement.

AWS CodePipeline

AWS CodePipeline est un service de livraison continue, qui vous permet de modéliser, visualiser et automatiser les étapes nécessaires à la publication de votre logiciel. Grâce à AWS CodePipeline, vous modélisez l'ensemble du processus de publication pour la création de votre code, son déploiement vers les environnements de pré-production, le test de votre application et son lancement en production. AWS CodePipeline crée, teste et déploie ensuite votre application en fonction du flux de travail défini, chaque fois que le code est modifié. Vous pouvez intégrer des outils de partenaires

APN et vos propres outils personnalisés à n'importe quel stade du processus de publication pour former une solution de diffusion continue de bout en bout.

AWS CodePipeline présente plusieurs avantages qui s'alignent sur le principe DevOps de déploiement continu :

Livraison rapide : AWS CodePipeline automatise votre processus de publication du logiciel, ce qui vous permet de mettre rapidement à la disposition des utilisateurs de nouvelles fonctionnalités. Avec CodePipeline, vous pouvez itérer rapidement en fonction des commentaires reçus et envoyer les nouvelles fonctions aux clients plus rapidement.

Amélioration de la qualité : en automatisant vos processus de création, de test et de publication, AWS CodePipeline vous permet d'augmenter la vitesse et la qualité des mises à jour de vos logiciels en vérifiant toutes les nouvelles modifications par le biais d'un ensemble cohérent de contrôles qualité.

Facile à intégrer : AWS CodePipeline peut facilement être étendu pour s'adapter à la spécificité de vos besoins. Vous pouvez utiliser les modules d'extension préintégrés ou vos modules d'extension personnalisés, et ce à n'importe quelle étape de votre processus de publication. Vous pouvez, par exemple, extraire votre code source de GitHub, utiliser votre serveur de développement Jenkins sur site, exécuter des tests de charge à l'aide d'un service tiers ou transférer les informations sur le déploiement vers le tableau de bord de vos opérations personnalisées.

Flux de travail configurable : AWS CodePipeline vous permet de modéliser les différentes étapes de votre processus de publication de logiciels à l'aide de l'interface de la console, de l'AWS CLI, AWS CloudFormation ou des kits SDK AWS. Vous pouvez facilement spécifier les tests à exécuter et personnaliser les étapes de déploiement de votre application et de ses dépendances.

Stratégies de déploiement

Les stratégies de déploiement définissent la manière dont vous souhaitez mettre à disposition votre logiciel. Les organisations suivent différentes stratégies de déploiement en fonction de leur modèle opérationnel. Certaines peuvent choisir de fournir un logiciel entièrement testé, tandis que d'autres peuvent souhaiter que leurs utilisateurs fournissent des commentaires et laissent leurs utilisateurs évaluer les fonctions en cours de développement (par exemple, dans le cas de versions bêta). Dans la section suivante, nous aborderons différentes stratégies de déploiement.

Rubriques

Déploiements sur place

Dans cette stratégie, le déploiement est effectué dans les conditions suivantes : l'application sur chaque instance du groupe de déploiement est arrêtée, la dernière révision de l'application est installée et la nouvelle version de l'application est lancée et validée. Vous pouvez choisir d'utiliser un équilibreur de charge afin d'annuler l'inscription de chaque instance lors de son déploiement, et de la restaurer dans le service une fois le déploiement terminé. Les déploiements sur place peuvent être réalisés simultanément, en supposant une panne de service, ou réalisés sous forme de mise à jour progressive. AWS CodeDeploy et AWS Elastic Beanstalk proposent des configurations pour un déploiement un à la fois, la moitié en une fois et en une seule fois. Ces stratégies de déploiement pour les déploiements sur place sont disponibles dans les déploiements bleu/vert.

Déploiements bleu/vert

Le déploiement bleu/vert, parfois appelé rouge-noir, est une technique permettant de publier des applications en alternant le trafic entre deux environnements identiques exécutant des versions différentes de l'application. Les déploiements bleu/vert vous aident à minimiser les temps d'arrêt pendant les mises à jour des applications, en atténuant les risques liés aux temps d'arrêt et à la restauration. Les déploiements bleu/vert vous permettent de lancer une nouvelle version (verte) de

votre application parallèlement à l'ancienne version (bleue), et de surveiller et de tester la nouvelle version avant de rediriger le trafic vers celle-ci, avec la possibilité de restaurer si des problèmes sont détectés.

Déploiements canari

Le trafic est déplacé en deux incréments. Un déploiement Canary est une stratégie bleu/vert plus prudente, dans laquelle une approche progressive est utilisée. Pendant ce déploiement, qui peut être mené en deux étapes ou être linéaire, un nouveau code d'application est déployé et exposé à des fins d'essai. Lorsqu'il est accepté, il est déployé dans le reste de l'environnement ou de manière linéaire.

Déploiements linéaires

Lors des déploiements linéaires, le trafic est déplacé en incréments égaux avec un nombre égal de minutes entre chaque incrément. Vous pouvez choisir parmi les options linéaires prédéfinies qui définissent le pourcentage de trafic déplacé pour chaque incrément et le nombre de minutes entre chaque incrément.

Déploiements simultanés

Pendant les déploiements simultanés, l'ensemble du trafic est transféré de l'environnement d'origine vers l'environnement de remplacement en une seule fois.

Matrice de stratégies de déploiement

La matrice suivante répertorie les stratégies de déploiement prises en charge pour Amazon Elastic Container Service (Amazon ECS), AWS Lambda et Amazon EC2/sur site.

- Amazon ECS est un service d'orchestration entièrement géré.
- AWS Lambda vous permet d'exécuter du code sans avoir à allouer ni gérer de serveurs.
- Amazon EC2 vous permet d'exécuter une capacité de calcul redimensionnable et sécurisée dans le cloud.

	A	B	C	D
1	Matrice de stratégies de déploiement	Amazon ECS	AWS Lambda	Amazon EC2/sur site
2	Sur place	✓	✓	✓
3	Bleu/vert	✓	✓	✓*
4	Canary	✓	✓	X
5	Linéaire	✓	✓	X
6	Simultané	✓	✓	X

> **Note**
>
> Le déploiement bleu/vert avec EC2/sur site ne fonctionne qu'avec les instances EC2.

Stratégies de déploiement AWS Elastic Beanstalk

AWS Elastic Beanstalk prend en charge les types de stratégies de déploiement suivants :

- Simultané : effectue un déploiement sur place sur toutes les instances.

- Propagation : divise les instances en lots et les déploie lot après lot.
- Propagation avec un lot supplémentaire : divise les déploiements en lots, mais pour le premier lot, crée de nouvelles instances EC2 au lieu de les déployer sur les instances EC2 existantes.
- Inaltérable : si vous devez déployer avec une nouvelle instance au lieu d'utiliser une instance existante.
- Fractionnement du trafic : procède à un déploiement inaltérable, puis transfère le pourcentage du trafic vers les nouvelles instances pendant une durée prédéterminée. Si les instances restent saines, transférez l'ensemble du trafic vers les nouvelles instances et fermez les anciennes instances.

Infrastructure as Code

Un principe fondamental du DevOps est de traiter l'infrastructure de la même manière que les développeurs traitent le code. Le code d'application possède 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 ou de contrôle de code source qui enregistre un historique du développement du code, des modifications et des corrections de bogues. Lorsque le code est compilé ou intégré à des applications, nous nous attendons à ce qu'une application cohérente soit créée et que la build soit reproductible et fiable.

Utiliser Infrastructure as Code signifie appliquer la même rigueur de 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 contrôle de source tel que AWS CodeCommit, comme pour le code d'application. La mise en service, l'orchestration et le déploiement de l'infrastructure doivent également prendre en charge l'utilisation de Infrastructure as Code.

Traditionnellement, l'infrastructure était mise en service à l'aide d'une combinaison de scripts et de processus manuels. Parfois, ces scripts étaient stockés dans des systèmes de contrôle de version ou documentés étape par étape dans des fichiers texte ou des runbooks. Souvent, la personne qui écrit les runbooks n'est pas la même que celle qui exécute ces scripts ou qui applique les runbooks. Si ces scripts ou ces runbooks ne sont pas mis à jour fréquemment, ils peuvent devenir un obstacle majeur dans les déploiements. Les nouveaux environnements créés peuvent alors ne pas être reproductibles, fiables ou cohérents.

À l'inverse de ce qui précède, AWS fournit une méthode orientée DevOps pour créer et entretenir une infrastructure. De la même façon que les développeurs logiciels écrivent le code de l'application, AWS fournit des services qui permettant la création, le déploiement et la maintenance de l'infrastructure de façon déclarative, descriptive et par programme. Ces services apportent rigueur, clarté et fiabilité. Les services AWS présentés dans ce document sont essentiels à une méthodologie DevOps et constituent la base de nombreux principes et pratiques AWS DevOps de niveau supérieur.

AWS propose les services suivants pour définir l'infrastructure en tant que code.

- AWS CloudFormation
- AWS Cloud Development Kit (AWS CDK)
- AWS Cloud Development Kit pour Kubernetes

AWS CloudFormation

AWS CloudFormation est un service qui permet aux développeurs de créer des ressources AWS de manière ordonnée et prévisible. Les ressources sont écrites dans des fichiers texte au format JSON (JavaScript Object Notation) ou YALM (Yet Another Markup Language). Les modèles nécessitent une syntaxe et une structure spécifiques qui dépendent des types de ressources créées et gérées. Vous créez vos ressources au format JSON ou YAML avec n'importe quel éditeur de code tel qu'AWS Cloud9, vous les archivez dans un système de contrôle de version, puis CloudFormation crée les services spécifiés de manière sûre et reproductible.

Un modèle CloudFormation est déployé dans l'environnement AWS en tant que pile. Vous pouvez gérer les piles via la console de gestion AWS, l'interface de ligne de commande AWS ou les API AWS CloudFormation. Pour modifier les ressources en cours d'exécution d'une pile, vous devez mettre à jour la pile. Avant d'apporter des modifications à vos ressources, vous pouvez générer un jeu de modifications, qui représente un résumé des modifications proposées. Les jeux de modification vous permettent d'avoir un aperçu de l'impact possible des modifications d'une pile sur les ressources exécutées (y compris les ressources critiques) avant d'appliquer ces modifications.

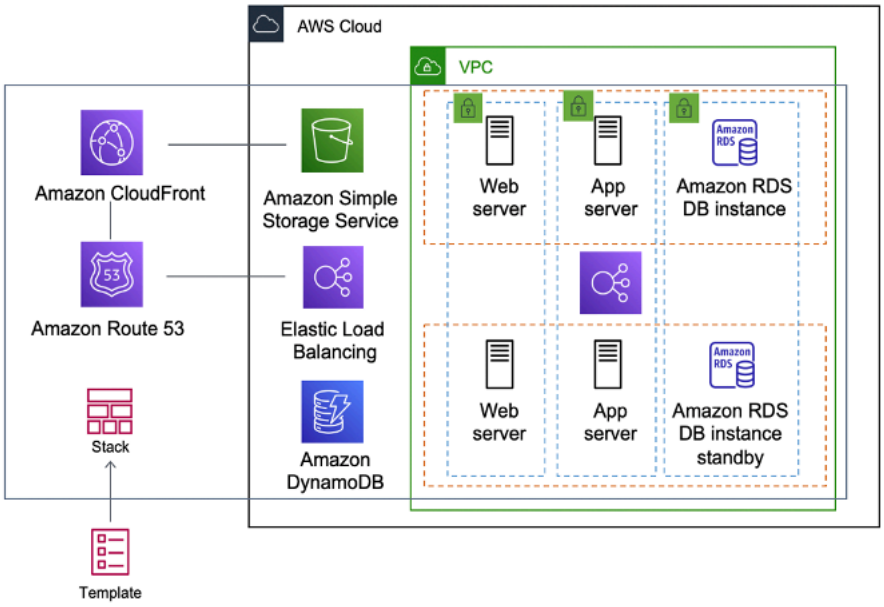

Figure 1 - AWS CloudFormation crée un environnement complet (pile) à partir d'un seul modèle de flux de travail

Vous pouvez utiliser un seul modèle pour créer et mettre à jour un environnement entier ou des modèles distincts pour gérer plusieurs couches au sein d'un environnement. Cela permet d'organiser

par modules les modèles et fournit également une couche de gouvernance importante pour de nombreuses organisations.

Lorsque vous créez ou mettez à jour une pile dans la console, des événements s'affichent pour indiquer l'état de la configuration. En cas d'erreur, la pile est restaurée par défaut à son état précédent. Amazon Simple Notification Service (Amazon SNS) fournit des notifications sur les événements. Par exemple, vous pouvez utiliser Amazon SNS pour suivre l'avancement de la création et de la suppression d'une pile par e-mail, et bénéficier d'une intégration aux autres processus par programme.

AWS CloudFormation facilite l'organisation et le déploiement d'un ensemble de ressources AWS et vous permet de décrire n'importe quelle dépendance ainsi que de transmettre des paramètres spéciaux lorsque la pile est configurée.

Avec les modèles CloudFormation, vous pouvez travailler avec un large éventail de services AWS, tels qu'Amazon S3, Auto Scaling, Amazon CloudFront, Amazon DynamoDB, Amazon EC2, Amazon ElastiCache, AWS Elastic Beanstalk, Elastic Load Balancing, AWS OpsWorks, IAM et Amazon VPC. Pour obtenir la liste la plus récente des ressources prises en charge, veuillez consulter Référence des types de propriété et de ressource AWS.

AWS Cloud Development Kit

AWS Cloud Development Kit (AWS CDK) est un cadre de développement de logiciel open source permettant de modéliser et d'allouer les ressources de votre application cloud à l'aide de langages de programmation courants. Grâce à AWS CDK, vous pouvez modéliser l'infrastructure des applications à l'aide de TypeScript, Python, Java et .NET. Les développeurs peuvent tirer parti de leur environnement de développement intégré (IDE) existant, en tirant parti d'outils tels que le remplissage automatique et la documentation en ligne pour accélérer le développement de l'infrastructure.

AWS CDK utilise AWS CloudFormation en arrière-plan pour provisionner des ressources de manière sûre et reproductible. Les constructions sont les éléments de base du code CDK. Une construction représente un composant cloud et encapsule tout ce dont AWS CloudFormation a besoin pour créer le composant. AWS CDK inclut la bibliothèque AWS Construct, qui contient des constructions représentant de nombreux services AWS. En combinant des constructions, vous pouvez rapidement et facilement créer des architectures complexes pour le déploiement dans AWS.

AWS Cloud Development Kit pour Kubernetes

Le kit AWS Cloud Development Kit pour Kubernetes (cdk8s) est un framework de développement logiciel open source qui permet de définir les applications Kubernetes à l'aide de langages de programmation à usage général.

Une fois que vous avez défini votre application dans un langage de programmation (à la date de publication, seuls Python et TypeScript sont pris en charge), cdk8s convertira la description de votre application en YML antérieur à Kubernetes. Ce fichier YML peut ensuite être utilisé par n'importe quel cluster Kubernetes, quel que soit l'endroit où il est exécuté. La structure étant définie dans un langage de programmation, vous pouvez utiliser les fonctions avancées fournies par le langage de programmation. Vous pouvez utiliser la fonction d'abstraction du langage de programmation pour créer votre propre code standard et le réutiliser dans tous les déploiements.

Automatisation

L'automatisation est une autre philosophie et pratique fondamentale du DevOps. L'automatisation se concentre sur la configuration, le déploiement et le support de l'infrastructure et des applications qui s'y exécutent. L'automatisation vous permet de configurer des environnements plus rapidement de manière standardisée et reproductible. La suppression des processus manuels est essentielle à la réussite d'une stratégie DevOps. Historiquement, la configuration des serveurs et le déploiement des applications étaient essentiellement des processus manuels. Les environnements deviennent non standard et il est difficile de reproduire un environnement lorsque des problèmes surviennent.

L'utilisation de l'automatisation est essentielle pour tirer pleinement parti des avantages du cloud. En interne, AWS s'appuie fortement sur l'automatisation pour fournir les principales fonctionnalités d'élasticité et de capacité de mise à l'échelle. Les processus manuels sont source d'erreurs, peu fiables et inadaptés pour soutenir une entreprise agile. Il arrive fréquemment qu'une organisation mobilise des ressources hautement qualifiées pour fournir une configuration manuelle, alors qu'il est préférable de consacrer du temps à d'autres activités plus critiques et à plus forte valeur ajoutée au sein de l'entreprise.

Les environnements d'exploitation modernes reposent généralement sur une automatisation complète qui vise à supprimer les interventions manuelles ou l'accès aux environnements de production. Elle inclut la publication de tous les logiciels, la configuration des machines, les correctifs du système d'exploitation, le dépannage ou la correction de bogues. De nombreux niveaux de pratiques d'automatisation peuvent être utilisés ensemble pour fournir un processus automatisé de bout en bout de niveau supérieur.

L'automatisation présente les principaux avantages suivants :

- Changements rapides
- Amélioration de la productivité
- Configurations reproductibles
- Environnements reproductibles
- Exploitation de d'élasticité
- Exploitation de la scalabilité automatique
- Tests automatisés

L'automatisation est la pierre angulaire des services AWS. Elle est prise en charge en interne dans tous les services, fonctions et offres.

Rubriques

- AWS OpsWorks
- AWS Elastic Beanstalk

AWS OpsWorks

AWS OpsWorks amène les principes de DevOps encore plus loin qu'AWS Elastic Beanstalk. Ce service peut être considéré comme un service de gestion d'applications plutôt qu'un simple conteneur d'applications. AWS OpsWorks fournit encore plus de niveaux d'automatisation avec des fonctions supplémentaires telles que l'intégration au logiciel de gestion de la configuration (Chef) et à la gestion du cycle de vie des applications. Vous pouvez utiliser la gestion du cycle de vie des applications pour définir le moment où les ressources sont configurées, déployées, annulées ou arrêtées.

Pour plus de flexibilité, AWS OpsWorks vous permet de définir votre application dans des piles configurables. Vous pouvez également sélectionner des piles d'applications prédéfinies. Les piles d'applications contiennent l'ensemble de l'allocation des ressources AWS dont votre application a besoin, y compris les serveurs d'applications, les serveurs web, les bases de données et les équilibreurs de charge.

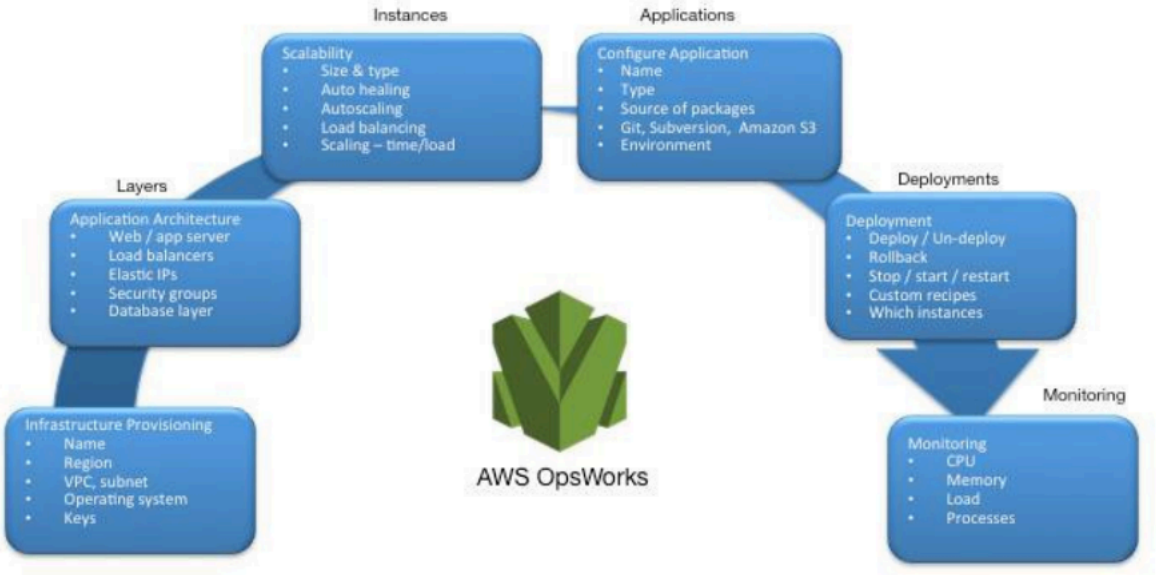

Figure 2 - AWS OpsWorks présentant les fonctions et l'architecture DevOps

Les piles d'applications sont organisées en couches architecturales afin que les piles puissent être gérées de manière indépendante. Le niveau Web, le niveau Application et le niveau Base de données constituent des exemples de couches. Prêt à l'emploi, AWS OpsWorks simplifie également la configuration des groupes Auto Scaling et des équilibreurs de charge Elastic Load Balancing. Ce service illustra encore davantage le principe d'automatisation DevOps. Tout comme AWS Elastic Beanstalk, AWS OpsWorks prend en charge la gestion des versions des applications, le déploiement continu et la gestion de la configuration de l'infrastructure.

AWS OpsWorks prend également en charge les pratiques DevOps de surveillance et de journalisation (abordées dans la section suivante). La prise en charge de la surveillance est fournie par Amazon CloudWatch. Tous les événements du cycle de vie sont consignés et un journal Chef distinct documente toutes les recettes Chef exécutées, ainsi que les exceptions.

AWS Elastic Beanstalk

AWS Elastic Beanstalk est un service permettant de déployer et de mettre à l'échelle des applications et des services web développés avec Java, .NET, PHP, Node.js, Python, Ruby, Go et Docker sur des serveurs connus, tels qu'Apache, Nginx, Passenger et IIS.

Elastic Beanstalk est une abstraction au-dessus d'Amazon EC2, Auto Scaling, et simplifie le déploiement en offrant des fonctions supplémentaires telles que le clonage, les déploiements bleu/vert, l'interface de ligne de commande Elastic Beanstalk (cli eb) et l'intégration avec AWS Toolkit for Visual Studio, Visual Studio Code, Eclipse et IntelliJ pour accroitre la productivité des développeurs.

Surveillance et journalisation

La communication et la collaboration sont fondamentales dans une philosophie DevOps. Et les commentaires sont essentiels pour réussir. Dans AWS, les commentaires sont fournis par deux services principaux : Amazon CloudWatch et AWS CloudTrail. Ensemble, ces services fournissent une infrastructure robuste de surveillance, d'alerte et d'audit afin que les développeurs et les équipes opérationnelles puissent travailler ensemble de manière étroite et transparente.

AWS fournit les services suivants pour la surveillance et la journalisation :

Rubriques

- Amazon CloudWatch
- Alarmes Amazon CloudWatch
- Amazon CloudWatch Logs
- Amazon CloudWatch Logs Insights
- Amazon CloudWatch Events
- Amazon EventBridge
- AWS CloudTrail

Amazon CloudWatch

Les métriques Amazon CloudWatch collectent automatiquement les données des services AWS tels que les instances Amazon EC2, les volumes Amazon EBS et les instances de bases de données Amazon RDS. Ces mesures peuvent ensuite être organisées sous forme de tableaux de bord, et des alarmes ou des événements peuvent être créés pour déclencher des événements ou effectuer des actions Auto Scaling.

Alarmes Amazon CloudWatch

Vous pouvez configurer des alarmes en fonction des métriques collectées par Métriques Amazon CloudWatch. L'alarme peut ensuite envoyer une notification à la rubrique Amazon Simple Notification Service (Amazon SNS) ou initier des actions Auto Scaling. Une alarme nécessite une période (durée d'évaluation d'une métrique), une période d'évaluation (nombre de points de données les plus récents) et une valeur de points de données avant l'alarme (nombre de points de données au cours de la période d'évaluation).

Amazon CloudWatch Logs

Amazon CloudWatch Logs est un service d'agrégation et de surveillance des journaux. AWS CodeBuild, CodeCommit, CodeDeploy et CodePipeline fournissent des intégrations avec les journaux CloudWatch afin que tous les journaux puissent être surveillés de manière centralisée. En outre, les services mentionnés précédemment et divers autres services AWS fournissent une intégration directe avec CloudWatch.

Avec CloudWatch Logs, vous pouvez :

- interroger les données de vos journaux ;
- surveiller les journaux des instances Amazon EC2 ;
- surveiller les événements consignés dans AWS CloudTrail ;
- définir la stratégie de conservation des journaux ;

Amazon CloudWatch Logs Insights

Amazon CloudWatch Logs Insights analyse vos journaux et vous permet d'effectuer des requêtes et des visualisations interactives. Il comprend différents formats de journaux et détecte automatiquement les champs des journaux JSON.

Amazon CloudWatch Events

Amazon CloudWatch Events fournit un flux d'événements système en quasi temps réel décrivant les modifications apportées aux ressources AWS. À l'aide de règles simples et rapidement configurées, vous pouvez faire correspondre des événements et les acheminer vers un ou plusieurs flux, ou une ou plusieurs fonctions cibles. CloudWatch Events prend connaissance des changements opérationnels au fur et à mesure qu'ils surviennent. CloudWatch Events répond à ces changements opérationnels et, le cas échéant, prend des mesures correctives en envoyant des messages pour répondre à l'environnement, en activant des fonctions, en procédant à des modifications et en capturant des informations de statut.

Vous pouvez configurer des règles dans CloudWatch Events pour vous alerter des modifications apportées aux services AWS et intégrer ces événements à d'autres systèmes tiers à l'aide d'Amazon EventBridge. Voici les services liés à AWS DevOps qui sont intégrés à CloudWatch Events.

- Événements Application Auto Scaling

- Événements CodeBuild
- Événements CodeCommit
- Événements CodeDeploy
- Événements CodePipeline

Amazon EventBridge

Amazon CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités.

Amazon EventBridge est un bus d'événements sans serveur qui permet des intégrations entre les services AWS, les logiciels en tant que services (SaaS) et vos applications. En plus de créer des applications basées sur les événements, EventBridge peut être utilisé pour notifier les événements à partir de services tels que CodeBuild, CodeDeploy, CodePipeline et CodeCommit.

AWS CloudTrail

Pour adopter les principes DevOps de collaboration, de communication et de transparence, il est important de comprendre qui apporte des modifications à votre infrastructure. Dans AWS, cette transparence est assurée par le service AWS CloudTrail. Toutes les interactions AWS sont gérées par le biais d'appels d'API AWS qui sont surveillés et consignés 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) Amazon S3. Tous les appels d'API sont consignés, qu'ils proviennent directement d'un utilisateur ou d'un service AWS pour le compte d'un utilisateur. De nombreux groupes peuvent tirer parti des journaux CloudTrail, notamment les équipes opérationnelles pour le support, les équipes de sécurité pour la gouvernance et les équipes financières pour la facturation.

Communication et collaboration

Que vous adoptiez la culture DevOps dans votre organisation ou que vous optiez pour une communication sur la transformation culturelle DevOps, la collaboration est un élément important de votre approche. Chez Amazon, nous avons compris qu'il était nécessaire d'apporter un changement dans l'état d'esprit des équipes et avons donc adopté le concept d'équipes two-pizza.

Rubriques

Équipes two-pizza

« Nous essayons de créer des équipes auxquelles deux pizzas peuvent suffire », a déclaré Jeff Bezos. « C'est ce que nous appelons la règle des équipes two-pizza. »

Plus l'équipe est petite, meilleure est la collaboration. La collaboration est également très importante, car les versions logicielles évoluent plus rapidement que jamais. Et la capacité d'une équipe à fournir le logiciel peut être un facteur de différenciation pour votre organisation par rapport à vos concurrents. Imaginez une situation dans laquelle une nouvelle fonction du produit doit être publiée ou dans laquelle un bogue doit être corrigé. Vous souhaitez que cela se produise le plus rapidement possible afin de pouvoir bénéficier d'un délai de mise sur le marché plus court. C'est également important, car vous ne voulez pas que la transformation soit un processus lent plutôt qu'une approche agile où des vagues de changements entraînent des répercussions.

La communication entre les équipes est également importante : nous nous dirigeons vers le modèle de responsabilité partagée et commençons à sortir de l'approche de développement cloisonnée. Le concept de propriété s'impose alors à l'équipe et fait évoluer le point de vue de l'équipe qui doit le considérer comme un concept de bout en bout. Votre équipe ne doit pas considérer vos environnements de production comme des boîtes noires dans lesquelles elles n'ont aucune visibilité.

La transformation culturelle est également importante si vous créez une équipe DevOps commune. L'autre approche consiste à dédier un ou plusieurs membres de l'équipe au DevOps. Ces deux approches introduisent une responsabilité partagée au sein de l'équipe.

Sécurité

Que vous opériez une transformation DevOps ou que vous mettiez en œuvre les principes DevOps pour la première fois, vous devez considérer que la sécurité doit être intégrée à vos processus DevOps. La sécurité doit être une préoccupation transversale des étapes de création et de test du déploiement.

Avant d'aborder la sécurité dans DevOps sur AWS, examinons le modèle de responsabilité partagée d'AWS.

Rubriques

Modèle de responsabilité partagée d'AWS

AWS et le client se partagent la responsabilité d'assurer la sécurité. Les différentes parties du modèle de responsabilité partagée sont expliquées ci-dessous :

- Responsabilité d'AWS : « sécurité du cloud ». AWS est responsable de la protection de l'infrastructure exécutant tous les services proposés dans le cloud AWS. Cette infrastructure se compose de matériel, de logiciels, de réseaux et d'installations exécutant les services AWS Cloud services.
- Responsabilité du client : « sécurité dans le cloud ». La responsabilité du client est déterminée en fonction des services AWS Cloud services que ce dernier choisit. Cette responsabilité détermine la quantité de travail de configuration que doit réaliser le client dans le cadre de ses responsabilités en matière de sécurité.

Ce modèle partagé peut atténuer la charge opérationnelle qui pèse sur le client, car AWS exploite, gère et commande les composants depuis le système d'exploitation hôte et la couche de virtualisation jusqu'à la sécurité physique des installations dans lesquelles les services sont exploités. Dans les cas où le client souhaite comprendre la sécurité de ses environnements de génération, cela s'avère essentiel.

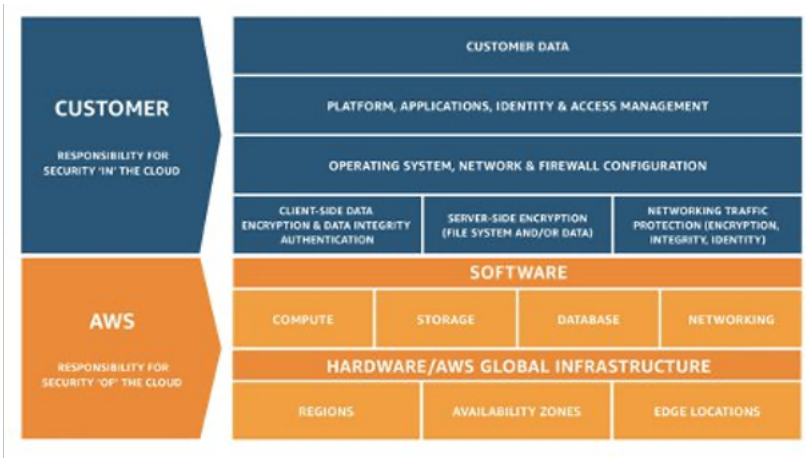

Figure 3 - Modèle de responsabilité partagée d'AWS

Gestion des identités et des accès

AWS Identity and Access Management (IAM) définit les contrôles et les politiques utilisés pour gérer l'accès aux ressources AWS. À l'aide d'IAM, vous pouvez créer des utilisateurs et des groupes, et définir des autorisations pour divers services DevOps.

Outre les utilisateurs, divers services peuvent également avoir besoin d'accéder aux ressources AWS. Par exemple, votre projet CodeBuild peut avoir besoin d'un accès pour stocker des images Docker dans Amazon Elastic Container Registry (Amazon ECR) ainsi que d'autorisations pour écrire sur Amazon ECR. Ces types d'autorisations sont définis par un rôle de type spécial appelé « fonction du service ».

IAM est l'un des composants de l'infrastructure de sécurité AWS. Avec IAM, vous pouvez gérer de façon centralisée les groupes, les utilisateurs, les fonctions du service 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. La politique IAM vous permet de définir l'ensemble des autorisations. Cette politique peut ensuite être attachée à un rôle, à un utilisateur ou à un service pour définir son autorisation. Vous pouvez également utiliser IAM pour créer des rôles largement utilisés dans le cadre de la stratégie DevOps souhaitée. Dans certains cas, il peut être parfaitement logique d'effectuer une action AssumeRole par programme au lieu d'obtenir directement les autorisations. Lorsqu'un service ou un utilisateur assume des rôles, il reçoit des informations d'identification temporaires qui lui permettent d'accéder à un service auquel vous n'avez normalement pas accès.

Conclusion

Pour rendre la transition vers le cloud fluide, efficiente et efficace, les entreprises technologiques doivent adopter les principes et les pratiques DevOps. Ces principes sont intégrés à la plateforme AWS. En effet, ils constituent la pierre angulaire de nombreux services AWS, en particulier ceux des offres de déploiement et de surveillance.

Commencez par définir votre infrastructure en tant que code (Infrastructure as Code) à l'aide du service AWS CloudFormation ou AWS Cloud Development Kit (AWS CDK). Ensuite, définissez la manière dont vos applications utiliseront le déploiement continu à l'aide de services tels que AWS CodeBuild, AWS CodeDeploy, AWS CodePipeline et AWS CodeCommit. Au niveau de l'application, utilisez des conteneurs tels qu'AWS Elastic Beanstalk, Amazon Elastic Container Service (Amazon ECS) ou Amazon Elastic Kubernetes Service (Amazon EKS), et AWS OpsWorks pour simplifier la configuration d'architectures courantes. L'utilisation de ces services facilite également l'inclusion d'autres services importants tels qu'Auto Scaling et Elastic Load Balancing. Enfin, utilisez la stratégie DevOps de surveillance telle qu'Amazon CloudWatch et des pratiques de sécurité solides telles qu'AWS IAM.

En collaborant avec AWS, vos principes DevOps apportent de l'agilité à votre entreprise et à votre organisation informatique, et accélèrent votre transition vers le cloud.

Révisions du document

Pour être informé des mises à jour de ce livre blanc, abonnez-vous au flux RSS.

update-history-change	update-history-description	update-history-date
Restauration de la section manquante relative aux contributeurs	Restauration de la section manquante relative aux contributeurs et modifications mineures du texte	21 novembre 2020
Sections mises à jour pour inclure de nouveaux services	Sections mises à jour pour inclure de nouveaux services	16 octobre 2020
Publication initiale	Première publication du livre blanc	1 décembre 2014

Participants

Ont contribué à la préparation du présent document :

- Muhammad Mansoor, Architecte de solutions
- Ajit Zadgaonkar, World Wide Tech Leader, Modernization
- Juan Lamadrid, Architecte de solutions
- Darren Ball, Architecte de solutions
- Rajeswari Malladi, Architecte de solutions
- Pallavi Nargund, Architecte de solutions
- Bert Zahniser, Architecte de solutions
- Abdullahi Olaoye, Architecte de solutions cloud
- Mohamed Kiswani, Responsable du développement logiciel
- Tara McCann, Responsable senior Architecte de solutions

Mentions légales

Les clients sont responsables de leur propre évaluation indépendante des informations contenues dans ce document. Le présent document : (a) est fourni à titre informatif uniquement, (b) représente les offres et pratiques actuelles de produits AWS, qui sont susceptibles d'être modifiées sans préavis, et (c) ne crée aucun engagement ou assurance de la part d'AWS et de ses affiliés, fournisseurs ou concédants de licences. Les produits ou services AWS sont fournis « en l'état » sans garantie, représentation ou condition, de quelque nature que ce soit, explicite ou implicite. Les responsabilités et obligations d'AWS envers ses clients sont déterminées par les contrats AWS, et le présent document ne fait pas partie d'un contrat entre AWS et ses clients, ni le modifie.