



Livre blanc AWS

# Présentation des options de déploiement sur AWS



# Présentation des options de déploiement sur AWS: Livre blanc AWS

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Les marques et la présentation commerciale d'Amazon ne peuvent être utilisées en relation avec un produit ou un service qui n'est pas d'Amazon, d'une manière susceptible de créer une confusion parmi les clients, ou d'une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d'Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

# Table of Contents

Résumé .....	1
Résumé .....	1
Introduction .....	2
Services de déploiement AWS .....	3
AWS CloudFormation .....	3
AWS Elastic Beanstalk .....	6
AWS CodeDeploy .....	10
AWS CodeDeploy pour AWS Lambda .....	13
Amazon Elastic Container Service .....	14
Amazon ECS Anywhere .....	18
Amazon Elastic Container Service sur AWS Outposts .....	19
Amazon Elastic Kubernetes Service .....	19
Amazon EKS Anywhere .....	24
AWS App Runner .....	24
Amazon Lightsail .....	26
Conteneurs Amazon Lightsail .....	26
Red Hat OpenShift Service on AWS .....	27
Zones locales AWS .....	27
AWS Wavelength .....	28
Services de déploiement supplémentaires .....	28
Amazon Simple Storage Service .....	28
AWS Proton .....	29
AWS App2Container .....	29
Copilote AWS .....	30
AWS Serverless Application Model .....	30
AWS Cloud Development Kit (AWS CDK) .....	31
Amazon EC2 Image Builder .....	32
Stratégies de déploiement .....	34
Précuisson ou amorçage AMIs .....	34
Déploiements bleus/verts .....	34
Déploiements propagés .....	35
Déploiements Canary .....	35
Déploiements sur place .....	36
Combiner les services de déploiement .....	36

---

Conclusion .....	38
Collaborateurs .....	39
Suggestions de lecture .....	40
Révisions du document .....	41
Notifications .....	42
.....	xliii

# Présentation des options de déploiement sur AWS

Date de publication : 31 mai 2024 ([Révisions du document](#))

## Résumé

Amazon Web Services (AWS) propose plusieurs options pour le provisionnement de l'infrastructure et le déploiement de vos applications. Que votre architecture d'application soit une simple application Web à trois niveaux ou un ensemble complexe de charges de travail, AWS propose des services de déploiement répondant aux exigences de votre application et de votre organisation.

Ce livre blanc est destiné aux personnes qui souhaitent avoir une vue d'ensemble des différents services de déploiement proposés par AWS. Il présente les fonctionnalités communes disponibles dans ces services de déploiement et définit les stratégies de base pour le déploiement et la mise à jour des piles d'applications.

# Introduction

La conception d'une solution de déploiement pour votre application est essentielle à la création d'une application bien conçue sur AWS. En fonction de la nature de votre application et des services sous-jacents dont elle a besoin, vous pouvez utiliser les services AWS pour créer une solution de déploiement flexible qui peut être adaptée aux besoins de votre application et de votre organisation.

Le catalogue sans cesse croissant de services AWS complique non seulement le processus de sélection des services qui composeront l'architecture de votre application, mais également le processus de sélection de la manière dont vous allez créer, gérer et mettre à jour votre application. Lorsque vous concevez une solution de déploiement sur AWS, vous devez réfléchir à la manière dont votre solution répondra aux fonctionnalités suivantes :

- **Fourniture** : créez l'infrastructure brute ou l'infrastructure de services gérés requise pour votre application.
- **Configuration** : personnalisez votre infrastructure en fonction de l'environnement, du temps d'exécution, de la sécurité, de la disponibilité, des performances, du réseau ou des autres exigences des applications.
- **Déploiement** : installez ou mettez à jour les composants de votre application sur les ressources de l'infrastructure et gérez la transition d'une version d'application précédente vers une nouvelle version d'application.
- **Échelle** : ajustez de manière proactive ou réactive la quantité de ressources disponibles pour votre application en fonction d'un ensemble de critères définis par l'utilisateur.
- **Surveiller** : offrez une visibilité sur les ressources lancées dans le cadre de l'architecture de votre application. Suivez l'utilisation des ressources, le succès ou l'échec du déploiement, l'état des applications, les journaux des applications, les erreurs de configuration, etc.

Ce livre blanc présente les services de déploiement proposés par AWS et décrit les stratégies permettant de concevoir une architecture de déploiement efficace pour tout type d'application.

# Services de déploiement AWS

La tâche de conception d'une solution de déploiement évolutive, efficace et rentable ne doit pas se limiter à la manière dont vous allez mettre à jour la version de votre application, mais doit également prendre en compte la manière dont vous allez gérer l'infrastructure de support tout au long du cycle de vie des applications. Le provisionnement des ressources, la gestion de la configuration, le déploiement d'applications, les mises à jour logicielles, la surveillance, le contrôle d'accès et d'autres préoccupations sont autant de facteurs importants à prendre en compte lors de la conception d'une solution de déploiement.

Les services AWS peuvent fournir des fonctionnalités de gestion pour un ou plusieurs aspects du cycle de vie de vos applications. En fonction de l'équilibre souhaité entre contrôle (gestion manuelle des ressources) et commodité (gestion des ressources AWS) et du type d'application, ces services peuvent être utilisés seuls ou combinés pour créer une solution de déploiement riche en fonctionnalités. Cette section fournit une vue d'ensemble des services AWS qui peuvent être utilisés pour permettre aux entreprises de créer et de fournir des applications de manière plus rapide et plus fiable.

## AWS CloudFormation

[AWS CloudFormation](#) est un service qui permet aux clients de fournir et de gérer presque toutes les ressources AWS à l'aide d'un langage de modèle personnalisé exprimé en YAML ou JSON. Un CloudFormation modèle crée des ressources d'infrastructure dans un groupe appelé pile, et vous permet de définir et de personnaliser tous les composants nécessaires au fonctionnement de votre application tout en conservant le contrôle total de ces ressources. L'utilisation de modèles permet d'implémenter le contrôle de version sur votre infrastructure et de répliquer rapidement et de manière fiable votre infrastructure.

CloudFormation offre un contrôle granulaire du provisionnement et de la gestion de tous les composants de l'infrastructure d'applications, des composants de bas niveau tels que les tables de routage ou les configurations de sous-réseau aux composants de haut niveau tels que les distributions. CloudFront CloudFormation est couramment utilisé avec d'autres services de déploiement AWS ou des outils tiers, en combinaison CloudFormation avec des services de déploiement plus spécialisés pour gérer les déploiements de code d'application sur les composants de l'infrastructure.

AWS propose des extensions au CloudFormation service en plus de ses fonctionnalités de base :

- [AWS Cloud Development Kit \(AWS CDK\)](#) est un kit de développement logiciel (SDK) open source permettant de modéliser par programmation l'infrastructure AWS avec JavaScript Python TypeScript, Java ou C#/.NET.
- [AWS Serverless Application Model \(AWS SAM\)](#) est un framework open source destiné à simplifier la création d'applications sans serveur sur AWS. Il fournit une syntaxe abrégée pour exprimer les fonctions APIs, les bases de données et les mappages de sources d'événements.

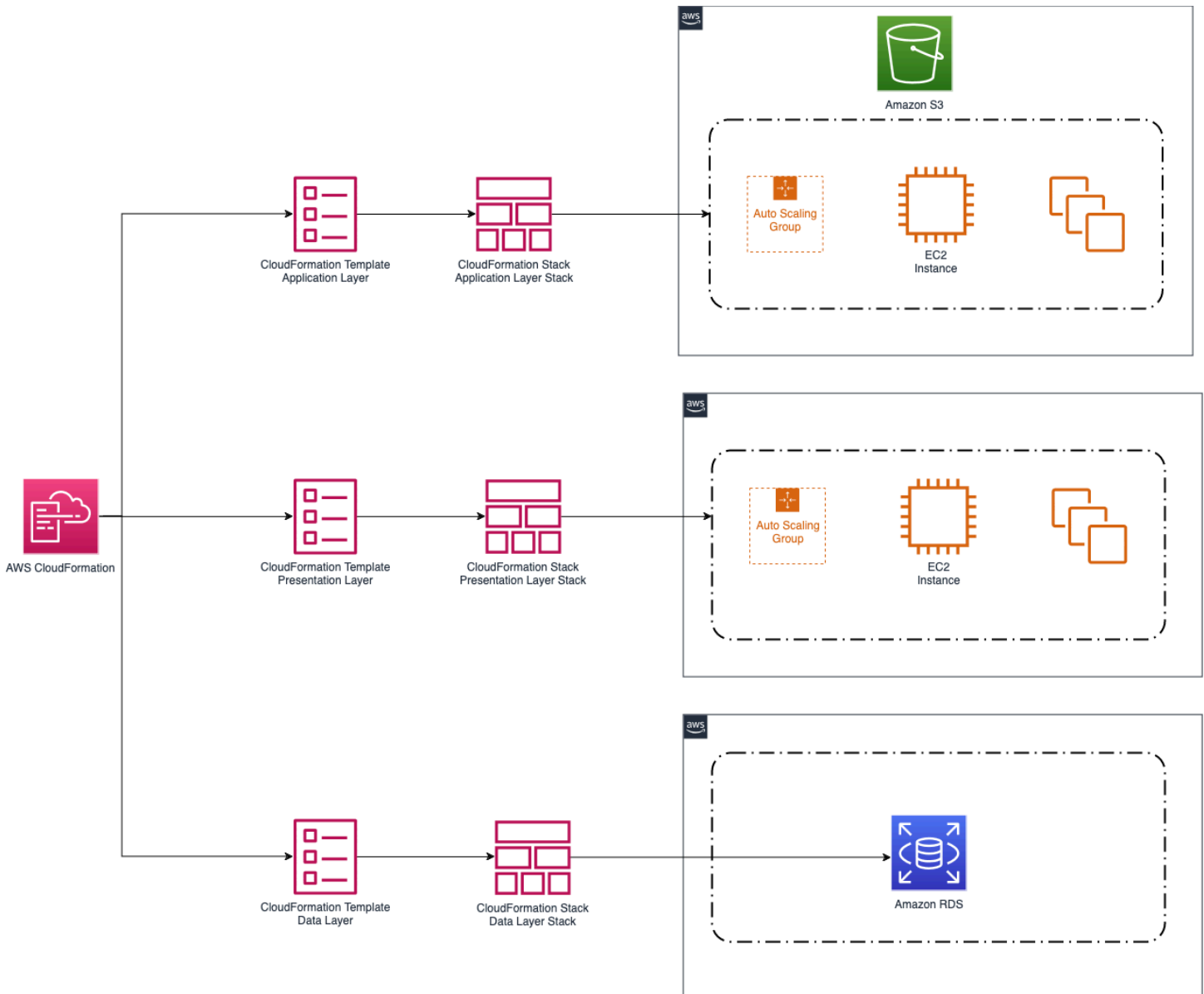
Tableau 1 : fonctionnalités AWS CloudFormation de déploiement

Capacité	Description
Disposition	<p>CloudFormation créera et mettra à jour automatiquement les composants d'infrastructure définis dans un modèle.</p> <p>Reportez-vous à la section <a href="#">AWS CloudFormation Meilleures pratiques</a> pour plus de détails sur la création d'une infrastructure à l'aide CloudFormation de modèles.</p>
Configurer	<p>CloudFormation les modèles offrent une grande flexibilité pour personnaliser et mettre à jour tous les composants de l'infrastructure.</p> <p>Reportez-vous à la section <a href="#">Anatomie des CloudFormation modèles</a> pour plus de détails sur la personnalisation des modèles.</p>
Déploiement	<p>Mettez à jour vos CloudFormation modèles pour modifier les ressources d'une pile. En fonction de l'architecture de votre application, vous aurez peut-être besoin d'un service de déploiement supplémentaire pour mettre à jour la version de l'application exécutée sur votre infrastructure.</p> <p>Reportez-vous à la section <a href="#">Déploiement EC2 d'applications sur Amazon AWS CloudForm</a></p>

Capacité	Description
	<p><a href="#">ation</a> pour plus de détails sur la manière dont elle CloudFormation peut être utilisée comme solution de déploiement.</p>
Échelle	<p>CloudFormation ne gèrera pas automatiquement le dimensionnement de l'infrastructure en votre nom ; toutefois, vous pouvez configurer des politiques de dimensionnement automatique pour vos ressources dans un CloudFormation modèle.</p>
Surveillance	<p>CloudFormation fournit une surveillance native du succès ou de l'échec des mises à jour de l'infrastructure définie dans un modèle, ainsi qu'une détection de dérive pour surveiller les cas où les ressources définies dans un modèle ne répondent pas aux spécifications. Des solutions de surveillance supplémentaires devront être mises en place pour la surveillance et les mesures au niveau des applications.</p> <p>Reportez-vous à la section <a href="#">Surveillance de la progression d'une mise à jour d'une pile</a> pour plus de détails sur la manière dont les mises à jour de l'infrastructure sont CloudFormation surveillées.</p>

Le schéma suivant montre un cas d'utilisation courant pour CloudFormation. Ici, CloudFormation des modèles sont créés pour définir tous les composants d'infrastructure nécessaires à la création d'une application Web simple à trois niveaux. Dans cet exemple, nous utilisons des scripts bootstrap définis dans CloudFormation pour déployer la dernière version de notre application sur des EC2 instances Amazon ; toutefois, il est également courant de combiner des services de déploiement supplémentaires CloudFormation ( CloudFormation uniquement pour la gestion de l'infrastructure et les capacités de provisionnement). Notez que plusieurs CloudFormation modèles sont utilisés pour créer l'infrastructure. Dans le diagramme, CloudFormation il est utilisé pour créer tous les

composants de l'infrastructure, notamment les rôles IAM, les sous-réseaux VPCs, les tables de routage, les groupes de sécurité et les politiques de compartiment Amazon S3. Des CloudFormation modèles distincts sont utilisés pour créer chaque domaine de l'architecture de l'application.



AWS CloudFormation cas d'utilisation

## AWS Elastic Beanstalk

[AWS Elastic Beanstalk](#) est un easy-to-use service de déploiement et de mise à l'échelle d'applications et de services Web développés avec Java, .NET, .NET Core, PHP, Node.js, Python, Ruby, Go ou Docker sur des serveurs courants tels qu'Apache, Nginx, Passenger et IIS. Elastic Beanstalk est

une solution complète de gestion des applications qui gère toutes les tâches d'infrastructure et de plateforme en votre nom.

Avec Elastic Beanstalk, vous pouvez rapidement déployer, gérer et faire évoluer des applications sans la charge opérationnelle liée à la gestion de l'infrastructure. Elastic Beanstalk simplifie la gestion des applications Web, ce qui en fait le choix idéal pour les entreprises qui utilisent AWS pour la première fois ou qui souhaitent déployer une application Web le plus rapidement possible.

Lorsque vous utilisez Elastic Beanstalk comme solution de déploiement, il vous suffit de télécharger votre code source et Elastic Beanstalk fournira et exploitera toute l'infrastructure nécessaire, notamment les serveurs, les bases de données, les équilibreurs de charge, les réseaux et les groupes de mise à l'échelle automatique. Bien que ces ressources soient créées en votre nom, vous conservez le contrôle total de ces ressources, ce qui permet aux développeurs de les personnaliser selon leurs besoins. Elastic Beanstalk répond aux critères de conformité aux normes ISO, PCI, SOC 1, SOC 2 et SOC 3, ainsi qu'aux critères d'éligibilité à la loi HIPAA. Cela signifie que les applications exécutées sur Elastic Beanstalk peuvent traiter des données financières réglementées ou des informations de santé protégées (PHI).

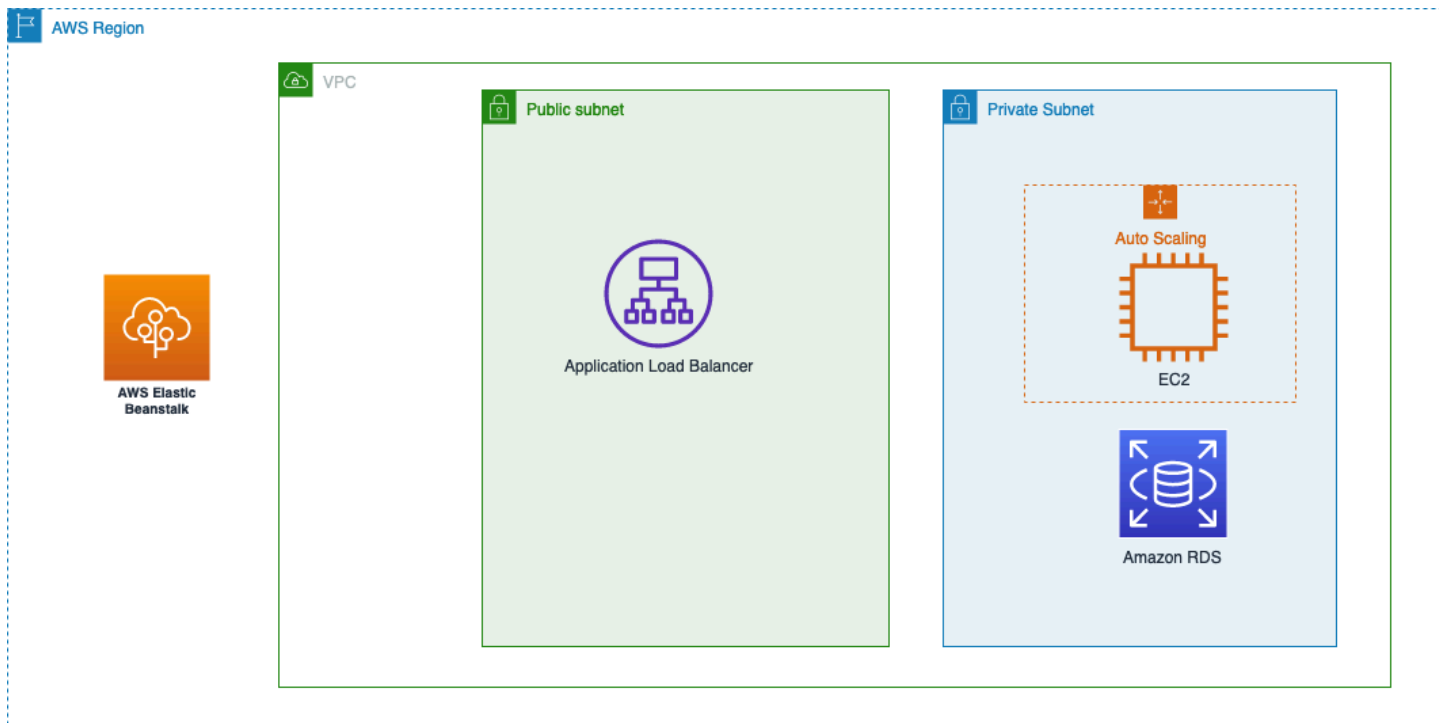
Tableau 2 : Fonctionnalités AWS Elastic Beanstalk de déploiement

Capacité	Description
Disposition	<p>Elastic Beanstalk créera tous les composants d'infrastructure nécessaires au fonctionnement d'une application ou d'un service Web qui s'exécute sur l'une de ses plateformes prises en charge. Si vous avez besoin d'une infrastructure supplémentaire, celle-ci devra être créée en dehors d'Elastic Beanstalk.</p> <p>Reportez-vous à <a href="#">Elastic Beanstalk Platforms</a> pour plus de détails sur les plateformes d'applications Web prises en charge par Elastic Beanstalk.</p>
Configurer	<p>Elastic Beanstalk propose un large éventail d'options pour personnaliser les ressources de votre environnement.</p>

Capacité	Description
Déploiement	<p>Reportez-vous à <a href="#">la section Configuration des environnements Elastic Beanstalk</a> pour plus d'informations sur la personnalisation des ressources créées par Elastic Beanstalk.</p> <p>Elastic Beanstalk gère automatiquement les déploiements d'applications et crée un environnement qui exécute une nouvelle version de votre application sans affecter les utilisateurs existants.</p> <p>Reportez-vous à la section <a href="#">Déploiement d'applications AWS Elastic Beanstalk</a> pour plus de détails sur les déploiements d'applications avec Elastic Beanstalk.</p>
Échelle	<p>Elastic Beanstalk utilise Elastic Load Balancing et Auto Scaling pour dimensionner automatiquement votre application en fonction de ses besoins spécifiques. Les zones de disponibilité multiples vous permettent d'améliorer la fiabilité et la disponibilité des applications.</p> <p>Reportez-vous à <a href="#">Auto Scaling Group pour votre environnement Elastic Beanstalk</a> pour plus de détails sur le dimensionnement automatique avec Elastic Beanstalk.</p>

Capacité	Description
Surveillance	<p>Elastic Beanstalk propose une surveillance intégrée de l'environnement pour les applications, notamment le succès/les échecs des déploiements, l'état de l'environnement, les performances des ressources et les journaux des applications.</p> <p>Reportez-vous à <a href="#">la section Surveillance d'un environnement</a> pour en savoir plus sur la surveillance complète avec Elastic Beanstalk.</p>
Support Graviton	<p>Les processeurs basés sur AWS Graviton arm64 offrent le meilleur rapport prix/performances pour vos charges de travail cloud exécutées sur Amazon. EC2 Avec AWS Graviton on Elastic Beanstalk, vous pouvez sélectionner les types d'instances Amazon pour répondre aux besoins d'optimisation de vos charges de travail et bénéficier d'un meilleur rapport prix/performance par rapport à un processeur x86 comparable.</p>

Elastic Beanstalk facilite le déploiement et la gestion rapides des applications Web dans AWS. L'exemple suivant illustre un cas d'utilisation général d'Elastic Beanstalk utilisé pour déployer une application Web simple. Toute l'infrastructure d'applications (y compris les groupes de sécurité, les rôles IAM et les CloudWatch alarmes) est créée et gérée par Elastic Beanstalk. Les EC2 instances Amazon sont automatiquement approvisionnées avec un environnement d'exécution et des packages de déploiement. Les environnements Elastic Beanstalk peuvent s'intégrer à des ressources telles qu'Amazon Relational Database Service (Amazon RDS) créées en dehors d'Elastic Beanstalk.



## AWS Elastic Beanstalk cas d'utilisation

# AWS CodeDeploy

[AWS CodeDeploy](#) est un service de déploiement entièrement géré qui automatise les déploiements d'applications vers des services informatiques tels qu'Amazon, EC2 [Amazon Elastic Container Service](#) (Amazon ECS) ou des [AWS Lambda](#) serveurs sur site. Organisations peuvent l'utiliser CodeDeploy pour automatiser les déploiements d'une application et supprimer les opérations manuelles sujettes aux erreurs du processus de déploiement. CodeDeploy peut être utilisé avec une grande variété de contenus d'applications, notamment du code, des fonctions sans serveur, des fichiers de configuration, etc.

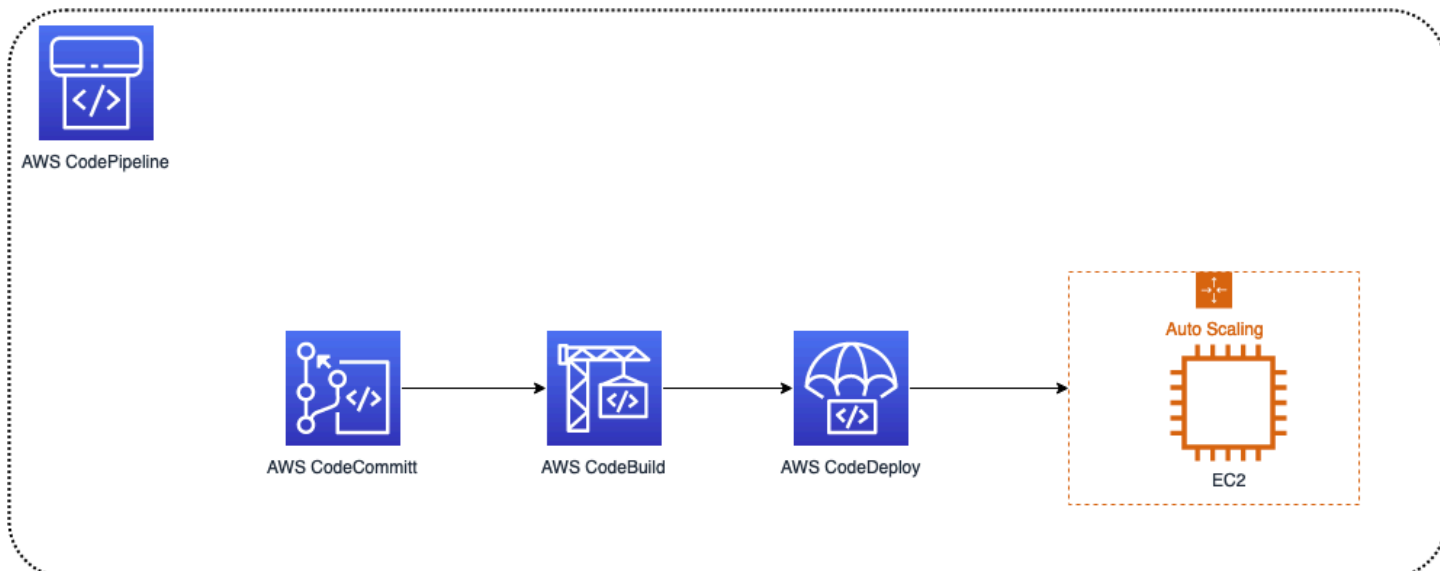
CodeDeploy est destiné à être utilisé comme un service modulaire destiné à aider les développeurs d'applications à déployer et à mettre à jour des logiciels exécutés sur une infrastructure existante. Il ne s'agit pas d'une solution de gestion d'end-to-end applications et est destinée à être utilisée conjointement avec d'autres services de déploiement AWS tels que [AWS CodeStar](#) d'autres [outils de développement AWS](#) et des services tiers (voir [Intégrations de AWS CodeDeploy produits](#) pour une liste complète des intégrations de produits) dans le cadre d'un pipeline complet CI/CD. [AWS CodePipeline](#) En outre, CodeDeploy ne gère pas la création de ressources pour le compte de l'utilisateur.

Tableau 3 : fonctionnalités AWS CodeDeploy de déploiement

Capacité	Description
Disposition	<p>CodeDeploy est destiné à être utilisé avec les ressources informatiques existantes et ne crée pas de ressources en votre nom. CodeDeploy nécessite que les ressources de calcul soient organisées dans une structure appelée groupe de déploiement afin de déployer le contenu de l'application.</p> <p>Reportez-vous à la section <a href="#">Utilisation des groupes de déploiement CodeDeploy</a> pour plus de détails sur la liaison CodeDeploy aux ressources de calcul.</p>
Configurer	<p>CodeDeploy utilise un fichier de spécifications d'application pour définir les personnalisations des ressources de calcul.</p> <p>Reportez-vous à la section <a href="#">CodeDeploy AppSpec File Reference</a> pour plus de détails sur les personnalisations des ressources avec CodeDeploy.</p>
Déploiement	<p>Selon le type de ressource informatique CodeDeploy utilisée, CodeDeploy propose différentes stratégies pour déployer votre application.</p> <p>Reportez-vous à la section <a href="#">Utilisation des déploiements CodeDeploy</a> pour plus de détails sur les types de processus de déploiement pris en charge.</p>
Échelle	<p>CodeDeploy ne prend pas en charge le dimensionnement de votre infrastructure d'applications sous-jacente ; toutefois, en</p>

Capacité	Description
	fonction de vos <a href="#">configurations de déploiement</a> , cela peut créer des ressources supplémentaires pour prendre en charge blue/green les déploiements.
Surveillance	<p>CodeDeploy peut surveiller le succès ou l'échec des déploiements et propose un historique de tous les déploiements, mais ne fournit pas de mesures de performance ou au niveau de l'application.</p> <p>Reportez-vous à <a href="#">la section Surveillance des déploiements CodeDeploy</a> pour plus de détails sur les types de fonctionnalités de surveillance proposés par CodeDeploy</p>

Le schéma suivant illustre un cas d'utilisation général dans CodeDeploy le cadre d'une CI/CD solution complète. Dans cet exemple, CodeDeploy il est utilisé conjointement avec des outils de développement AWS supplémentaires, à savoir AWS CodePipeline (automatiser les CI/CD pipelines), [AWS CodeBuild](#)(créer et tester des composants d'application) et [AWS CodeCommit](#)(référentiel de code source) pour déployer une application sur un groupe d' EC2 instances Amazon. CodeDeploy est utilisé avec d'autres outils dans le cadre d'un CI/CD pipeline complet. CodeDeploy gère le déploiement des composants de l'application sur les ressources informatiques faisant partie d'un groupe de déploiement. Tous les composants de l'infrastructure sont créés à l'extérieur de CodeDeploy.



## AWS CodeDeploy cas d'utilisation

### AWS CodeDeploy pour AWS Lambda

AWS CodeDeploy car vous AWS Lambda permet d'automatiser vos déploiements sans serveur, vous offrant ainsi un meilleur contrôle et une meilleure visibilité sur les versions de vos applications. Vous pouvez l'utiliser CodeDeploy pour déployer une nouvelle version de votre fonction sans serveur pour un faible pourcentage d'utilisateurs ou de trafic et augmenter progressivement le trafic à mesure que vous gagnez en confiance dans la nouvelle version. Avec CodeDeploy, vous pouvez définir des groupes de déploiement, qui représentent un ensemble de fonctions Lambda recevant du trafic provenant de la même source d'événements. Par exemple, vous pouvez créer un groupe de déploiement pour un ensemble de fonctions Lambda initiées par API Gateway ou une règle Amazon EventBridge . Vous pouvez ensuite créer un déploiement à l'aide de CodeDeploy, qui déploie la nouvelle version de votre fonction sans serveur vers un groupe de déploiement spécifié.

CodeDeploy vous permet également de définir une configuration de déploiement, qui spécifie les paramètres d'un déploiement, tels que le type de déploiement, la stratégie de déploiement et les règles de transfert du trafic. Vous pouvez utiliser la stratégie de déploiement de Canary pour déployer la nouvelle version de votre fonction sans serveur sur un faible pourcentage du trafic et surveiller l'état et les performances de la nouvelle version avant d'augmenter le trafic vers celle-ci.

En utilisant CodeDeploy for serverless, vous pouvez automatiser votre processus de déploiement, réduire le temps et les efforts nécessaires pour publier de nouvelles versions de votre application et améliorer la stabilité et la fiabilité de vos fonctions sans serveur.

# Amazon Elastic Container Service

Amazon Elastic Container Service (Amazon ECS) est un service d'orchestration de conteneurs entièrement géré qui prend en charge les conteneurs Docker et vous permet d'exécuter facilement des applications sur un cluster géré. Amazon ECS élimine le besoin d'installer, d'exploiter et de dimensionner l'infrastructure de gestion des conteneurs, et simplifie la création d'environnements dotés de fonctionnalités AWS courantes, telles que [Security Groups](#), [Elastic Load Balancing](#) et [Gestion des identités et des accès AWS \(IAM\)](#).

Lorsque vous exécutez des applications sur Amazon ECS, vous pouvez choisir de fournir la puissance de calcul sous-jacente à vos conteneurs avec des EC2 instances Amazon ou avec [AWS Fargate](#) un moteur de calcul sans serveur pour les conteneurs. Dans les deux cas, Amazon ECS place et fait évoluer automatiquement vos conteneurs sur votre cluster conformément aux configurations définies par l'utilisateur. Bien qu'Amazon ECS ne crée pas de composants d'infrastructure tels que des équilibrateurs de charge ou des rôles IAM en votre nom, le service Amazon ECS en fournit un certain nombre APIs pour simplifier la création et l'utilisation de ces ressources dans un cluster Amazon ECS.

Amazon ECS permet aux développeurs d'avoir un contrôle direct et précis sur tous les composants de l'infrastructure, ce qui permet de créer des architectures d'applications personnalisées. En outre, Amazon ECS prend en charge différentes stratégies de déploiement pour mettre à jour les images de vos conteneurs d'applications.

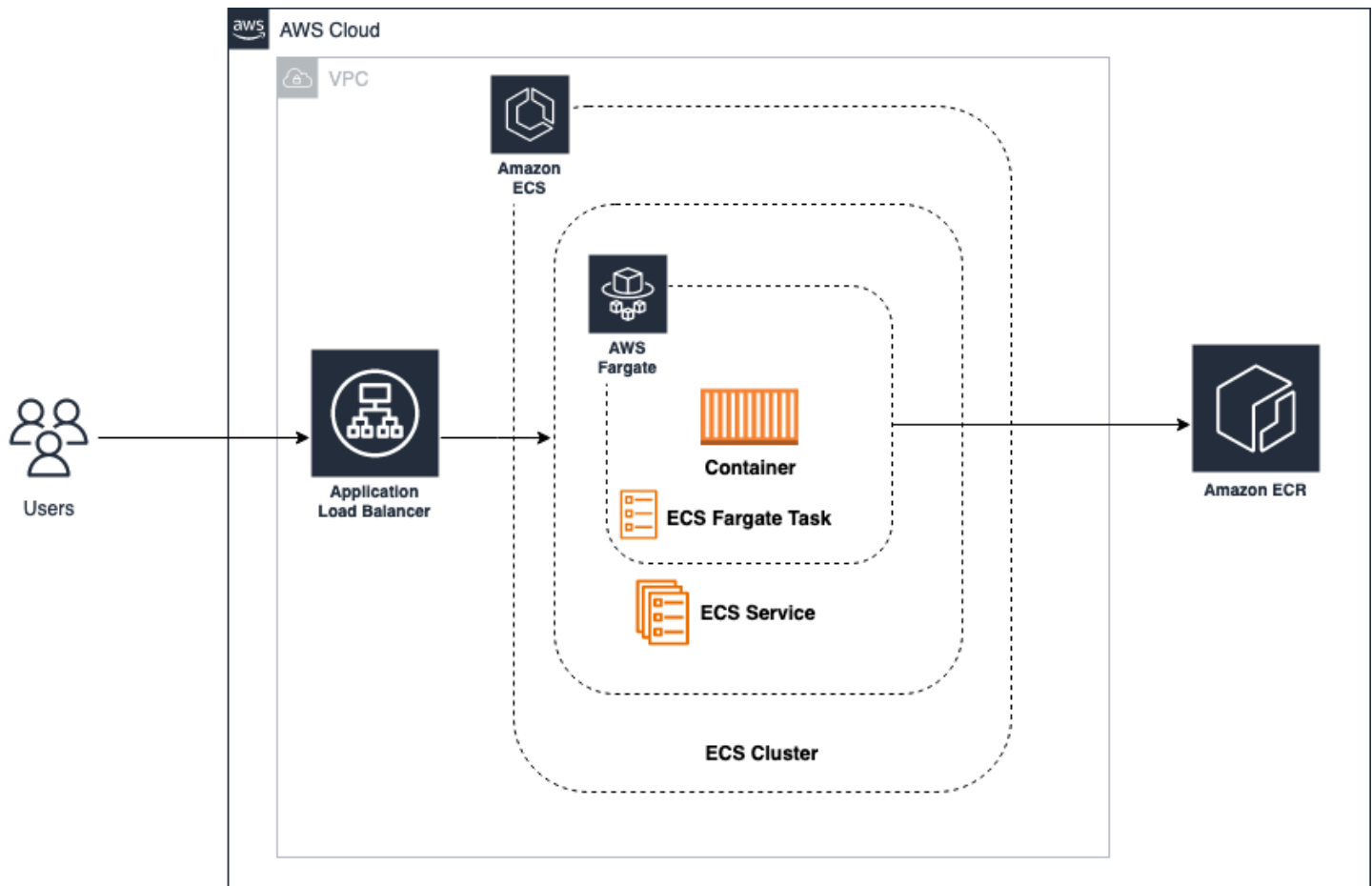
Tableau 4 : Fonctionnalités de déploiement d'Amazon ECS

Capacité	Description
Disposition	<p>Amazon ECS fournira de nouvelles instances de conteneurs d'applications et de nouvelles ressources de calcul en fonction des politiques de dimensionnement et des configurations Amazon ECS. Les ressources d'infrastructure telles que les équilibrateurs de charge devront être créées en dehors d'Amazon ECS.</p> <p>Reportez-vous à <a href="#">Getting Started with Amazon ECS</a> pour plus de détails sur les types de</p>

Capacité	Description
	<p>ressources qui peuvent être créés avec Amazon ECS.</p>
Configurer	<p>Amazon ECS prend en charge la personnalisation des ressources de calcul créées pour exécuter une application conteneurisée, ainsi que des conditions d'exécution des conteneurs d'applications (par exemple, les variables d'environnement, les ports exposés, la mémoire/le processeur réservés). La personnalisation des ressources de calcul sous-jacentes n'est disponible que si vous utilisez EC2 des instances Amazon.</p> <p>Reportez-vous à <a href="#">la section Création d'un cluster</a> pour plus de détails sur la façon de personnaliser un cluster Amazon ECS pour exécuter des applications conteneurisées.</p>
Déploiement	<p>Amazon ECS prend en charge plusieurs stratégies de déploiement pour vos applications conteneurisées.</p> <p>Reportez-vous à la section <a href="#">Types de déploiement Amazon ECS</a> pour plus de détails sur les types de processus de déploiement pris en charge.</p>

Capacité	Description
Échelle	<p>Amazon ECS peut être utilisé avec des politiques de dimensionnement automatique pour ajuster automatiquement le nombre de conteneurs exécutés dans votre cluster Amazon ECS.</p> <p>Reportez-vous à <a href="#">Service Auto Scaling</a> pour plus de détails sur la configuration du dimensionnement automatique pour vos applications conteneurisées sur Amazon ECS.</p>
Surveillance	<p>Amazon ECS prend en charge la surveillance des ressources informatiques et des conteneurs d'applications avec CloudWatch.</p> <p>Reportez-vous à la section <a href="#">Surveillance d'Amazon ECS</a> pour plus de détails sur les types de fonctionnalités de surveillance proposés par Amazon ECS.</p>

Le schéma suivant illustre l'utilisation d'Amazon ECS pour gérer une application conteneurisée simple. Dans cet exemple, les composants d'infrastructure sont créés en dehors d'Amazon ECS, et Amazon ECS est utilisé pour gérer le déploiement et le fonctionnement des conteneurs d'applications sur le cluster



## Cas d'utilisation d'Amazon ECS

### Note

- L'infrastructure des applications (y compris les référentiels Amazon Elastic Container Registry (Amazon ECR), les configurations Amazon ECS et les équilibreurs de charge) est mise en service et gérée en dehors de votre déploiement Amazon ECS.
- Amazon ECS gère le déploiement de conteneurs d'applications exécutés au sein du service Amazon ECS sous forme de tâches provenant d'un registre de conteneurs tel qu'Amazon ECR.

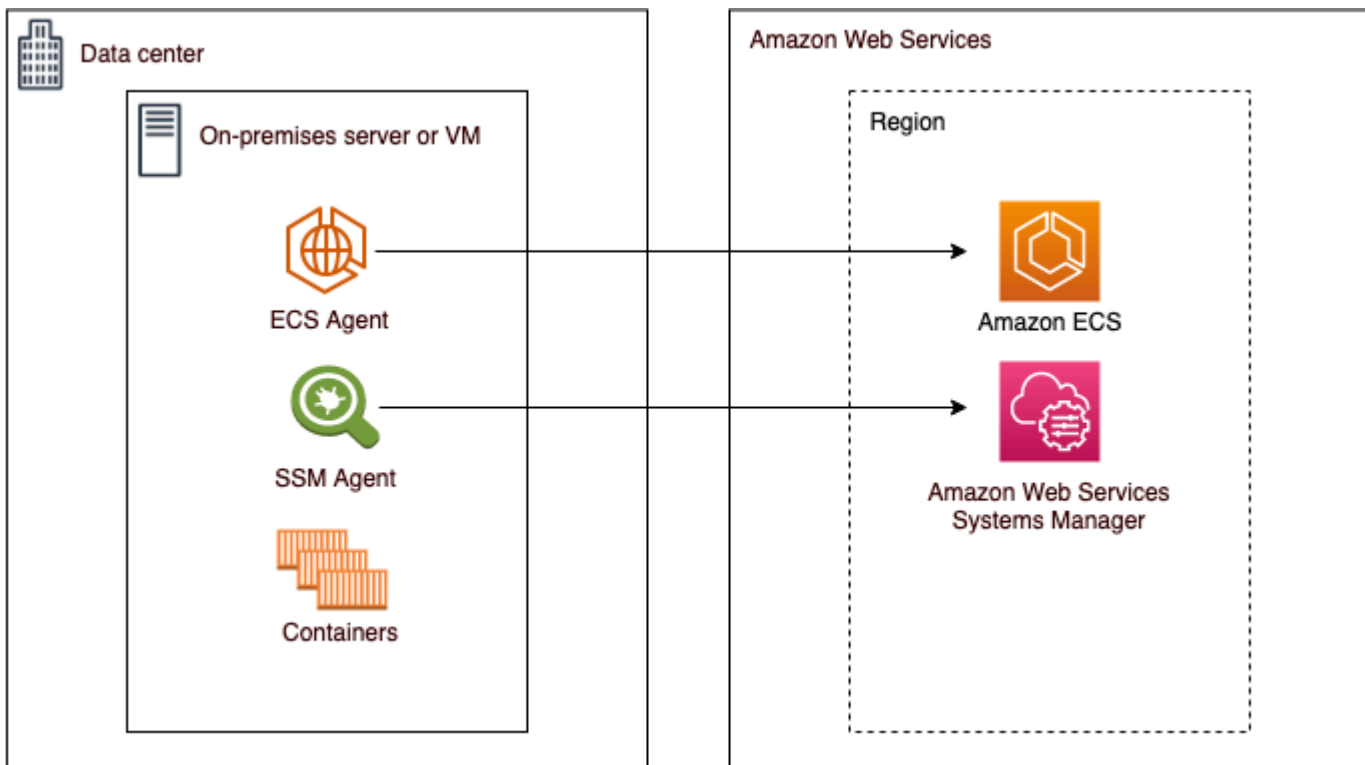
Amazon ECS prend en charge plusieurs types d'instances de conteneur tels que Linux et Windows, ainsi que des types d'instances externes tels qu'une machine virtuelle (VM) sur site avec Amazon ECS Anywhere.

## Amazon ECS Anywhere

[Amazon ECS Anywhere](#) vous permet d'exécuter des tâches Amazon ECS n'importe où, que ce soit sur site ou dans d'autres environnements cloud. Avec Amazon ECS Anywhere, vous pouvez facilement déployer et gérer des applications conteneurisées au sein de votre infrastructure hybride, tout en garantissant une expérience opérationnelle cohérente. Le service fonctionne en étendant la plateforme Amazon ECS à n'importe quel environnement, y compris les centres de données sur site, les bureaux distants et les autres environnements cloud. Il vous permet d'utiliser le même Amazon ECS APIs et les mêmes outils habituels pour déployer et gérer des conteneurs dans tous vos environnements, sans avoir à vous soucier de l'infrastructure sous-jacente.

Amazon ECS Anywhere utilise l'agent Amazon ECS pour gérer le déploiement et le cycle de vie des conteneurs, ce qui vous permet d'utiliser les mêmes définitions de tâches et fichiers de configuration Amazon ECS que ceux que vous utilisez dans le AWS Cloud. Cela peut contribuer à simplifier le processus de déploiement et de gestion des conteneurs dans votre infrastructure hybride et à réduire le temps et les efforts nécessaires à la configuration et à la gestion manuelles.

Avec Amazon ECS Anywhere, vous pouvez également tirer parti d'autres services AWS, tels que IAM et Amazon ECR CloudFormation, pour gérer vos applications conteneurisées. Cela peut contribuer à garantir la sécurité, la conformité et l'intégration de vos applications aux autres services AWS.



Amazon ECS Anywhere architecture

## Amazon Elastic Container Service sur AWS Outposts

[Amazon ECS on AWS Outposts](#) est un service AWS entièrement géré qui vous permet d'exécuter des tâches Amazon ECS sur site, en utilisant les mêmes APIs outils que ceux que vous utilisez dans le AWS Cloud. Avec Amazon ECS activé AWS Outposts, vous pouvez déployer et gérer des applications conteneurisées de manière cohérente et familière, que vous les exécutiez sur site ou dans le cloud. AWS Outposts est un service entièrement géré qui étend l'infrastructure, les services et les outils AWS à vos environnements sur site. Avec Amazon ECS activé AWS Outposts, vous pouvez exécuter des tâches Amazon ECS sur du matériel dédié à votre organisation, sans avoir à vous soucier de l'infrastructure sous-jacente. Cela peut vous aider à garantir que vos applications sont déployées de manière sécurisée et conforme, tout en vous permettant de tirer parti de la flexibilité et de l'évolutivité du cloud.

Amazon ECS on AWS Outposts fonctionne en déployant un ensemble de services AWS dans votre environnement sur site, ce qui vous permet d'exécuter des tâches Amazon ECS sur du matériel dédié. Cela inclut l'agent Amazon ECS, qui gère le déploiement et le cycle de vie des conteneurs, et l'AWS Outposts infrastructure, qui fournit un environnement sécurisé et conforme pour exécuter des applications conteneurisées. Avec Amazon ECS activé AWS Outposts, vous pouvez utiliser le même Amazon ECS APIs et les mêmes outils que ceux que vous utilisez dans le AWS Cloud, ce qui facilite le déploiement et la gestion des applications conteneurisées de manière cohérente et familière. Cela peut contribuer à réduire le temps et les efforts nécessaires à la configuration et à la gestion manuelles, et à améliorer la cohérence et la fiabilité de votre infrastructure hybride. Amazon ECS on AWS Outposts s'intègre également à d'autres services AWS, tels que IAM et Amazon ECR CloudFormation, pour gérer vos applications conteneurisées. Cela peut contribuer à garantir la sécurité, la conformité et l'intégration de vos applications aux autres services AWS.

## Amazon Elastic Kubernetes Service

[Amazon Elastic Kubernetes Service](#) (Amazon EKS) est un service entièrement géré et certifié conforme à Kubernetes qui simplifie le processus de création, de sécurisation, d'exploitation et de maintenance des clusters Kubernetes sur AWS. Amazon EKS s'intègre aux principaux services AWS tels que CloudWatch Auto Scaling Groups et IAM afin de fournir une expérience fluide de surveillance, de dimensionnement et d'équilibrage de charge de vos applications conteneurisées.

Amazon EKS fournit un plan de contrôle évolutif et hautement disponible pour les charges de travail Kubernetes. Lorsque vous exécutez des applications sur Amazon EKS, comme avec Amazon ECS,

vous pouvez choisir de fournir la puissance de calcul sous-jacente à vos conteneurs avec des EC2 instances Amazon ou avec AWS Fargate.

Amazon VPC Lattice est un service de mise en réseau d'applications entièrement géré intégré directement à l'infrastructure réseau AWS que vous pouvez utiliser pour connecter, sécuriser et surveiller vos services sur plusieurs comptes et clouds privés virtuels (). VPCs Avec Amazon EKS, vous pouvez tirer parti de VPC Lattice en utilisant le contrôleur d'API AWS Gateway, une implémentation de l'API Kubernetes Gateway. À l'aide de VPC Lattice, vous pouvez configurer la connectivité entre clusters avec la sémantique standard de Kubernetes de manière simple et cohérente.

Vous pouvez utiliser Amazon EKS avec l'une des options de déploiement suivantes :

- [Amazon EKS Distro](#) – Amazon EKS Distro est une distribution des mêmes logiciels et dépendances Kubernetes open source déployés par Amazon EKS dans le cloud. Amazon EKS Distro suit le même cycle de publication de version Kubernetes qu'Amazon EKS et est fourni en tant que projet open source. Pour en savoir plus, consultez [Amazon EKS Distro](#).
- [Amazon EKS activé AWS Outposts](#) : AWS Outposts active les services, l'infrastructure et les modèles d'exploitation AWS natifs dans vos installations sur site. Amazon EKS activé AWS Outposts, vous pouvez choisir d'exécuter des clusters étendus ou locaux. Avec les clusters étendus, le plan de contrôle Kubernetes s'exécute dans un Région AWS et les nœuds s'exécutent sur. AWS Outposts Avec les clusters locaux, l'ensemble du cluster Kubernetes s'exécute localement AWS Outposts, y compris le plan de contrôle Kubernetes et les nœuds.
- [Amazon EKS Anywhere](#) – Amazon EKS Anywhere est une option de déploiement pour Amazon EKS qui vous permet de créer et d'exploiter facilement des clusters Kubernetes sur site. Amazon EKS et Amazon EKS Anywhere sont tous deux basés sur Amazon EKS Distro. Pour en savoir plus sur Amazon EKS Anywhere, consultez les sections [Exécution de charges de travail de conteneurs hybrides avec Amazon EKS Anywhere](#), [présentation d'Amazon EKS Anywhere](#) et [Comparaison entre Amazon EKS Anywhere et Amazon EKS](#).

Lorsque vous choisissez les options de déploiement à utiliser pour votre cluster Kubernetes, tenez compte des éléments suivants :

Tableau 5 : Fonctionnalités de déploiement de Kubernetes

Fonctionnalité	Amazon EKS	Amazon EKS sur AWS Outposts	Amazon EKS Anywhere	Amazon EKS Distro
Matériel	Fourni par AWS	Fourni par AWS	Fourni par vous	Fourni par vous
Emplacement de déploiement	AWS Cloud	Votre centre de données	Votre centre de données	Votre centre de données
Emplacement du plan de contrôle Kubernetes	AWS Cloud	AWS Cloud ou votre centre de données	Votre centre de données	Votre centre de données
Emplacement du plan de données Kubernetes	AWS Cloud	Votre centre de données	Votre centre de données	Votre centre de données
Support	AWS soutien	AWS soutien	AWS soutien	Support de la communauté OSS

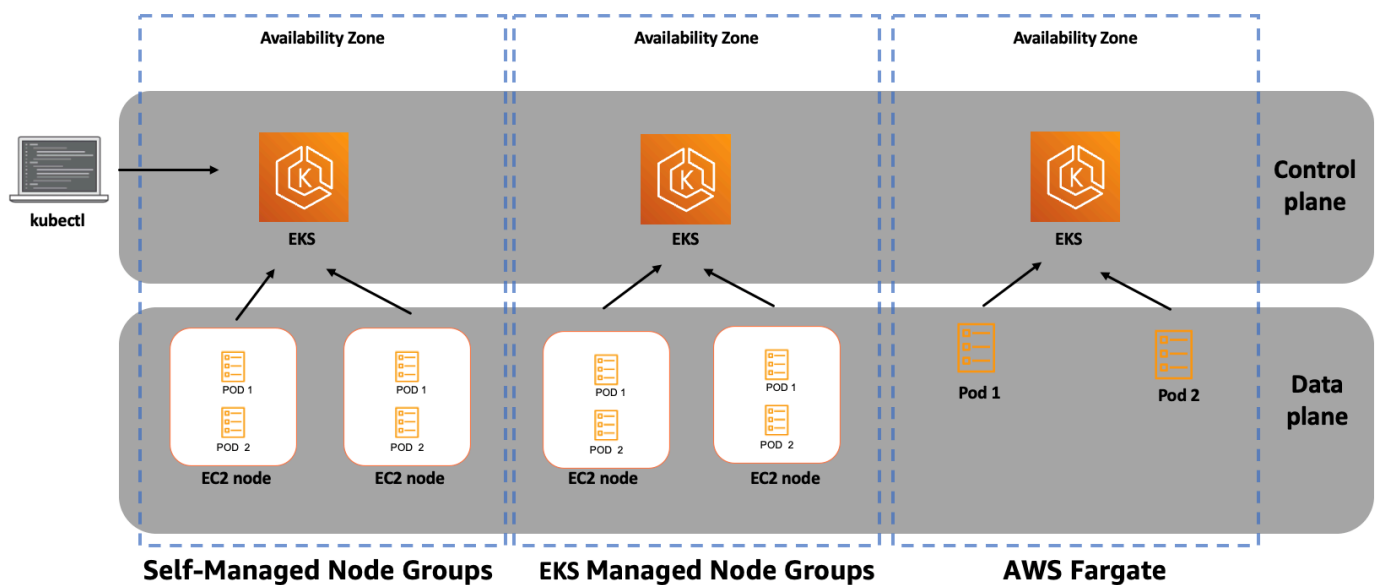
Tableau 6 : Fonctionnalités de déploiement d'Amazon EKS

Capacité	Description
Disposition	<p>Amazon EKS fournit certaines ressources pour prendre en charge les applications conteneur isées :</p> <ul style="list-style-type: none"> <li>• Équilibres de charge, si nécessaire</li> <li>• Ressources informatiques ou travailleurs (Amazon EKS prend en charge Windows et Linux)</li> <li>• Instances de conteneurs d'applications, ou pods</li> </ul>

Capacité	Description
	Reportez-vous à <a href="#">Getting Started with Amazon EKS</a> pour plus de détails sur le provisionnement de clusters Amazon EKS.
Configurer	<p>Amazon EKS prend en charge la personnalisation des ressources de calcul (travailleurs) si vous utilisez des EC2 instances Amazon pour fournir de la puissance de calcul. Amazon EKS prend également en charge la personnalisation des conditions d'exécution des conteneurs d'applications (pods).</p> <p>Reportez-vous à la <a href="#">documentation de configuration des nœuds</a> de travail et <a href="#">de Fargate Pod pour</a> plus de détails.</p>
Déploiement	Amazon EKS prend en charge les mêmes stratégies de déploiement que Kubernetes. Voir <a href="#">Rédaction d'une spécification de déploiement Kubernetes -&gt; Stratégie</a> pour plus de détails.
Échelle	Amazon EKS fait évoluer les collaborateurs avec <a href="#">Kubernetes Cluster Autoscaler</a> , et les pods avec <a href="#">Kubernetes Horizontal Pod Autoscaler</a> et Kubernetes Vertical Pod Autoscaler. Amazon EKS prend également en charge <a href="#">Karpenter</a> , un autoscaleur de clusters Kubernetes open source, flexible et performant, qui permet d'améliorer la disponibilité de vos applications et l'efficacité de votre cluster en lançant rapidement des ressources de calcul adaptées à l'évolution de la charge des applications.

Capacité	Description
Surveillance	<p>Les journaux du plan de contrôle Amazon EKS fournissent des informations d'audit et de diagnostic directement aux CloudWatch journaux. Le plan de contrôle Amazon EKS s'intègre également AWS CloudTrail à Amazon EKS pour enregistrer les actions effectuées dans Amazon EKS.</p> <p>Reportez-vous à la section <a href="#">Logging and Monitoring Amazon EKS</a> pour plus de détails.</p>

Amazon EKS permet aux entreprises de tirer parti des outils et plug-ins Kubernetes open source, et peut constituer un bon choix pour les organisations qui migrent vers AWS avec des environnements Kubernetes existants. Le schéma suivant illustre l'utilisation d'Amazon EKS pour gérer une application conteneurisée générale.



Amazon EKS use case

## Amazon EKS Anywhere

[Amazon EKS Anywhere](#) vous permet de créer et d'exploiter des clusters Kubernetes sur votre propre infrastructure. Amazon EKS Anywhere s'appuie sur les points forts d'Amazon EKS Distro et fournit des logiciels open source à jour et corrigés afin que vous puissiez disposer d'un environnement Kubernetes sur site plus fiable qu'une offre Kubernetes autogérée.

Amazon EKS Anywhere crée un cluster Kubernetes sur site pour le fournisseur choisi. Les fournisseurs pris en charge incluent Bare Metal (via Tinkerbell) et CloudStack vSphere. Pour gérer ce cluster, vous pouvez exécuter des commandes de création et de suppression de clusters à partir d'un ordinateur d'administration Ubuntu ou Mac.

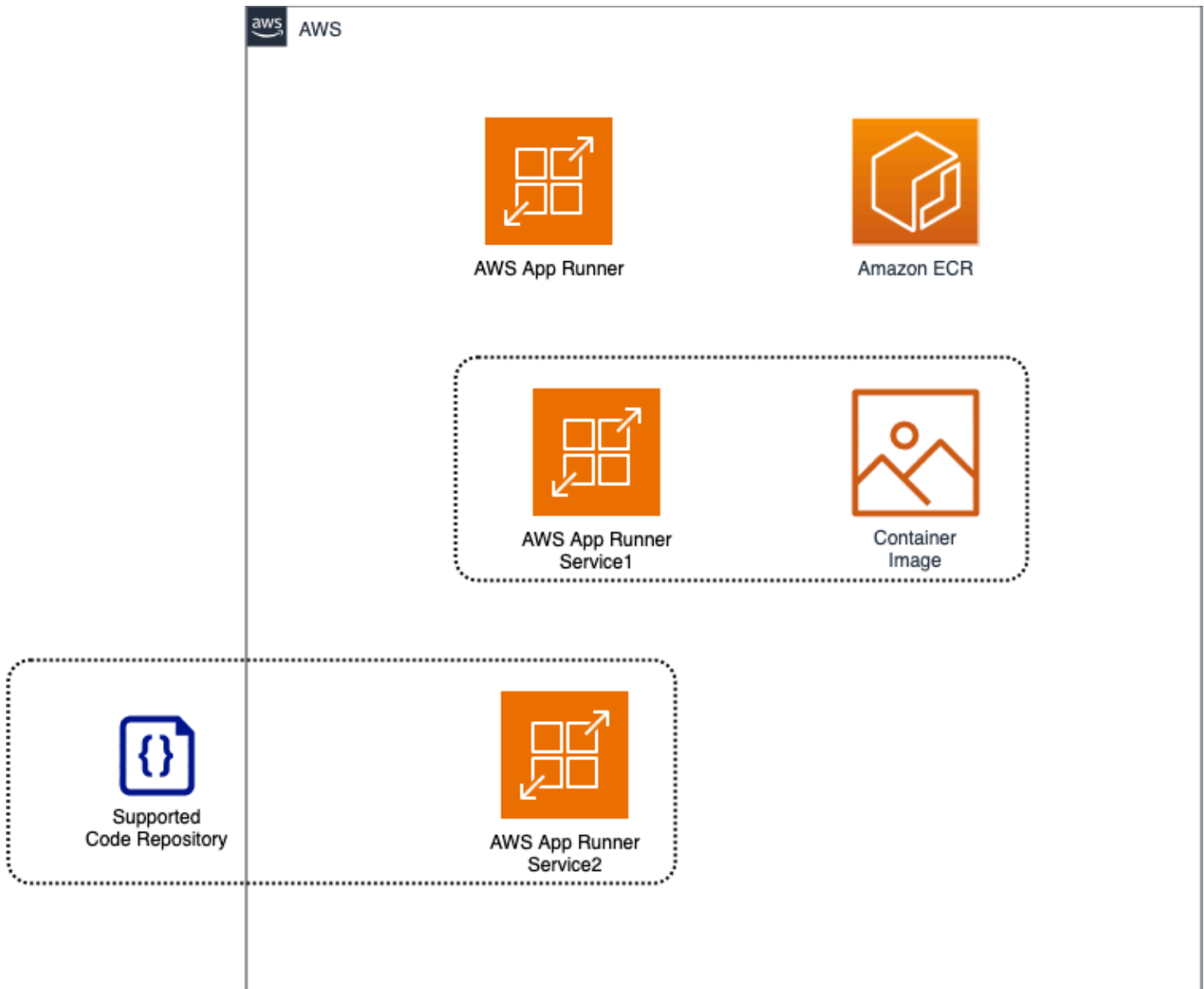
## AWS App Runner

[AWS App Runner](#) est un service d'applications de conteneur entièrement géré qui vous permet de créer, de déployer et d'exécuter des applications Web conteneurisées et des services d'API sans expérience préalable en matière d'infrastructure ou de conteneur. App Runner se connecte directement à votre référentiel de code ou d'images. Il fournit un pipeline d'intégration et de livraison automatique avec des opérations entièrement gérées, des performances, une évolutivité et une sécurité élevées.

App Runner extrait votre code source ou votre image source d'un référentiel, puis crée et gère un service Web en cours d'exécution pour vous dans le AWS Cloud. Généralement, vous n'avez besoin d'appeler qu'une seule action App Runner pour créer votre service. `CreateService` Avec un référentiel d'images source, vous fournissez une image de ready-to-use conteneur qu'App Runner peut déployer pour exécuter votre service Web. Avec un référentiel de code source, vous fournissez votre code et les instructions pour créer et exécuter un service Web et vous ciblez un environnement d'exécution spécifique. App Runner prend en charge plusieurs plateformes de programmation, chacune avec un ou plusieurs environnements d'exécution gérés pour les versions majeures de la plateforme. App Runner prend en charge les images de conteneur ainsi que les environnements d'exécution et les frameworks Web tels que Node.js et Python. App Runner surveille le nombre de demandes simultanées envoyées à votre application et ajoute automatiquement des instances supplémentaires en fonction du volume de demandes. Si votre application ne reçoit aucune demande entrante, App Runner réduira les conteneurs à une instance provisionnée, une instance régulée par le processeur prête à traiter les demandes entrantes en quelques millisecondes.

À l'heure actuelle, App Runner peut récupérer votre code source depuis un GitHub référentiel ou récupérer votre image source depuis Amazon ECR dans votre Compte AWS.

Le schéma suivant présente une vue d'ensemble de l'architecture du service App Runner. Le schéma présente deux exemples de services : l'un déploie le code source depuis GitHub Amazon ECR et l'autre déploie une image source depuis Amazon ECR.



## App Runner use case

App Runner prend en charge le développement complet, y compris les applications Web frontales et dorsales qui utilisent les protocoles HTTP et HTTPS. Ces applications incluent les services d'API, les services Web principaux et les sites Web. App Runner prend en charge les images de conteneur ainsi que les environnements d'exécution et les frameworks Web tels que Node.js et Python.

# Amazon Lightsail

[Amazon Lightsail](#) est un service cloud simple et rentable qui permet aux petites entreprises, aux start-ups et aux particuliers de déployer et de gérer facilement leurs applications dans le cloud. Il fournit une interface conviviale qui fait abstraction d'une grande partie de la gestion de l'infrastructure sous-jacente et facilite le lancement et l'exécution d'applications dans le cloud. Avec Lightsail, vous pouvez rapidement déployer et gérer des serveurs privés virtuels (VPS), des bases de données et des instances de stockage. Le service fournit des instances préconfigurées optimisées pour diverses charges de travail, telles que WordPress, Drupal et Joomla, entre autres. Cela peut contribuer à réduire le temps et les efforts nécessaires à l'installation et à la configuration de votre environnement. Lightsail fournit également un équilibreur de charge intégré et une mise à l'échelle automatique, vous permettant de gérer l'évolution de la demande de trafic sans intervention manuelle. Le service fournit également une surveillance et des alertes, afin que vous puissiez rester au fait de l'état et des performances de vos applications.

L'un des principaux avantages de Lightsail est sa simplicité et sa facilité d'utilisation. Le service est conçu pour être accessible aux utilisateurs ayant une expérience minimale du cloud computing, ce qui en fait une bonne option pour les petites entreprises ou les particuliers qui souhaitent se lancer rapidement dans le cloud. En outre, Lightsail est rentable, avec des prix prévisibles qui incluent le calcul, le stockage et le transfert de données.

## Conteneurs Amazon Lightsail

Amazon Lightsail Containers est un service de conteneur entièrement géré AWS qui facilite le déploiement et la gestion d'applications conteneurisées dans le cloud. Il fournit un moyen simple et économique de lancer et d'exécuter des conteneurs à l'aide d'outils de gestion de conteneurs populaires, tels que Docker et Kubernetes.

Lightsail Containers fournit un environnement intégré pour créer, tester et déployer des applications conteneurisées. Il simplifie le processus de déploiement et de gestion des conteneurs en fournissant une interface conviviale qui fait abstraction d'une grande partie de la gestion de l'infrastructure sous-jacente.

Avec Lightsail Containers, vous pouvez déployer vos applications conteneurisées sur un VPC en quelques clics. Le service fournit des images de conteneur préconfigurées pour les langages de programmation courants, tels que Node.js, Python, Ruby et Java. Cela peut contribuer à réduire le temps et les efforts nécessaires à la mise en place et à la configuration de votre environnement de conteneurs.

Lightsail Containers fournit également un équilibreur de charge intégré capable de répartir automatiquement le trafic entre vos instances de conteneur, améliorant ainsi la disponibilité et l'évolutivité des applications. En outre, le service fournit une mise à l'échelle automatique des instances de conteneur, ce qui vous permet de gérer l'évolution de la demande de trafic sans intervention manuelle.

Avec Lightsail Containers, vous pouvez surveiller les performances de vos applications conteneurisées à l'aide de métriques et de journaux intégrés. Vous pouvez également intégrer d'autres services AWS, tels qu'Amazon S3, Amazon RDS AWS CodePipeline, et créer un CI/CD pipeline entièrement automatisé et intégré pour vos applications conteneurisées.

## Red Hat OpenShift Service on AWS

[Red Hat OpenShift Service on AWS](#)(ROSA) est un service géré disponible via l'AWS Management Console. Avec ROSA, en tant qu' OpenShift utilisateur de Red Hat, vous pouvez créer, dimensionner et gérer des applications conteneurisées sur AWS. Vous pouvez utiliser ROSA pour créer des clusters Kubernetes à l'aide de Red Hat OpenShift APIs et d'outils, et avoir accès à l'ensemble des services AWS. ROSA rationalise le transfert des charges de OpenShift travail Red Hat sur site vers AWS et offre une intégration étroite avec les autres services AWS. Vous pouvez également accéder aux OpenShift licences, à la facturation et au support Red Hat directement via AWS.

Chaque cluster ROSA est fourni avec un plan de contrôle entièrement géré et des nœuds de calcul. L'installation, la gestion, la maintenance et les mises à niveau sont effectuées par Red Hat SRE avec le support conjoint de Red Hat et d'Amazon. Les services de cluster (tels que la journalisation, les métriques et la surveillance) sont également disponibles. Seuls les utilisateurs de Red Hat Enterprise Linux CoreOS (RHCOS) sont pris en charge par ROSA.

ROSA s'intégrera à une gamme de services AWS de calcul, de stockage, de base de données, d'analyse, d'apprentissage automatique, de mise en réseau, de téléphonie mobile et à divers services d'applications, ce qui permettra aux clients de bénéficier du solide portefeuille de services AWS évolutifs à la demande dans le monde entier. Ces services AWS natifs seront directement accessibles pour déployer et faire évoluer rapidement les services via la même interface de gestion.

## Zones locales AWS

Une [zone locale AWS](#) est une extension d'une Région AWS zone géographique proche de vos utilisateurs. Les Local Zones ont leurs propres connexions à Internet et prennent en charge AWS Direct Connect. Les ressources créées dans une zone locale peuvent servir les utilisateurs locaux

avec des communications à faible latence. Une zone locale est représentée par un code de région suivi d'un identifiant indiquant l'emplacement (par exemple, us-west-2-lax-1a).

Amazon ECS prend en charge les charges de travail qui utilisent des zones locales lorsqu'une faible latence ou un traitement local des données sont requis. Le plan de contrôle Amazon ECS s'exécutera toujours dans le Région AWS.

Amazon EKS prend en charge certaines ressources dans Local Zones. Cela inclut les [EC2 nœuds Amazon autogérés](#), les volumes Amazon EBS et les équilibrateurs de charge d'applications. Le plan de contrôle Kubernetes géré par Amazon EKS s'exécute toujours dans la Région AWS. Le plan de contrôle Kubernetes géré par Amazon EKS ne peut pas s'exécuter dans la Local Zone. Étant donné que les Local Zones apparaissent comme un sous-réseau au sein de votre VPC, Kubernetes voit vos ressources de zone locale comme faisant partie de ce sous-réseau.

## AWS Wavelength

[AWS Wavelength](#) est une infrastructure AWS qui vous permet de déployer des charges de travail à proximité d'utilisateurs et d'appareils connectés à la 5G. Vous pouvez utiliser Wavelength pour déployer EC2 des instances Amazon, des clusters Amazon EKS et une suite de solutions partenaires prises en charge disponibles sur AWS Marketplace. Les zones de longueur d'onde sont des centres de données isolés logiquement au sein des réseaux des fournisseurs de télécommunications qui sont reconnectés à la région AWS par le biais d'une connectivité redondante, à faible latence et à haut débit.

Parmi les principales fonctionnalités de Wavelength, citons la possibilité de créer des EC2 instances Amazon, des volumes Amazon EBS, des sous-réseaux Amazon VPC et des passerelles de transport dans les zones Wavelength. Vous pouvez également utiliser des services qui orchestrent ou fonctionnent avec Amazon EC2, Amazon EBS et Amazon VPC, tels qu'Amazon EC2 Auto Scaling, les clusters Amazon EKS, les clusters Amazon ECS, Amazon Systems Manager, Amazon CloudWatch, AWS CloudTrail, AWS CloudFormation, Amazon et Application Load Balancer. Les services Wavelength font partie d'un VPC connecté via une connexion fiable à haut débit à une région AWS afin d'accéder facilement à des services tels qu'Amazon DynamoDB et Amazon Relational Database Service (Amazon RDS).

## Services de déploiement supplémentaires

[Amazon Simple Storage Service](#) (Amazon S3) peut être utilisé comme serveur Web pour le contenu statique et les applications d'une seule page (SPA). Associée CloudFront à Amazon pour améliorer

les performances en matière de diffusion de contenu statique, l'utilisation d'Amazon S3 peut être un moyen simple et puissant de déployer et de mettre à jour du contenu statique. Vous trouverez plus de détails sur cette approche dans le AWS livre blanc [sur l'hébergement de sites Web statiques](#).

## AWS Proton

[AWS Proton](#) est un service entièrement géré qui simplifie et automatise le processus de déploiement et de gestion des microservices et des applications basées sur des conteneurs. Il fournit une expérience de déploiement unifiée et cohérente qui s'intègre aux DevOps outils et services courants, facilitant ainsi la gestion et la rationalisation du développement des applications. Proton permet aux développeurs de définir et de créer des composants d'application, tels que l'infrastructure, le code et les pipelines, sous forme de modèles réutilisables. Ces modèles peuvent être utilisés pour créer plusieurs environnements, tels que le développement, les tests et la production, et peuvent être partagés entre des équipes ou des organisations. Cette approche permet de réduire la complexité du déploiement et de la gestion des microservices et des applications basées sur des conteneurs, qui peuvent être chronophages et source d'erreurs.

AWS Proton fournit des modèles prédéfinis pour les types courants de microservices, tels que les applications Web et les bases de données, qui peuvent être personnalisés pour répondre à des besoins spécifiques. Il s'intègre également à des DevOps outils populaires tels qu'AWS CodePipeline, AWS et AWS CodeCommit CodeBuild, pour permettre des flux de travail d'intégration et de déploiement continus (CI/CD).

En utilisant AWS Proton, les développeurs peuvent réduire le temps et les efforts nécessaires au déploiement et à la gestion des microservices et des applications basées sur des conteneurs. Cette approche permet aux équipes de se concentrer sur le développement et l'amélioration de leurs applications, plutôt que de consacrer du temps au processus de déploiement et de gestion.

## AWS App2Container

[AWS App2Container](#) est un outil de ligne de commande permettant de migrer et de moderniser les applications Web Java et .NET au format conteneur. App2Container analyse et dresse un inventaire des applications exécutées sur du matériel vierge, des machines virtuelles, des EC2 instances Amazon ou dans le cloud. Il vous suffit de sélectionner l'application que vous souhaitez conteneuriser, et App2Container empaquète l'artefact de l'application et les dépendances identifiées dans des images de conteneur, configure les ports réseau et génère les définitions de tâches ECS et de pods Kubernetes. App2Container identifie les applications ASP.NET et Java prises en charge exécutées sur une machine virtuelle afin de créer un inventaire complet de toutes les applications de

votre environnement. App2Container peut conteneuriser des applications Web ASP.NET exécutées dans IIS sous Windows ou des applications Java exécutées sous Linux, de manière autonome ou sur des serveurs d'applications tels qu'Apache Tomcat, Springboot JBoss, IBM Websphere et Oracle Weblogic.

## Copilote AWS

[AWS Copilot](#) est une interface de ligne de commande (CLI) que vous pouvez utiliser pour lancer et gérer rapidement des applications conteneurisées sur AWS. Il simplifie l'exécution des applications sur Amazon ECS, Fargate et App Runner. AWS Copilot prend actuellement en charge les systèmes Linux, macOS et Windows. Copilot vous permet d'utiliser des modèles de service tels qu'un service Web à charge équilibrée pour provisionner l'infrastructure, de déployer dans plusieurs environnements tels que les tests ou la production, et même d'utiliser un pipeline de AWS CodePipeline publication pour les déploiements automatisés.

## AWS Serverless Application Model

The [AWS Serverless Application Model](#)(AWS SAM) est un framework open source permettant de créer des applications sans serveur. Il fournit une syntaxe abrégée pour exprimer les fonctions APIs, les bases de données et les mappages de sources d'événements. Avec seulement quelques lignes par ressource, vous pouvez définir l'application que vous souhaitez et la modéliser à l'aide de YAML. Au cours du déploiement, SAM transforme et étend la syntaxe SAM en CloudFormation syntaxe AWS, ce qui vous permet de créer des applications sans serveur plus rapidement.

La AWS SAM CLI est un outil de ligne de commande open source qui facilite le développement, le test et le déploiement d'applications sans serveur sur AWS. Il s'agit d'une interface de ligne de commande permettant de créer des applications sans serveur à l'aide de la spécification AWS SAM, qui est une extension d'AWS. CloudFormation

La AWS SAM CLI permet aux développeurs de définir et de tester leurs applications sans serveur localement avant de les déployer sur AWS. Il fournit un environnement de test local qui simule AWS Lambda et API Gateway, permettant aux développeurs de tester leur code et leurs configurations avant de les déployer dans le cloud.

La AWS SAM CLI inclut également diverses fonctionnalités utiles, telles que le déploiement automatique du code, la journalisation et les fonctionnalités de débogage. Il permet aux développeurs de créer, d'empaqueter et de déployer leurs applications à l'aide d'une seule commande, réduisant ainsi le temps et les efforts nécessaires au déploiement et à la gestion des applications sans serveur.

En outre, l'interface de ligne de commande AWS SAM prend en charge divers langages de programmation, notamment Node.js, Python, Java et .NET Core. Cela permet aux développeurs d'utiliser leur langage de programmation et leurs outils préférés pour créer et déployer leurs applications sans serveur.

La CLI AWS SAM s'intègre à d'autres services AWS, tels qu'AWS CodePipeline et AWS CodeBuild, afin de fournir un CI/CD pipeline entièrement automatisé et intégré pour les applications sans serveur. Il permet également aux développeurs d'utiliser d'autres services AWS, tels qu'Amazon S3, Amazon DynamoDB et Amazon SNS, dans le cadre de leurs applications sans serveur.

## AWS Cloud Development Kit (AWS CDK)

The [AWS Cloud Development Kit \(AWS CDK\)](#) (AWS CDK) est un framework de développement de logiciels open source permettant de définir l'infrastructure cloud sous forme de code utilisant des langages de programmation modernes et de la déployer via AWS CloudFormation. AWS Cloud Development Kit (AWS CDK) accélère le développement du cloud en utilisant des langages de programmation courants pour modéliser vos applications. L'AWS CDK vous permet de créer des applications fiables, évolutives et économiques dans le cloud grâce à la puissance expressive considérable d'un langage de programmation.

Considérez le CDK AWS comme une boîte à outils centrée sur le développeur qui tire parti de toute la puissance des langages de programmation modernes pour définir votre infrastructure AWS sous forme de code. Lorsque les applications AWS CDK sont exécutées, elles sont compilées sous forme de modèles CloudFormation JSON/YAML complets qui sont ensuite soumis au service pour le provisionnement. CloudFormation Grâce à l'utilisation d'AWS CDK CloudFormation, vous bénéficiez toujours de tous les avantages CloudFormation tels que le déploiement sécurisé, le rollback automatique et la détection des dérives.

Cette approche présente de nombreux avantages, notamment :

- Développez à l'aide de structures de haut niveau qui fournissent automatiquement des valeurs par défaut sensées et sécurisées pour vos ressources AWS, définissant ainsi davantage d'infrastructures avec moins de code.
- Utilisez des expressions idiomatiques de programmation telles que les paramètres, les conditions, les boucles, la composition et l'héritage pour modéliser la conception de votre système à partir des éléments de base fournis par AWS et d'autres fournisseurs.
- Centralisez votre infrastructure, votre code d'application et votre configuration en un seul endroit, afin de disposer à chaque étape d'un système complet pouvant être déployé dans le cloud.

- Utilisez des pratiques d'ingénierie logicielle telles que la révision du code, les tests unitaires et le contrôle des sources pour renforcer la robustesse de votre infrastructure.
- AWS Solutions Constructs est une extension de bibliothèque open source d'AWS CDK. AWS Solutions Constructs met à votre disposition un ensemble de modèles d'architecture multiservices approuvés conçus selon les meilleures pratiques établies par le AWS Well-Architected Framework.

Le modèle d'application sans serveur AWS et le CDK AWS résument l'infrastructure AWS sous forme de code, ce qui vous permet de définir plus facilement votre infrastructure cloud. AWS SAM se concentre spécifiquement sur les cas d'utilisation et les architectures sans serveur et vous permet de définir votre infrastructure dans des modèles déclaratifs JSON/YAML compacts. AWS CDK offre une large couverture de tous les services AWS et vous permet de définir une infrastructure cloud dans des langages de programmation modernes

## Amazon EC2 Image Builder

[EC2 Image Builder](#) simplifie la création, le test et le déploiement d'images de machines virtuelles et de conteneurs à utiliser sur AWS ou sur site. La conservation des images de machines virtuelles et de conteneurs up-to-date peut s'avérer chronophage, gourmande en ressources et source d'erreurs. Actuellement, les clients mettent à jour et capturent manuellement des images VMs ou ont des équipes qui créent des scripts d'automatisation pour gérer les images. Image Builder réduit considérablement les efforts liés à la conservation up-to-date et à la sécurité des images en fournissant une interface graphique simple, une automatisation intégrée et des paramètres de sécurité fournis par AWS. Avec Image Builder, il n'y a aucune étape manuelle pour mettre à jour une image et vous n'avez pas besoin de créer votre propre pipeline d'automatisation. Image Builder est proposé gratuitement, hormis le coût des ressources AWS sous-jacentes utilisées pour créer, stocker et partager les images.

EC2 Image Builder peut faciliter les déploiements sur AWS en simplifiant le processus de création et de gestion d'images personnalisées à utiliser avec Amazon EC2, les conteneurs et les serveurs sur site. Le service fournit un moyen simplifié et flexible de créer et de gérer des images personnalisées, avec des pipelines de création automatisés qui vous permettent de rationaliser le processus de création et de gestion des images.

EC2 Image Builder fournit une interface conviviale qui fait abstraction d'une grande partie de la gestion de l'infrastructure sous-jacente, ce qui permet aux développeurs de créer et de gérer plus facilement des images personnalisées. Avec EC2 Image Builder, les développeurs peuvent spécifier le système d'exploitation, les applications et les packages qu'ils souhaitent inclure dans l'image,

et le service automatise le processus de création et de test de l'image, y compris les mises à jour, les correctifs et les correctifs de sécurité. Les pipelines de génération automatisés permettent aux développeurs de rationaliser le processus de création et de gestion des images, réduisant ainsi le temps et les efforts nécessaires à la création et au test manuels des images. Cela peut contribuer à améliorer la cohérence, à réduire les erreurs et à garantir la sécurité et la conformité des images. up-to-date

Voici quelques-uns des avantages d' EC2 Image Builder :

- **Création d'images simplifiée** : EC2 Image Builder fournit un moyen simplifié et flexible de créer des images personnalisées à utiliser avec Amazon EC2, les conteneurs et les serveurs sur site. Cela peut contribuer à réduire le temps et les efforts nécessaires pour créer et gérer des images personnalisées, et vous permettre de vous concentrer sur d'autres aspects du déploiement, tels que le développement et les tests d'applications.
- **Pipelines de création d'images automatisés** : EC2 Image Builder fournit des pipelines automatisés pour créer, tester et déployer des images personnalisées, ce qui peut contribuer à rationaliser le processus de création et de gestion des images. Cela peut contribuer à garantir la sécurité et la conformité de vos images, et à réduire le temps et les efforts nécessaires à la création et au test manuels des images. up-to-date
- **Intégration aux services AWS** : EC2 Image Builder s'intègre à d'autres services AWS, tels qu'Amazon Elastic Container Registry (ECR) et Amazon Elastic Kubernetes Service (EKS), pour vous permettre de créer des images personnalisées à utiliser avec des conteneurs. Cela peut aider à rationaliser le processus de création et de déploiement du conteneur, ce qui vous permet de créer des images personnalisées qui incluent vos applications, vos bibliothèques et vos configurations.
- **Création d'images flexible** : EC2 Image Builder fournit un moyen flexible de créer des images personnalisées, vous permettant de spécifier le système d'exploitation, les applications et les packages que vous souhaitez inclure dans l'image. Cela permet de garantir que vos images sont adaptées à votre cas d'utilisation et à vos exigences spécifiques, et de réduire le risque d'erreurs ou d'incompatibilités lors du déploiement.
- **Sécurité et conformité des images améliorées** : EC2 Image Builder vous permet d'automatiser les tests d'images, y compris les analyses de vulnérabilité et de conformité, afin de garantir la sécurité et la conformité de vos images. Cela peut contribuer à réduire le risque de violations de sécurité, à améliorer la conformité et à vous permettre de déployer vos applications en toute confiance.

# Stratégies de déploiement

Outre la sélection des bons outils pour mettre à jour le code de votre application et l'infrastructure de support, la mise en œuvre des processus de déploiement appropriés est un élément essentiel d'une solution de déploiement complète et performante. Les processus de déploiement que vous choisissez pour mettre à jour votre application peuvent dépendre de l'équilibre que vous souhaitez atteindre en termes de contrôle, de rapidité, de coût, de tolérance au risque et d'autres facteurs.

Chaque service de déploiement AWS prend en charge un certain nombre de stratégies de déploiement. Cette section fournit une vue d'ensemble des stratégies de déploiement générales qui peuvent être utilisées avec votre solution de déploiement.

## Précuisson ou amorçage AMIs

Si votre application repose principalement sur la personnalisation ou le déploiement d'applications sur des EC2 instances Amazon, vous pouvez optimiser vos déploiements grâce à des pratiques d'amorçage et de précuisson.

L'installation de votre application, de vos dépendances ou de vos personnalisations à chaque fois qu'une EC2 instance Amazon est lancée s'appelle le démarrage d'une instance. Si vous avez besoin d'une application complexe ou de téléchargements volumineux, cela peut ralentir les déploiements et les événements de dimensionnement.

Une [Amazon Machine Image](#) (AMI) fournit les informations requises pour lancer une instance (systèmes d'exploitation, volumes de stockage, autorisations, packages logiciels, etc.). Vous pouvez lancer plusieurs instances identiques à partir d'une seule AMI. Chaque fois qu'une EC2 instance est lancée, vous sélectionnez l'AMI à utiliser comme modèle. Le précuisson est le processus qui consiste à intégrer une partie importante des artefacts de votre application dans une AMI.

Le prémontage des composants de l'application dans une AMI peut accélérer le lancement et l'opérationnalisation d'une instance Amazon EC2 . Les pratiques de précuisson et d'amorçage peuvent être combinées au cours du processus de déploiement afin de créer rapidement de nouvelles instances personnalisées en fonction de l'environnement actuel.

## Déploiements bleus/verts

Un blue/green déploiement est une stratégie de déploiement dans laquelle vous créez deux environnements distincts mais identiques. Un environnement (bleu) exécute la version actuelle de

l'application et un environnement (vert) exécute la nouvelle version de l'application. L'utilisation d'une stratégie de blue/green déploiement augmente la disponibilité des applications et réduit les risques liés au déploiement en simplifiant le processus de restauration en cas d'échec d'un déploiement. Une fois les tests terminés sur l'environnement vert, le trafic réel des applications est dirigé vers l'environnement vert et l'environnement bleu est obsolète.

Un certain nombre de services de déploiement AWS prennent en charge les stratégies de blue/green déploiement, notamment Elastic OpsWorks Beanstalk CloudFormation,, CodeDeploy, et Amazon ECS. Reportez-vous à la section [Déploiements bleu/vert sur AWS](#) pour plus de détails et pour connaître les stratégies de mise en œuvre des processus de blue/green déploiement pour votre application.

## Déploiements propagés

Un déploiement progressif est une stratégie de déploiement qui remplace progressivement les versions précédentes d'une application par de nouvelles versions d'une application en remplaçant complètement l'infrastructure sur laquelle l'application s'exécute. Par exemple, lors d'un déploiement continu dans Amazon ECS, les conteneurs exécutant les versions précédentes de l'application seront one-by-one remplacés par des conteneurs exécutant les nouvelles versions de l'application.

Un déploiement progressif est généralement plus rapide qu'un blue/green déploiement ; toutefois, contrairement à un blue/green déploiement continu, il n'y a aucune isolation environnementale entre les anciennes et les nouvelles versions de l'application. Cela permet aux déploiements progressifs de se terminer plus rapidement, mais augmente également les risques et complique le processus de restauration en cas d'échec d'un déploiement.

Les stratégies de déploiement progressif peuvent être utilisées avec la plupart des solutions de déploiement. [Reportez-vous aux politiques de CloudFormation mise à jour pour plus d'informations sur les déploiements progressifs avec Amazon ECS CloudFormation ; aux mises à jour continues avec Amazon ECS ; aux mises à jour de configuration de l'environnement évolutif d'Elastic Beanstalk pour plus de détails sur les déploiements progressifs avec Elastic Beanstalk ; et à l'utilisation d'un déploiement progressif pour plus de détails sur le déploiement avec. AWS OpsWorks OpsWorks](#)

## Déploiements Canary

Les [déploiements Canary](#) constituent un type de stratégie de blue/green déploiement moins risqué. Cette stratégie implique une approche progressive dans laquelle le trafic est transféré vers une nouvelle version de l'application en deux étapes. Le premier incrément correspond à un faible

pourcentage du trafic, que l'on appelle le groupe Canary. Ce groupe est utilisé pour tester la nouvelle version et, en cas de succès, le trafic est transféré vers la nouvelle version au cours du deuxième incrément.

Les déploiements Canary peuvent être mis en œuvre en deux étapes ou de manière linéaire. Dans le cadre de l'approche en deux étapes, le nouveau code d'application est déployé et exposé à des fins d'essai. Une fois accepté, il est déployé soit dans le reste de l'environnement, soit de manière linéaire. L'approche linéaire consiste à augmenter progressivement le trafic vers la nouvelle version de l'application jusqu'à ce que tout le trafic soit acheminé vers la nouvelle version.

## Déploiements sur place

Un [déploiement sur place](#) est une stratégie de déploiement qui met à jour la version de l'application sans remplacer les composants de l'infrastructure. Lors d'un déploiement sur place, la version précédente de l'application sur chaque ressource de calcul est arrêtée, la dernière application est installée et la nouvelle version de l'application est démarrée et validée. Cela permet aux déploiements d'applications de se dérouler en perturbant le moins possible l'infrastructure sous-jacente.

Un déploiement sur place vous permet de déployer votre application sans créer de nouvelle infrastructure ; toutefois, la disponibilité de votre application peut être affectée lors de ces déploiements. Cette approche minimise également les coûts d'infrastructure et les frais de gestion associés à la création de nouvelles ressources.

Reportez-vous à la section [Présentation d'un déploiement sur place](#) pour plus de détails sur l'utilisation de stratégies de déploiement sur place avec CodeDeploy.

## Combiner les services de déploiement

Il n'existe pas de solution de déploiement « universelle » sur AWS. Dans le contexte de la conception d'une solution de déploiement, il est important de prendre en compte le type d'application, car cela peut dicter les services AWS les plus appropriés. Pour fournir des fonctionnalités complètes de provisionnement, de configuration, de déploiement, de mise à l'échelle et de surveillance de votre application, il est souvent nécessaire de combiner plusieurs services de déploiement.

Un modèle courant pour les applications sur AWS consiste à utiliser CloudFormation (et ses extensions) pour gérer l'infrastructure à usage général et à utiliser une solution de déploiement plus spécialisée pour gérer les mises à jour des applications. Dans le cas d'une application conteneurisée,

elle CloudFormation peut être utilisée pour créer l'infrastructure de l'application, et Amazon ECS et Amazon EKS peuvent être utilisés pour approvisionner, déployer et surveiller des conteneurs.

Les services de déploiement AWS peuvent également être combinés à des services de déploiement tiers. Cela permet aux entreprises d'intégrer facilement les services de déploiement AWS dans leurs CI/CD pipelines ou leurs solutions de gestion d'infrastructure existants. Par exemple, il OpsWorks peut être utilisé pour synchroniser les configurations entre les nœuds sur site et les nœuds AWS, et CodeDeploy peut être utilisé avec un certain nombre de CI/CD services tiers dans le cadre d'un pipeline complet.

## Conclusion

AWS fournit un certain nombre d'outils pour simplifier et automatiser le provisionnement de l'infrastructure et le déploiement des applications ; chaque service de déploiement propose différentes fonctionnalités de gestion des applications. Pour créer une architecture de déploiement efficace, évaluez les fonctionnalités disponibles de chaque service en fonction des besoins de votre application et de votre organisation.

# Collaborateurs

Les personnes qui ont contribué à ce document incluent :

- Manikandan Chandrasekaran, technologue principal
- Anil Nadiminti, architecte de solutions senior
- Bryant Bost, consultant AWS ProServe

## Suggestions de lecture

Pour en savoir plus, voir :

- [Page des livres blancs AWS](#)
- [Présentation d' DevOps AWS - Stratégies de déploiement](#)

## Révisions du document

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

Modification	Description	Date
<a href="#">Livre blanc mis à jour</a>	Entièrement mis à jour pour tenir compte des derniers services et stratégies de déploiement	31 mai 2024
<a href="#">Mise à jour mineure</a>	La section sur les déploiements bleu/vert a été révisée pour plus de clarté.	08 avril 2021
<a href="#">Livre blanc mis à jour</a>	Mis à jour avec les derniers services et fonctionnalités.	3 juin 2020
<a href="#">Publication initiale</a>	Livre blanc publié pour la première fois	1er mars 2015

## Notifications

Il incombe aux clients de procéder à une évaluation indépendante des informations contenues dans le présent document. Ce document : (a) est fourni à titre informatif uniquement, (b) représente les offres de produits et les pratiques actuelles d'AWS, qui sont susceptibles d'être modifiées sans préavis, et (c) ne crée aucun engagement ni aucune garantie de la part d'AWS et de ses filiales, fournisseurs ou concédants de licence. Les produits ou services AWS sont fournis « tels quels » sans garanties, déclarations ou conditions d'aucune sorte, qu'elles soient explicites ou implicites. Les responsabilités et obligations d'AWS vis-à-vis de ses clients sont régies par les contrats AWS. Le présent document ne fait partie d'aucun, et ne modifie aucun, contrat entre AWS et ses clients.

© 2024 Amazon Web Services, Inc. ou ses filiales. Tous droits réservés.

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.