Choisir le bon GitOps outil pour votre cluster Amazon EKS

AWS Conseils prescriptifs

AWS Conseils prescriptifs: Choisir le bon GitOps outil pour votre cluster Amazon EKS

Table of Contents

Introduction 1
Résultats commerciaux ciblés 2
Intégration parfaite avec Amazon EKS 2
Scalabilité et performance 2
Sécurité et conformité 2
Facilité d'utilisation et courbe d'apprentissage 3
Soutien à la communauté et au réseau 3
Fonctionnalités de gestion multi-clusters 3
Observabilité et surveillance 4
Flexibilité et personnalisation 4
Livraison continue et assistance au déploiement progressif 4
Rentabilité et utilisation des ressources 5
GitOps outils pour les clusters EKS 6
Argo CD 6
GitOps soutien 6
Architecture 9
Flux 10
GitOps soutien 10
Architecture 13
Tisser GitOps 13
GitOps soutien 14
Architecture 17
Jenkins X 18
GitOps soutien 18
Architecture 21
GitLab CI/CD 22
GitOps soutien 22
Spinnaker 26
GitOps soutien 26
Architecture 29
Flotte d'éleveurs 31
GitOps soutien 31
Architecture 34
Code Fresh 35

GitOps soutien 35
Pulumi 39
GitOps soutien 39
GitOps comparaison des outils 44
Facilité d'utilisation 44
Intégration avec Kubernetes 44
Capacités CI/CD 44
GitOps pureté 44
Support multicloud 45
Prise en charge de plusieurs clusters 45
Integration 45
Communauté et soutien 45
Fonctionnalités d'entreprise 45
Flexibilité et extensibilité 46
Evolutivité 46
Gestion de l'infrastructure 46
Support du modèle de programmation et du langage 46
Cas d'utilisation d'Argo CD et de Flux 47
Considérations d’ordre général 47
Cas d'utilisation d'Argo CD 47
Cas d'utilisation de Flux 48
Comparaison des fonctionnalités 50
Bonnes pratiques pour choisir un GitOps outil 52
FAQ 58
Ressources 61
Historique du document 62
Glossaire 63
..... 63
A 64
B 67
C 69
D 72
E 76
F 79
G 81
H 82

I 84
L 86
M 87
O 92
P 94
Q 97
R 98
S 101
T 105
U 106
V 107
W 108
Z 109
..... cx

Choisir le bon GitOps outil pour votre cluster Amazon EKS

Pradip Kumar Pandey et Pratap Kumar Nanda, Amazon Web Services ()AWS

Avril 2025 (historique du document)

Dans le paysage en évolution rapide des technologies natives du cloud, GitOps elle est devenue une puissante méthodologie pour gérer et déployer des applications et des infrastructures. Si vous utilisez Amazon Elastic Kubernetes Service (Amazon EKS) GitOps , la mise en œuvre de principes peut considérablement améliorer vos processus de déploiement, améliorer la fiabilité et rationaliser les opérations. De nombreux GitOps outils sont disponibles, et choisir celui qui convient à votre cluster EKS est une décision critique qui peut avoir un impact sur l'efficacité de votre équipe et le succès global de vos DevOps pratiques.

La sélection d'un GitOps outil adapté à votre environnement Amazon EKS implique un examen attentif de divers facteurs, notamment vos exigences spécifiques, l'expertise de votre équipe, vos besoins en évolutivité et les capacités d'intégration avec les outils existants Services AWS. Chaque outil possède son propre ensemble de fonctionnalités, de points forts et de limites potentielles. Il est donc essentiel d'aligner votre choix sur les objectifs et le contexte opérationnel de votre organisation.

Ce guide explore les principaux facteurs à prendre en compte lors de la sélection d' GitOps outils pour Amazon EKS, compare les options fréquemment utilisées et fournit des informations pour vous aider à prendre une décision éclairée. Il couvre neuf GitOps outils populaires :

- Argo CD
- Flux
- Tisser GitOps
- Jenkins X
- GitLab CI/CD
- Spinnaker
- Flotte de ranchers
- Code Fresh
- Pulumi

Résultats commerciaux ciblés

La liste suivante décrit les objectifs et les résultats potentiels lorsque vous choisissez un outil pour mettre en œuvre des GitOps principes dans vos processus de développement et d'exploitation.

Intégration parfaite avec Amazon EKS

Votre GitOps outil doit s'intégrer parfaitement à Amazon EKS et être compatible avec les fonctionnalités et optimisations spécifiques à Amazon EKS.

- Support natif d'Amazon EKS : recherchez des outils offrant une prise en charge intégrée d'Amazon EKS, notamment une connexion et une gestion simplifiées des clusters.
- Service AWS intégration : assurez-vous que l'outil peut interagir avec d'autres outils Services AWS tels que Gestion des identités et des accès AWS (IAM), Amazon Elastic Container Registry (Amazon ECR) et Amazon. CloudWatch
- Compatibilité des modules complémentaires Amazon EKS : vérifiez que l'outil prend en charge les modules complémentaires Amazon EKS et qu'il est capable de les gérer efficacement.

Scalabilité et performance

Votre GitOps outil doit être capable de gérer l'échelle de vos opérations Amazon EKS, qu'il s'agisse de petits clusters ou de grands environnements à clusters multiples.

- Efficacité des ressources : évaluez la consommation de ressources de l'outil et son impact sur les performances du cluster.
- Opérations à grande échelle : évaluez la capacité de l'outil à gérer simultanément de nombreuses applications et clusters.
- Performances sous charge : considérez les performances de l'outil lors de mises à jour fréquentes et de déploiements à grande échelle.

Sécurité et conformité

Les fonctionnalités de sécurité et de conformité sont cruciales, en particulier dans les secteurs réglementés ou lorsque vous manipulez des données sensibles.

- Contrôle d'accès : recherchez des fonctionnalités robustes de contrôle d'accès basé sur les rôles (RBAC) qui s'intègrent à IAM.

- Gestion des secrets : évaluez la manière dont l'outil gère les informations sensibles et s'intègre à AWS Secrets Managerd'autres solutions.
- Pistes d'audit : assurez-vous que l'outil fournit des fonctionnalités complètes de journalisation et d'audit pour la conformité et le dépannage.
- Analyse de sécurité : pensez aux outils qui proposent une analyse de sécurité intégrée pour détecter les vulnérabilités dans les déploiements.

Facilité d'utilisation et courbe d'apprentissage

L'outil doit être convivial et correspondre aux compétences de votre équipe pour garantir une adoption rapide et une utilisation efficace.

- Interface utilisateur : évaluez l'intuitivité des fonctionnalités de l'interface de ligne de commande (CLI) et de l'interface utilisateur graphique (GUI).
- Qualité de la documentation : recherchez une up-to-date documentation et des didacticiels complets.
- Ressources d'apprentissage : Tenez compte de la disponibilité du matériel de formation, des cours et des ressources communautaires.

Soutien à la communauté et au réseau

Une communauté et un réseau solides peuvent fournir des ressources précieuses, des plugins et une durabilité à long terme.

- Développement actif : vérifiez la fréquence des mises à jour et la réactivité des mainteneurs.
- Taille de la communauté : tenez compte de la taille et de l'activité de la communauté d'utilisateurs pour le soutien et le partage des connaissances.
- Intégrations tierces : évaluez la disponibilité des plugins et des intégrations avec les autres outils de votre stack.

Fonctionnalités de gestion multi-clusters

Si vous disposez de plusieurs clusters EKS, la capacité à les gérer efficacement est cruciale.

- Gestion centralisée : recherchez des fonctionnalités qui permettent de gérer plusieurs clusters à partir d'un seul plan de contrôle.

- Fédération de clusters : pensez aux outils qui prennent en charge la fédération Kubernetes pour les applications multi-clusters.
- Parité environnementale : évaluez dans quelle mesure l'outil maintient la cohérence entre les différents environnements tels que le développement, la mise en scène et la production.

Observabilité et surveillance

L'outil doit fournir des informations claires sur l'état de vos déploiements et sur l'état de santé de votre cluster.

- Visibilité du déploiement : recherchez des fonctionnalités offrant une vue claire de l'état et de l'historique du déploiement.
- Intégration aux outils de surveillance : déterminez dans quelle mesure l'outil s'intègre aux solutions de surveillance populaires telles que Prometheus et Grafana.
- Capacités d'alerte : évaluez la capacité de l'outil à configurer et à gérer des alertes en cas de problème de déploiement ou de dérive.

Flexibilité et personnalisation

La capacité d'adapter l'outil à vos flux de travail et à vos exigences spécifiques est importante pour une satisfaction à long terme.

- Extensibilité : recherchez des architectures de plug-in ou des architectures APIs qui vous permettent d'étendre les fonctionnalités de l'outil.
- Support des ressources personnalisées : vérifiez que l'outil peut gérer efficacement les ressources Kubernetes personnalisées.
- Personnalisation des flux de travail : évaluez la facilité avec laquelle vous pouvez adapter les GitOps flux de travail aux besoins de votre équipe.

Livraison continue et assistance au déploiement progressif

Les stratégies de déploiement avancées sont souvent cruciales pour minimiser les risques et garantir des mises à jour fluides.

- Déploiements Canary : recherchez un support intégré pour les versions de Canary.
- Blue/green deployments: Assess the tool's capabilities for blue/greenstratégies de déploiement.

- Mécanismes d'annulation : garantissez des fonctionnalités robustes et de easy-to-use restauration pour une reprise rapide en cas d'échec des déploiements.

Rentabilité et utilisation des ressources

Tenez compte du coût global de l'adoption et de la maintenance de l'outil, y compris les coûts directs et indirects.

- Coûts de licence : comparez les options open source aux solutions commerciales et considérez le support et les fonctionnalités d'entreprise.
- Frais d'exploitation : évaluez les coûts opérationnels supplémentaires en termes de gestion et de maintenance.
- Consommation de ressources : évaluez l'efficacité de l'outil en termes de ressources de calcul et de stockage qui seraient nécessaires.

En examinant attentivement ces résultats et leurs aspects, vous pouvez prendre une décision éclairée quant à l' GitOps outil le plus adapté à votre cluster EKS et vous assurer que l'outil correspond aux besoins, aux capacités et à la stratégie à long terme de votre organisation.

GitOps outils pour les clusters EKS

Plusieurs GitOps outils pour Kubernetes sont actuellement disponibles sur le marché. Voici une liste des options les plus utilisées :

- Argo CD
- Flux
- Tisser GitOps
- Jenkins X
- GitLab CI/CD
- Spinnaker
- Flotte d'éleveurs
- Code Fresh
- Pulumi

Suivez les liens pour obtenir des informations détaillées sur la manière dont ces outils mettent en œuvre GitOps les pratiques. Chaque outil a ses points forts et ses cas d'utilisation. Le choix dépend de facteurs tels que vos exigences spécifiques, l'infrastructure existante, l'expertise de l'équipe et les fonctionnalités souhaitées. Il est important d'évaluer ces outils en fonction des besoins de votre entreprise et de la complexité de votre environnement Kubernetes.

Argo CD

Argo CD est un outil de diffusion GitOps continue (CD) largement utilisé pour Kubernetes qui respecte plusieurs principes clés. GitOps

GitOps soutien

Area	Capacités des outils
Configuration déclarative	Argo CD utilise des configurations déclarati ves stockées dans des référentiels Git. L'état souhaité de l'application et de l'infrastructure est défini dans les fichiers YAML. Ces configura

Area	Capacités des outils
	tions décrivent ce qui doit être déployé, et non comment le déployer.
Le système de contrôle de version en tant que source unique de vérité	Les référentiels Git constituent la source unique de vérité pour l'ensemble du système. Toutes les modifications apportées à l'application et à l'infrastructure sont effectuées via Git. Cela garantit une piste d'audit complète et la possibili té de revenir à n'importe quel état précédent.
Synchronisation automatisée	Argo CD surveille en permanence les modificat ions apportées au dépôt Git. Lorsque des modifications sont détectées, il synchronise automatiquement l'état réel du cluster avec l'état souhaité défini dans Git. Cela garantit que le cluster reflète toujours l'état décrit dans le référentiel.
Natif de Kubernetes	Argo CD est spécialement conçu pour les environnements Kubernetes. Il tire parti de la nature déclarative et des ressources personnal isées de Kubernetes pour gérer les applicati ons.
Auto-guérison et détection de dérive	Argo CD compare régulièrement l'état réel du cluster avec l'état souhaité dans Git. S'il détecte une dérive (différences entre l'état réel et l'état souhaité), il peut corriger automatiquement ces écarts.
Support multi-clusters et multi-locataires	Argo CD peut gérer plusieurs clusters Kubernetes à partir d'une seule instance. Il prend en charge la mutualisation, ce qui permet aux différentes équipes de gérer leurs applicati ons de manière indépendante.

Area	Capacités des outils
Définition de l'application	Les applications d'Argo CD sont définies à l'aide du CRD d'application (définition de ressource personnalisée). Cela permet de définir de manière native à Kubernetes ce qui doit être déployé et comment.
Séparation du déploiement et de la publication	Argo CD sépare le déploiement du code de sa mise à disposition des utilisateurs. Ceci est réalisé grâce à diverses stratégie s de déploiement telles que les blue/green déploiements Canary.
Observabilité et auditabilité	Argo CD fournit une interface utilisateur Web et une CLI pour observer l'état des applications et des clusters. Toutes les actions sont enregistr ées afin de fournir une piste d'audit claire des modifications et des déploiements.
Sécurité et RBAC	Argo CD s'intègre au contrôle d'accès basé sur les rôles (RBAC) de Kubernetes. Il prend en charge l'intégration de l'authentification unique pour l'authentification et l'autorisation.
Architecture enfichable	Argo CD prend en charge divers systèmes de gestion de contrôle de source, les graphique s Helm, Kustomize et d'autres formats de manifeste Kubernetes. Cette flexibilité lui permet de s'adapter à divers environnements et flux de travail.
Livraison continue (CD)	Bien qu'Argo CD se concentre sur la livraison continue, il peut être intégré à des outils d'intégration continue (CI) pour créer un CI/CD pipeline complet.

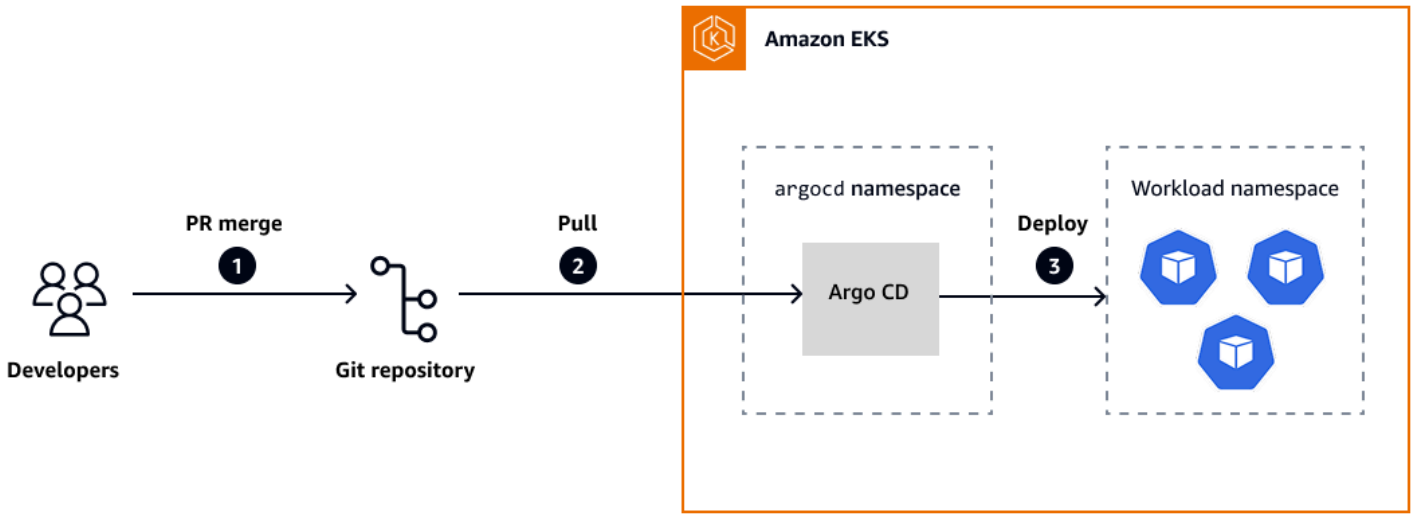
En respectant ces GitOps principes, Argo CD fournit un moyen robuste, évolutif et sécurisé de gérer les déploiements Kubernetes. Il garantit que l'état opérationnel de votre système est toujours synchronisé avec l'état souhaité défini dans votre référentiel Git, et favorise la cohérence, la fiabilité et la facilité de gestion dans les environnements Kubernetes complexes.

Pour les scénarios et les exigences auxquels Argo CD peut répondre, voir les cas d'utilisation d'Argo CD plus loin dans ce guide. Pour une comparaison entre Argo CD et Flux, voir Comparaison des fonctionnalités plus loin dans ce guide.

Pour plus d'informations, consultez la documentation du CD Argo.

Architecture

Le schéma suivant illustre un flux de travail CD GitOps piloté qui utilise Argo CD au sein d'un cluster EKS. Pour des informations détaillées, consultez la documentation du CD Argo.



où :

- Étape 1 : fusion par pull request (PR). Un développeur valide les modifications apportées aux manifestes Kubernetes ou aux graphiques Helm stockés dans un référentiel Git. Lorsque le PR a été révisé et fusionné dans la branche principale, l'état souhaité de l'application est mis à jour dans le contrôle de source.
- Étape 2 : synchronisation du référentiel. Argo CD s'exécute dans un espace de noms dédié (`argocd`) du cluster EKS et surveille en permanence le référentiel Git configuré. Lorsqu'il détecte des modifications, il extrait les dernières mises à jour pour réconcilier l'état déclaré.
- Étape 3 : Déploiement vers l'espace de noms cible Argo CD compare l'état souhaité dans Git avec l'état actif du cluster. Il applique ensuite les modifications nécessaires à l'espace de noms de

charge de travail cible afin que l'application soit déployée ou mise à jour en conséquence. Cela inclut la gestion des ressources Kubernetes telles que les déploiements, les services et les secrets afin de maintenir la cohérence du cluster avec la source de vérité Git. ConfigMaps

Flux

Flux est un autre outil pour Kubernetes qui implémente les GitOps principes d'une manière unique.

GitOps soutien

Area	Capacités des outils
Git comme source unique de vérité	Flux utilise les référentiels Git comme source définitive pour définir l'état souhaité du système. Toute la configuration des applicati ons et de l'infrastructure est stockée dans Git.
Configuration déclarative	Flux fonctionne avec des descriptions déclarati ves de l'état souhaité de votre cluster. Ces descriptions sont généralement des manifeste s Kubernetes, des diagrammes de Helm ou des incrustations Kustomize.
Synchronisation automatisée	Flux surveille en permanence les modifications apportées au dépôt Git. Lorsqu'il détecte des modifications, il les applique automatiquement au cluster.
Natif de Kubernetes	Flux est conçu comme un ensemble de contrôleurs Kubernetes et de ressources personnalisées. Il utilise les mécanismes d'extension de Kubernetes pour fournir des fonctionnalités. GitOps
Modèle de déploiement basé sur le pull	Contrairement aux CI/CD systèmes tradition nels basés sur le push, Flux utilise un modèle basé sur le pull. Le cluster extrait l'état souhaité

Area	Capacités des outils
	de Git au lieu d'utiliser un système externe pour appliquer les modifications.
Réconciliation continue	Flux compare constamment l'état réel du cluster avec l'état souhaité dans Git. Il corrige automatiquement toute dérive détectée entre ces états.
Multilocataire	Flux soutient la mutualisation grâce à ses concepts de kustomisations et. HelmReleases Les différentes équipes peuvent gérer leurs propres parties de la configuration de manière indépendante.
Livraison progressive	Flux prend en charge les stratégies de déploiement avancées, telles que les versions et les A/B tests de Canary, via son composant Flagger.
Intégration avec Helm	Flux inclut un support natif pour Helm, ce qui vous permet de gérer facilement les versions de Helm GitOps.
Automatisation de la mise à jour	Flux peut mettre à jour automatiquement les images des conteneurs dans Git lorsque de nouvelles versions sont disponibles dans le registre des conteneurs.
Personnaliser le support	Vous pouvez utiliser le support natif fourni par Flux for Kustomize pour personnaliser et corriger les manifestes Kubernetes.
Sécurité et RBAC	Flux s'intègre à Kubernetes RBAC pour le contrôle d'accès. Il prend en charge la gestion des secrets via différents backends.

Area	Capacités des outils
Observabilité	Flux fournit des informations sur le statut et des mesures relatives à la réconciliation et aux opérations. Il s'intègre aux outils de surveillance pour une meilleure observabilité.
Architecture basée sur les événements	Flux utilise une approche axée sur les événements pour mettre en œuvre les rapprochements et les mises à jour.
Extensibilité	L'outil est conçu pour être extensible, afin que vous puissiez ajouter des contrôleurs et des ressources personnalisés.
Synchronisation entre clusters	Flux prend en charge la gestion de plusieurs clusters à partir d'un seul ensemble de référenti els.
Gestion des dépendances	Il permet de définir les dépendances entre les différentes parties de votre système et garantit le bon ordre des opérations.
Récepteurs Webhook	Vous pouvez configurer Flux pour recevoir des webhooks de fournisseurs Git ou d'autres systèmes afin de démarrer une réconciliation immédiate.

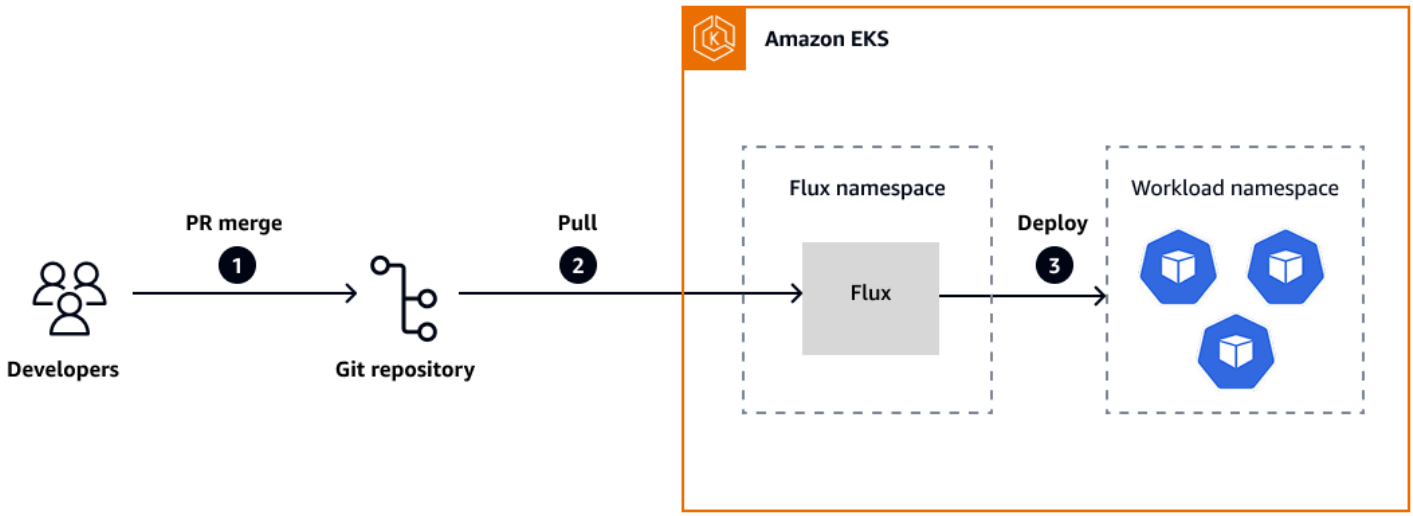
En mettant en œuvre ces GitOps principes, Flux fournit un système robuste et flexible pour gérer les clusters et les applications Kubernetes. Il garantit que votre infrastructure et vos applications sont toujours synchronisées avec vos référentiels Git et assure la cohérence, la fiabilité et la facilité de gestion dans les environnements Kubernetes complexes. L'approche native de Kubernetes de l'outil et l'accent mis sur l'automatisation le rendent particulièrement adapté aux environnements cloud natifs.

Pour les scénarios et les exigences auxquels Flux peut répondre, consultez les cas d'utilisation de Flux plus loin dans ce guide. Pour une comparaison entre Argo CD et Flux, voir Comparaison des fonctionnalités plus loin dans ce guide.

Pour plus d'informations, consultez la documentation de Flux.

Architecture

Le schéma suivant illustre un flux de travail GitOps sur CD piloté qui utilise Flux au sein d'un cluster EKS. Pour des informations détaillées, consultez la documentation de Flux.



où :

- Étape 1 : fusion par pull request (PR). Un développeur valide les modifications apportées aux manifestes Kubernetes ou aux graphiques Helm stockés dans un référentiel Git. Lorsque le PR a été révisé et fusionné dans la branche principale, l'état souhaité de l'application est mis à jour dans le contrôle de source.
- Étape 2 : synchronisation du référentiel. Flux s'exécute dans un espace de noms dédié dans le cluster EKS et surveille en permanence le référentiel Git configuré. Lorsqu'il détecte des modifications, il extrait les dernières mises à jour pour réconcilier l'état déclaré.
- Étape 3 : Déploiement vers l'espace de noms cible Flux compare l'état souhaité depuis Git avec l'état réel du cluster. Il applique ensuite les modifications nécessaires à l'espace de noms de charge de travail cible afin que l'application soit déployée ou mise à jour en conséquence.

Tisser GitOps

Weave GitOps a été développé par Weaveworks, la société qui a introduit le terme. GitOps Cet outil fournit une GitOps solution complète qui repose sur les GitOps principes de base.

GitOps soutien

Area	Capacités des outils
Git comme source unique de vérité	Weave GitOps utilise les référentiels Git comme source officielle pour définir l'état souhaité du système. Toutes les configurations, y compris les manifestes d'applications, les définitions d'infrastructure et les politiques, sont stockées dans Git.
Configuration déclarative	Le système repose sur des descriptions déclaratives de l'état complet du système. Ces descriptions sont généralement des manifeste s Kubernetes, des diagrammes de Helm ou d'autres formats déclaratifs.
Synchronisation automatisée	Weave surveille GitOps en permanence les dépôts Git pour détecter les modifications. Lorsqu'il détecte des modifications, il les applique automatiquement à l'environnement cible.
Architecture native de Kubernetes	Weave GitOps est conçu comme un ensemble de contrôleurs Kubernetes et de ressource s personnalisées. Il utilise les mécanismes d'extension de Kubernetes pour fournir des fonctionnalités. GitOps
Réconciliation continue	Cet outil compare constamment l'état réel du cluster avec l'état souhaité défini dans Git. Il corrige automatiquement toute dérive détectée entre ces états.
Gestion de plusieurs clusters	Weave GitOps prend en charge la gestion de plusieurs clusters Kubernetes à partir d'un seul plan de contrôle. Il permet un déploieme

Area	Capacités des outils
	nt cohérent des applications dans différents environnements.
La politique en tant que code	Weave GitOps intègre le concept de politique en tant que code permettant d'appliquer les règles de sécurité et de conformité. Les politiques sont contrôlées par les versions, ainsi que le code de l'application et les définitions de l'infrastructure.
Livraison progressive	Cet outil prend en charge les stratégies de déploiement avancées telles que les versions et les blue/green déploiements de Canary. Il s'intègre à Flagger pour une livraison automatis ée et progressive.
Observabilité et tableaux de bord	Weave GitOps fournit des tableaux de bord intégrés pour surveiller l'état des applications et des clusters. Il donne un aperçu des processus de réconciliation et de la santé des clusters.
Sécurisé dès la conception	L'outil met en œuvre les meilleures pratiques de sécurité, notamment l'intégration RBAC et la gestion des secrets. Il prend en charge différent es méthodes d'authentification et s'intègre aux fournisseurs d'identité d'entreprise.
Extensibilité et intégration	L'outil est conçu pour fonctionner avec un large éventail d'outils natifs du cloud. Il prend en charge des outils populaires tels que Flux, Helm et Kustomize.
Plateformes de développement en libre-service	Weave GitOps permet de créer des plateform es en libre-service pour les développeurs. Il fournit des modèles et des garde-fous pour le déploiement des applications.

Area	Capacités des outils
GitOps Automatisation	L'outil automatise de nombreux aspects du GitOps flux de travail, notamment la génération de pull requests pour les mises à jour.
Pipelines de livraison continue	Il s'intègre aux CI/CD systèmes pour créer des pipelines end-to-end de livraison.
Audit et conformité	Weave DevOps fournit une piste d'audit complète de toutes les modifications et actions. Il vous aide à répondre aux exigences de conformité grâce au contrôle des versions et à des processus automatisés.
Evolutivité	L'outil est conçu pour s'adapter aux petits projets aux déploiements de grande envergure destinés aux entreprises.
Collaboration en équipe	Weave GitOps facilite la collaboration entre les équipes de développement et d'exploitation grâce à des flux de travail basés sur Git.
GitOps en tant que service	Cet outil est proposé GitOps sous forme de service géré, ce qui simplifie l'adoption et la gestion.
Support hybride et multicloud	Weave GitOps permet une gestion cohérente entre les différents fournisseurs de cloud et les environnements sur site.
Sécurité continue	L'outil intègre l'analyse de sécurité et l'applica tion des politiques tout au long du processus de déploiement.

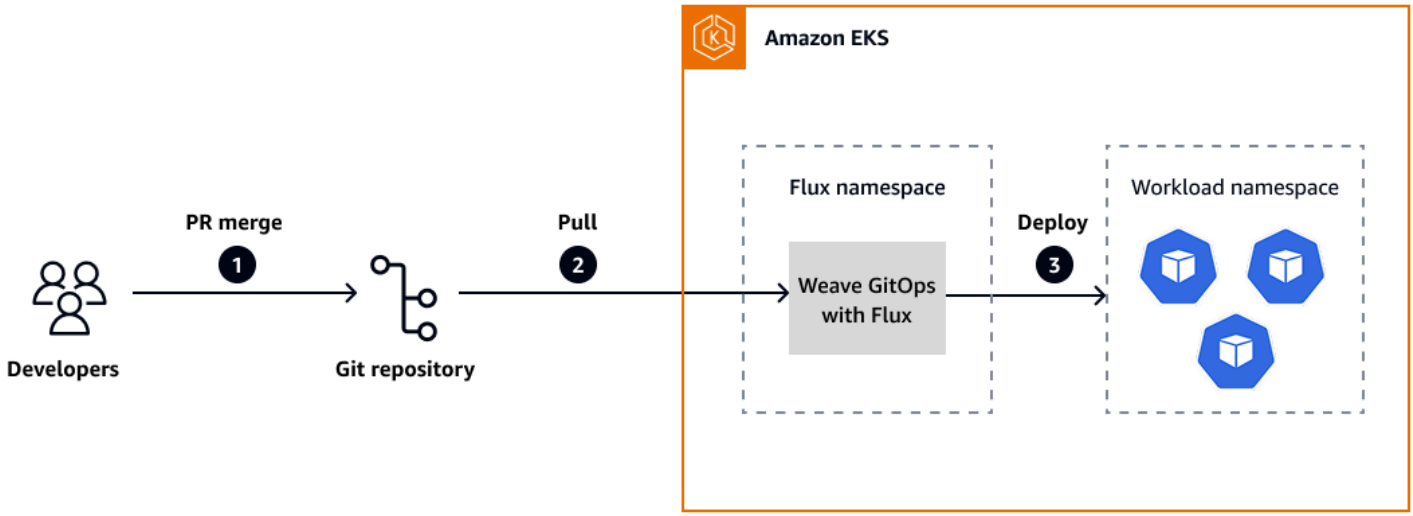
Weave GitOps met en œuvre ces principes pour fournir une GitOps solution complète qui va au-delà de l'automatisation de base du déploiement. Il vise à créer un modèle opérationnel complet pour les applications cloud natives qui met l'accent sur la sécurité, l'évolutivité et la facilité d'utilisation. En

adhérant à ces GitOps principes, Weave GitOps aide les entreprises à gérer de manière cohérente, auditable et efficace leurs environnements Kubernetes sur plusieurs clusters et fournisseurs de cloud.

Pour plus d'informations, consultez la GitOpsdocumentation Weave.

Architecture

Le schéma suivant illustre un flux de travail GitOps sur CD piloté qui utilise Weave GitOps au sein d'un cluster EKS. Pour des informations détaillées, consultez le GitOpsréférentiel Weave.



où :

- Étape 1 : fusion par pull request (PR). Un développeur valide les modifications apportées aux manifestes Kubernetes ou aux graphiques Helm stockés dans un référentiel Git. Lorsque le PR a été révisé et fusionné dans la branche principale, l'état souhaité de l'application est mis à jour dans le contrôle de source.
- Étape 2 : synchronisation du référentiel. Weave GitOps s'exécute dans l'espace de noms Flux du cluster EKS et surveille en permanence le référentiel Git configuré. Lorsqu'il détecte des modifications, il extrait les dernières mises à jour pour réconcilier l'état déclaré.
- Étape 3 : Déploiement vers l'espace de noms cible Weave GitOps compare l'état souhaité dans Git avec l'état réel du cluster. Il applique ensuite les modifications nécessaires à l'espace de noms de charge de travail cible afin que l'application soit déployée ou mise à jour en conséquence.

Jenkins X

Jenkins X est une CI/CD plateforme open source native pour le cloud qui implémente les GitOps principes des environnements Kubernetes. Bien que Jenkins X ne soit pas exclusivement un GitOps outil tel qu'Argo CD ou Flux, il intègre des GitOps pratiques dans ses flux de travail.

GitOps soutien

Area	Capacités des outils
Flux de travail centré sur Git	Jenkins X utilise les référentiels Git comme source principale de vérité pour le code et la configuration des applications. Toutes les modifications apportées aux applications et à l'infrastructure sont effectuées via Git.
L'environnement en tant que code (EAc)	Les environnements (tels que la mise en scène et la production) sont définis sous forme de code dans les référentiels Git. Cela permet de contrôler les versions et de revoir les configura tions de l'environnement.
CI/CD Pipelines automatisés	Jenkins X configure automatiquement des CI/ CD pipelines pour les projets. Ces pipelines sont définis en tant que code (pipeline en tant que code) et stockés dans Git.
Natif de Kubernetes	Jenkins X est spécialement conçu pour les environnements Kubernetes. Il utilise des ressources Kubernetes et des définitions de ressources personnalisées (). CRDs
Environnements de prévisualisation	Jenkins X crée automatiquement des environne ments temporaires pour les pull requests. Il permet de consulter et de tester facilement les modifications avant les fusions.

Area	Capacités des outils
Promotion entre les environnements	Jenkins X utilise une GitOps approche pour promouvoir les applications entre les environnements (par exemple, de la mise en scène à la production). Les promotions sont gérées à l'aide de pull requests afin de garantir des processus de révision et d'approbation appropriés.
Gestion du tableau de bord	Jenkins X utilise des diagrammes Helm pour empaqueter et déployer des applications. Les graphiques sont contrôlés par version dans les référentiels Git.
Gestion automatique des versions	Jenkins X gère automatiquement la gestion des versions des applications et des versions. Il utilise le versionnement sémantique et génère des notes de version.
ChatOps intégration	Jenkins X prend en charge ChatOps les opérations courantes. Cela correspond aux GitOps principes de l'automatisation et de la collaboration.
Extensibilité	Cet outil fournit un système de plugins pour étendre les fonctionnalités. Il permet l'intégration avec divers outils natifs du cloud.
Infrastructure en tant que code (IaC)	Jenkins X prend en charge Terraform, CloudFormation AWS Cloud Development Kit (AWS CDK), et d'autres outils IaC pour définir et gérer l'infrastructure. Les définitions d'infrastructure sont contrôlées par version parallèlement au code de l'application.

Area	Capacités des outils
Annulations automatisées	Jenkins X prend en charge les annulations automatiques si des problèmes sont détectés après le déploiement.
Gestion des secrets	L'outil s'intègre à des solutions externes de gestion des secrets pour traiter les informations sensibles en toute sécurité.
Observabilité	Jenkins X fournit une intégration avec les outils de surveillance et de journalisation pour l'observabilité.
Support multicloud	Jenkins X est conçu pour fonctionner avec différents fournisseurs de cloud et environne ments sur site.
Collaboration en équipe	Cet outil encourage la collaboration grâce à des flux de travail basés sur Git et à des pull requests.
Feedback continu	L'outil fournit des informations rapides sur les modifications par le biais d'environnements de test et de prévisualisation automatisés.
DevOps meilleures pratiques	Jenkins X met en œuvre DevOps les meilleure s pratiques par défaut, y compris GitOps les principes.
Configuration déclarative	L'outil utilise des configurations déclaratives pour définir les applications et les environne ments.
Mises à niveau automatiques	Jenkins X fournit des outils pour automatiser les mises à niveau de la plateforme Jenkins X elle-même.

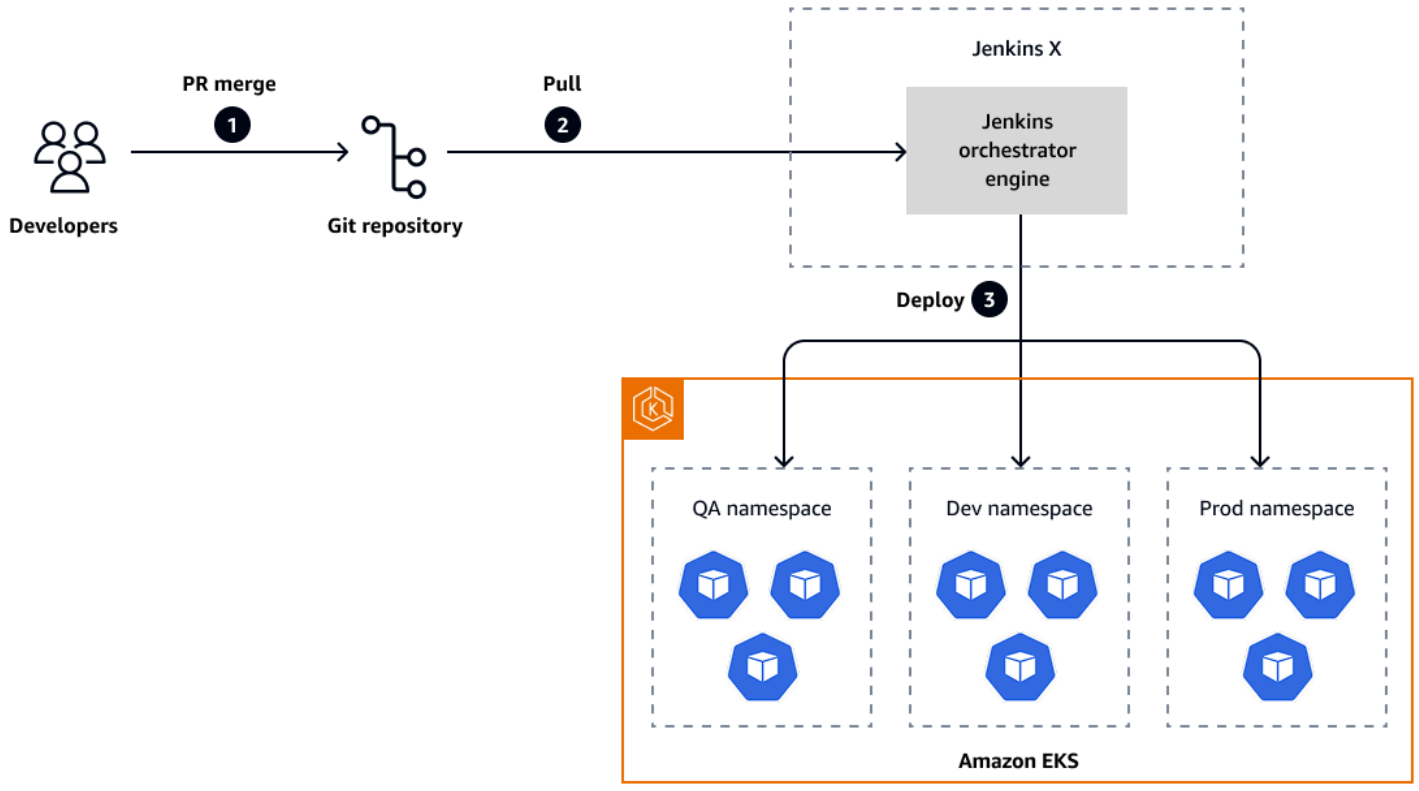
Jenkins X met en œuvre ces GitOps principes pour créer une solution CI/CD complète pour Kubernetes. Il vise à automatiser et à rationaliser l'ensemble du processus de livraison des logiciels, de la validation du code au déploiement en production, tout en respectant les GitOps pratiques. Ce faisant, il aide les équipes à réaliser des déploiements plus rapides, plus fiables et plus cohérents dans des environnements cloud natifs.

La principale différence entre Jenkins X et des outils tels qu'Argo CD ou Flux réside dans le fait que Jenkins X fournit une CI/CD solution plus complète, incluant l'automatisation des builds et la gestion des pipelines, tout en intégrant les GitOps principes de déploiement et de gestion de l'environnement. Cela le rend particulièrement adapté aux équipes qui ont besoin d'une all-in-one solution qui couvre à la fois les aspects CI et CD dans un GitOps cadre unique.

Pour plus d'informations, consultez la documentation de Jenkins X.

Architecture

Le schéma suivant illustre un flux de travail GitOps sur CD piloté par Jenkins X. Pour des informations détaillées, consultez la documentation de Jenkins X.



où :

- Étape 1 : fusion par pull request (PR). Un développeur crée une pull request qui inclut des modifications apportées aux manifestes Kubernetes, aux graphiques Helm ou au code d'application stocké dans un référentiel Git. Après examen et approbation, le PR est fusionné dans la branche principale et met à jour l'état souhaité dans le contrôle de source.
- Étape 2 : synchronisation du référentiel. Jenkins X déclenche automatiquement un CI/CD pipeline lorsqu'il détecte le changement. Le pipeline construit, teste et promeut l'application dans différents environnements (par exemple, la mise en scène et la production) en utilisant des GitOps principes.
- Étape 3 : Déploiement vers des espaces de noms cibles Jenkins X met à jour les référentiels d'environnement (test et production) avec les nouvelles versions de l'application. Le cluster concilie automatiquement les modifications en extrayant les derniers manifestes de Git et en déployant l'application dans les espaces de noms appropriés.

GitLab CI/CD

GitLab CI/CD is an integrated part of the GitLab platform that provides continuous integration, delivery, and deployment capabilities. Although GitLab CI/CDn'est pas exclusivement un GitOps outil, vous pouvez le configurer pour implémenter des GitOps principes, en particulier lorsque vous l'utilisez pour des déploiements Kubernetes.

GitOps soutien

Area	Capacités des outils
Git comme source unique de vérité	GitLab CI/CD utilise les référentiels Git pour stocker à la fois le code de l'application et les configurations d'infrastructure. Toutes les modifications apportées au système sont effectuées via Git, ce qui garantit un historique complet et une piste d'audit.
Configuration déclarative	GitLab Les pipelines CI/CD sont définis dans un fichier .gitlab-ci.yml, qui est une configura tion déclarative stockée dans le référentiel Git. Les manifestes Kubernetes, les diagramme s Helm ou d'autres fichiers d'infrastructure sous forme de code (IaC) peuvent être stockés

Area	Capacités des outils
	dans le même référentiel afin de définir l'état souhaité de l'infrastructure.
Pipelines automatisés	GitLab CI/CD déclenche automatiquement des pipelines lorsque les modifications sont transférées vers le référentiel. Ces pipelines peuvent inclure des étapes de création, de test et de déploiement d'applications.
Intégration avec Kubernetes	GitLab CI/CD fournit une intégration native à Kubernetes et prend en charge GitOps les déploiements de type « C » sur des clusters Kubernetes. Il peut créer et gérer automatiq uement des ressources Kubernetes en fonction de la configuration dans Git.
Gestion de l'environnement	GitLab CI/CD prend en charge la définition de plusieurs environnements (tels que la mise en scène et la production) sous forme de code. Les déploiements dans ces environnements peuvent être automatisés ou nécessiter une approbation manuelle, conformément aux GitOps pratiques en vigueur.
Examiner les demandes	GitLab peut créer automatiquement des environnements temporaires pour les demandes de fusion, similaires aux environne ments de prévisualisation dans d'autres GitOps outils. Cela facilite l'examen et le test des modifications avant les fusions.
Déploiement continu	GitLab Le CI/CD peut être configuré pour déployer automatiquement les modifications apportées aux clusters Kubernetes lorsque les modifications sont fusionnées avec des branches spécifiques.

Area	Capacités des outils
IaC	GitLab CI/CD prend en charge l'intégration avec des outils tels que Terraform et permet de gérer l'infrastructure CloudFormation en tant que code. Les définitions d'infrastructure peuvent être contrôlées par version en même temps que le code de l'application.
Observabilité et surveillance	GitLab Le CI/CD fournit des fonctionnalités intégrées de surveillance et d'observabilité, notamment l'intégration avec Prometheus et Grafana.
Analyse de sécurité	GitLab CI/CD includes built-in security scanning tools that can be integrated into the CI/CDpipe line pour renforcer la sécurité dans le cadre du GitOps flux de travail.
Registre des conteneurs	GitLab Le CI/CD inclut un registre de conteneur s intégré pour une intégration fluide de la gestion des images de conteneurs dans le GitOps flux de travail.
Automatique DevOps	La DevOps fonctionnalité Auto dans les GitLab CI/CD can automatically configure CI/CD pipelines qui suivent les GitOps principes des déploiements de Kubernetes.
Flux de travail d'approbation	GitLab Le CI/CD prend en charge les processus d'approbation des déploiements, qui fournissent des promotions contrôlées entre les environnements.
Gestion des secrets	GitLab CI/CD provides features to securely manage and use secrets within CI/CDpipelines.

Area	Capacités des outils
Versionnage et versions	GitLab CI/CD supports automatic versioning and release management as part of the CI/CDprocessus.
Annulations	GitLab CI/CD permet de revenir facilement aux versions précédentes si des problèmes sont détectés après le déploiement.
Journaux d'audit	GitLab Le CI/CD fournit des journaux d'audit complets pour toutes les actions visant à soutenir l'aspect traçabilité de. GitOps
Pipelines multiprojets	GitLab CI/CD prend en charge des GitOps flux de travail complexes qui s'étendent sur plusieurs projets ou référentiels.
ChatOps	GitLab CI/CD prend en charge les ChatOps intégrations, qui permettent la collaboration et les opérations via des interfaces de chat.
Gestion des clusters Kubernetes	GitLab CI/CD fournit des fonctionnalités permettant de gérer les clusters Kubernetes directement depuis l'interface. GitLab

Cependant GitLab CI/CD is not exclusively designed for GitOps, it can be used effectively to implement GitOps practices, especially for teams that already use GitLab as their primary development platform. Its integrated approach, which combines source control, CI/CD, et la gestion de Kubernetes, en font un outil puissant pour la mise en œuvre de flux de travail. GitOps

La principale différence entre les GitLab CI/CD and dedicated GitOps tools such as Argo CD or Flux is that GitLab provides a more comprehensive platform that includes source control management, issue tracking, and other development tools along with its CI/CD capacités. Cela le rend particulièrement adapté aux équipes qui ont besoin d'une all-in-one solution capable de mettre en œuvre GitOps des pratiques au sein d'un système de développement plus large.

Pour plus d'informations sur le GitLab CI/CD et son architecture, consultez la documentation du GitLab CI/CD.

Spinnaker

Bien que Spinnaker ne soit pas exclusivement conçu comme un GitOps outil, vous pouvez le configurer pour implémenter des GitOps principes, en particulier lorsque vous l'utilisez pour des déploiements cloud natifs et Kubernetes.

GitOps soutien

Area	Capacités des outils
Configuration déclarative	Spinnaker utilise des définitions de pipeline déclaratives, qui sont généralement stockées sous forme de fichiers JSON ou YAML. Ces définitions de pipeline peuvent être contrôlées par version dans les référentiels Git, conformément aux pratiques. GitOps
IaC	Spinnaker prend en charge la définition des configurations d'infrastructure et de déploieme nt sous forme de code. Ces définitions peuvent être stockées dans des référentiels Git et peuvent servir de source unique de vérité.
Déploiements multicloud	Spinnaker est conçu pour fonctionner avec plusieurs fournisseurs de cloud et clusters Kubernetes. Il permet GitOps des pratiques cohérentes dans divers environnements.
Pipeline en tant que code	Les pipelines Spinnaker peuvent être définis sous forme de code et stockés dans des référentiels Git. Cela permet de contrôler les versions et de revoir les processus de déploiement.

Area	Capacités des outils
Déploiements automatisés	Vous pouvez configurer Spinnaker pour démarrer automatiquement les déploieme nts en fonction des modifications apportées aux référentiels Git. L'outil prend en charge les pratiques de déploiement continu qui sont essentielles à GitOps.
Infrastructure immuable	Spinnaker promeut l'utilisation d'une infrastru cture immuable, qui est un concept clé dans. GitOps Il encourage le déploiement de nouvelles instances au lieu de modifier les instances existantes.
Annulations et gestion des versions	Spinnaker fournit des capacités de restaurat ion robustes et un retour rapide aux bons états connus précédents. Il prend en charge le versionnement des déploiements, conformém ent aux GitOps principes de traçabilité.
Flux de travail d'approbation	Spinnaker inclut des étapes de jugement manuel dans les pipelines afin de permettre des promotions contrôlées entre les environne ments. Cela favorise les GitOps pratiques de séparation entre les déploiements et les versions.
Canary et les blue/green déploiements	Spinnaker prend en charge des stratégies de déploiement avancées qui s'alignent sur GitOps les pratiques relatives aux versions sûres et contrôlées.
Intégration aux systèmes de contrôle de version	Spinnaker peut s'intégrer à différents fournisse urs Git pour démarrer des pipelines en fonction des événements du référentiel.

Area	Capacités des outils
Intégration avec Kubernetes	Spinnaker fournit un support natif pour Kubernetes et prend GitOps en charge la gestion de type Kubernetes des ressources Kubernetes.
Gestion des artifacts	Spinnaker prend en charge la gestion des artefacts et le contrôle des versions, qui sont essentiels au maintien d'un flux de travail. GitOps
Observabilité et surveillance	Spinnaker propose une intégration avec des outils de surveillance pour prendre en charge l'aspect observabilité de. GitOps
Piste d'audit	Spinnaker fournit des journaux et un historiqu e de déploiement détaillés, qui soutiennent le principe d'auditabilité de. GitOps
Contrôle d’accès basé sur les rôles (RBAC)	Cet outil implémente le RBAC pour un contrôle précis sur qui peut effectuer quelles actions, conformément aux GitOps pratiques de sécurité.
Modélisation et paramétrage	Spinnaker prend en charge la création de modèles dans les définitions de pipeline afin de permettre des déploiements réutilisables et paramétrés.
Promotion de l'environnement	Spinnaker facilite la promotion des applicati ons entre les environnements (par exemple, de la mise en scène à la production) de manière contrôlée.

Area	Capacités des outils
Intégration aux outils CI	Spinnaker peut s'intégrer à divers outils d'intégration continue (CI) pour fournir un CI/CD pipeline complet conforme aux principes. GitOps
Stages et extensions personnalisés	Cet outil prend en charge les étapes et les extensions personnalisées, afin que les équipes puissent mettre en œuvre des GitOps flux de travail adaptés à leurs besoins.
Gestion centralisée	Spinnaker fournit une plate-forme centralisée pour gérer les déploiements dans plusieurs environnements et fournisseurs de cloud.

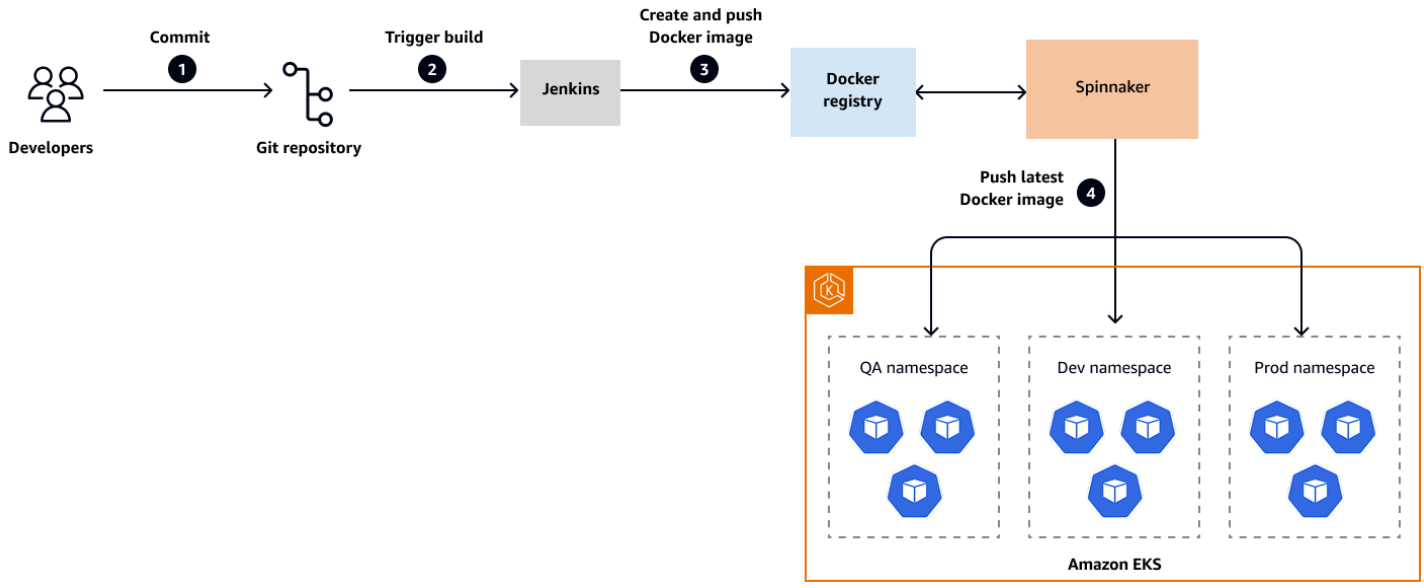
Bien que Spinnaker ne soit pas principalement commercialisé comme un GitOps outil, sa flexibilité et ses fonctionnalités robustes le rendent capable de mettre en œuvre des GitOps flux de travail, en particulier dans des environnements multicloud complexes. La principale différence entre Spinnaker et les GitOps outils dédiés tels qu'Argo CD ou Flux réside dans le fait que Spinnaker propose une plateforme de diffusion continue plus complète avec des stratégies de déploiement avancées et un support multicloud.

La force de Spinnaker réside dans sa capacité à gérer des scénarios de déploiement complexes auprès de différents fournisseurs de cloud et dans sa prise en charge de stratégies de déploiement avancées. Lorsque Spinnaker est correctement configuré, il peut mettre en œuvre les GitOps principes de manière efficace. Cela en fait un outil puissant pour les organisations qui souhaitent adopter GitOps des pratiques dans des environnements divers et complexes.

Pour plus d'informations, consultez la documentation de Spinnaker.

Architecture

Le schéma suivant illustre un flux de travail GitOps sur CD piloté par Spinnaker et Jenkins X. Pour des informations détaillées, consultez la documentation de Spinnaker.



où :

- Étape 1 : validation du code. Les développeurs valident les modifications du code de l'application dans un dépôt Git. Ces modifications peuvent inclure des mises à jour de l'application elle-même, de Dockerfiles ou des manifestes Kubernetes.
- Étape 2 : Construction de Jenkins et création d'images. Jenkins est automatiquement déclenché par le dépôt Git via un webhook ou un sondage. Jenkins construit l'application, crée une image Docker et envoie l'image créée vers un registre Docker configuré (tel qu'Amazon ECR ou Docker Hub).
- Étape 3 : surveillance de l'image Spinnaker et déclenchement du pipeline. Spinnaker surveille en permanence le registre Docker à la recherche de nouvelles images. Lorsqu'une nouvelle version d'image est détectée, Spinnaker déclenche automatiquement un pipeline pour démarrer le processus de déploiement.
- Étape 4 : Déploiement vers des espaces de noms cibles Spinnaker déploie la nouvelle image Docker sur Amazon EKS. Sur la base des configurations du pipeline, l'image est déployée sur les espaces de noms cibles du cluster. Spinnaker veille à ce que la dernière version de l'application soit déployée tout en respectant les stratégies de déploiement définies, telles que les blue/green déploiements Canary.

Flotte d'éleveurs

Rancher Fleet est une GitOps-at-scale solution spécialement conçue pour gérer plusieurs clusters Kubernetes. Il adhère étroitement aux GitOps principes tout en mettant l'accent sur l'évolutivité et la gestion multi-clusters.

GitOps soutien

Area	Capacités des outils
Git comme source unique de vérité	Fleet utilise les référentiels Git comme source officielle pour définir l'état souhaité des applications et des ressources sur plusieurs clusters. Toutes les configurations, y compris les manifestes Kubernetes, les graphiques Helm et les ressources personnalisées, sont stockées dans Git.
Configuration déclarative	Fleet fonctionne avec des descriptions déclaratives de l'état souhaité pour les applications et les ressources. Il peut s'agir de fichiers Kubernetes YAML bruts, de graphiques Helm, de fichiers Kustomize ou de ressources personnalisées spécifiques à Fleet.
Synchronisation automatisée	Fleet surveille en permanence les dépôts Git pour détecter les modifications. Il applique automatiquement les modifications aux clusters cibles lorsqu'il détecte des différences entre l'état Git et l'état du cluster.
Gestion de plusieurs clusters	Fleet est spécialement conçu pour gérer les déploiements sur plusieurs clusters Kubernetes. Il peut gérer des milliers de clusters à partir d'un seul plan de contrôle.
Architecture native de Kubernetes	Fleet est conçu comme un ensemble de ressources et de contrôleurs personnalisés

Area	Capacités des outils
	Kubernetes. Il utilise les mécanismes d'extensi on de Kubernetes pour les opérations. GitOps
Réconciliation continue	Fleet compare constamment l'état réel des clusters avec l'état souhaité défini dans Git. Il corrige automatiquement toute dérive détectée entre ces états.
Regroupement et ciblage de clusters	Fleet vous permet de regrouper des clusters et de cibler les déploiements vers des groupes spécifiques ou des clusters individuels. Il prend en charge le déploiement cohérent des applications dans différents environnements et types de clusters.
Configurations en couches	Fleet prend en charge les configurations en couches, qui fournissent des configurations de base avec des superpositions spécifiques à l'environnement. Cela correspond aux GitOps pratiques de gestion efficace de plusieurs environnements.
Intégration avec Helm	Fleet fournit un support natif pour les cartes Helm et permet de gérer facilement les applicati ons complexes. Il peut versionner et gérer les versions de Helm via des GitOps flux de travail.
Définitions de ressources personnalisées (CRDs)	GitRepo Fleet utilise des ressources personnal isées telles que Bundle pour définir les déploiements. Ils CRDs fournissent un moyen natif de Kubernetes de définir des flux de travail. GitOps

Area	Capacités des outils
Sécurité et RBAC	Fleet s'intègre à Kubernetes RBAC pour le contrôle d'accès. Il prend en charge la gestion sécurisée des informations sensibles et des informations d'identification.
Observabilité	Fleet fournit des informations sur l'état de synchronisation des clusters et des applicati ons. Il fournit des informations sur GitOps les processus de l'ensemble du parc de clusters.
Evolutivité	Fleet est conçu pour évoluer afin de gérer efficacement des milliers de clusters. Il prend en charge les GitOps opérations à grande échelle dans les environnements d'entreprise.
Gestion des dépendances	Vous pouvez définir des dépendances entre différentes ressources et applications. Fleet veille à ce que le bon ordre des opérations soit respecté dans les déploiements complexes.
Personnalisation et extensibilité	Fleet prend en charge les scripts personnalisés et les hooks de cycle de vie pour une personnal isation avancée des déploiements. Il permet l'intégration avec les outils et les flux de travail existants.
Support hors ligne et isolé	La flotte peut fonctionner dans des environne ments où la connectivité Internet est limitée ou inexistante. Il prend en charge les GitOps flux de travail dans des environnements hautement sécurisés ou réglementés.
Déploiements progressifs	Fleet prend en charge les déploiements échelonnés entre les clusters, ce qui permet de mettre en place des stratégies de déploiement contrôlées et progressives.

Area	Capacités des outils
Interface de gestion unifiée	Fleet fournit une interface unique pour gérer les GitOps flux de travail dans tous les clusters. Il simplifie les opérations dans les environne ments complexes à clusters multiples.
Intégration avec d'autres outils Rancher	Fleet s'intègre à d'autres outils Rancher pour fournir une solution complète de gestion Kubernetes.
Piste d'audit et conformité	Fleet conserve une piste d'audit claire de tous les changements et déploiements. Il vous aide à répondre aux exigences de conformité grâce à des opérations Git contrôlées par version.

Rancher Fleet met en œuvre ces GitOps principes en mettant l'accent sur l'évolutivité et la gestion multi-clusters. Sa conception est particulièrement adaptée aux entreprises qui gèrent un grand nombre de clusters Kubernetes dans différents environnements, centres de données ou fournisseurs de cloud.

Le principal facteur de différenciation de Fleet réside dans sa capacité à gérer GitOps à grande échelle. Cette fonctionnalité est particulièrement utile pour les grandes entreprises ou les fournisseurs de services gérés qui gèrent de nombreux clusters. Des outils tels qu'Argo CD ou Flux sont souvent utilisés pour la gestion de clusters individuels, tandis que Fleet est conçu pour gérer GitOps un large parc de clusters.

En respectant ces GitOps principes, Rancher Fleet fournit une solution aux entreprises qui souhaitent mettre en œuvre une gestion cohérente, évolutive et automatisée des applications et des ressources dans un environnement Kubernetes diversifié et à grande échelle.

Pour plus d'informations, consultez la documentation de la flotte.

Architecture

Pour obtenir des informations sur l'architecture et le flux de travail, consultez le référentiel Fleet.

Code Fresh

Codefresh est une CI/CD plateforme moderne qui soutient les GitOps principes, en particulier pour les déploiements de Kubernetes. Codefresh propose un ensemble complet de CI/CD fonctionnalités, et ses GitOps capacités sont remarquables.

GitOps soutien

Area	Capacités des outils
Git comme source unique de vérité	Codefresh utilise les référentiels Git comme source officielle pour le code des applications, les définitions d'infrastructure et les configura tions de pipeline. Toutes les modifications apportées au système sont effectuées via Git, ce qui garantit un historique complet et une piste d'audit.
Configuration déclarative	Codefresh prend en charge les définitio ns de pipeline déclaratives en utilisant des fichiers YAML stockés dans Git. Les manifeste s Kubernetes, les diagrammes Helm, les CloudFormation modèles et autres fichiers IaC peuvent être contrôlés par version dans les mêmes référentiels.
GitOps tableau de bord	Codefresh fournit un tableau de GitOps bord dédié à la visualisation et à la gestion des flux de travail. GitOps Il offre une vue claire de l'état de synchronisation entre les états de Git et du cluster.
Synchronisation automatisée	Codefresh surveille en permanence les dépôts Git pour détecter les modifications. Il démarre automatiquement des pipelines pour appliquer les modifications aux environnements cibles lorsqu'il détecte des différences.

Area	Capacités des outils
Intégration avec Kubernetes	Codefresh offre une intégration approfond ie avec Kubernetes pour prendre en charge GitOps les déploiements de type « type » sur plusieurs clusters. Il prend en charge diverses ressources Kubernetes et des définitions de ressources personnalisées (). CRDs
Gestion de l'environnement	Vous pouvez définir et gérer plusieurs environnements (tels que le développement, la mise en scène et la production) sous forme de code. Codefresh soutient la promotion entre les environnements en utilisant des pratiques. GitOps
Intégration à Argo CD	Codefresh s'intègre à Argo CD pour des fonctionnalités améliorées. GitOps Il combine ses capacités de CI avec les atouts CD d'Argo CD pour fournir une GitOps solution complète.
Support de casque	Codefresh prend en charge les graphique s Helm et permet de gérer facilement des applications complexes grâce à. GitOps Il propose également la gestion des versions et la promotion du graphique Helm.
Livraison progressive	Codefresh prend en charge les stratégies de déploiement avancées telles que Canary et les déploiements. blue/green Vous pouvez mettre en œuvre et gérer ces stratégies par le biais de GitOps flux de travail.
Annulations et gestion des versions	Codefresh permet de revenir facilement aux versions précédentes si des problèmes sont détectés après le déploiement. Il gère le versionnement des déploiements à des fins de traçabilité.

Area	Capacités des outils
Flux de travail d'approbation	Codefresh prend en charge les processus d'approbation manuels et automatisés pour les déploiements. Il permet des promotion s contrôlées entre les environnements, conformément aux GitOps pratiques.
IaC	Codefresh prend en charge l'intégration avec des outils IaC tels CloudFormation que Terraform. Il permet le contrôle de version des définitions d'infrastructure parallèlement au code de l'application.
Observabilité et surveillance	Codefresh fournit des fonctionnalités intégrées de surveillance et d'observabilité. Il propose également des intégrations avec des outils de surveillance externes pour une meilleure visibilit é.
Analyse de sécurité	Codefresh inclut des fonctionnalités d'analyse de sécurité qui peuvent être intégrées aux flux de travail. GitOps Les contrôles de sécurité font partie du processus de déploiement automatis é.
Pistes d'audit	Codefresh tient des journaux d'audit complets pour toutes les actions et modifications. Il prend en charge les aspects de traçabilité et de conformité de GitOps.
RBAC et contrôle d'accès	Codefresh met en œuvre un contrôle d'accès basé sur les rôles (RBAC) pour une gestion précise des autorisations. Cela permet de garantir la sécurité GitOps des opérations au sein des équipes et des environnements.

Area	Capacités des outils
GitOps Automatisation	Codefresh propose des fonctionnalités permettant d'automatiser divers aspects des GitOps flux de travail, notamment la création et la fusion de pull requests (PR).
Déploiements hybrides et multicloud	Codefresh prend en charge les GitOps flux de travail de plusieurs fournisseurs de cloud et d'environnements sur site.
Modélisation et paramétrage	Codefresh prend en charge les modèles dans les configurations de pipeline et de déploieme nt. Cela permet des flux de travail réutilisables et GitOps paramétrés.
Gestion d'image intégrée	Codefresh fournit des fonctionnalités intégrées de gestion des images de conteneur. Il intègre les constructions et les déploiements d'images dans les GitOps flux de travail.
GitOps pour la gestion des secrets	Codefresh propose des moyens sécurisés de gérer les secrets dans les flux de travail. GitOps Il s'intègre aux solutions externes de gestion des secrets.
Fonctionnalités de collaboration	Codefresh fournit des fonctionnalités pour la collaboration en équipe au sein des processus . GitOps Ces fonctionnalités incluent les commentaires, les notifications et les tableaux de bord partagés.

L'approche Codefresh GitOps se distingue par son intégration des capacités CI/CD aux pratiques. GitOps Il vise à fournir une plate-forme complète qui couvre l'ensemble du cycle de vie de livraison des logiciels tout en respectant les GitOps principes.

Le principal facteur de différenciation de Codefresh dans ce GitOps domaine est son approche de plate-forme unifiée, qui combine les capacités de CI avec le CD et les fonctionnalités. GitOps Cela

le rend particulièrement adapté aux équipes qui recherchent une all-in-one solution capable de gérer des CI/CD scénarios complexes tout en mettant en œuvre GitOps des pratiques.

Codefresh propose une plate-forme aux organisations qui souhaitent adopter des GitOps méthodologies dans un CI/CD contexte plus large, en particulier lorsqu'elles travaillent avec Kubernetes et des technologies natives du cloud.

Pour plus d'informations, consultez la documentation Codefresh.

Pulumi

Pulumi est une plateforme IaC qui n'est pas exclusivement conçue pour GitOps. Cependant, il peut être utilisé efficacement pour mettre en œuvre des GitOps principes, en particulier pour l'infrastructure cloud et les déploiements de Kubernetes.

GitOps soutien

Area	Capacités des outils
IaC	Pulumi vous permet de définir votre infrastru cture en utilisant des langages de programma tion polyvalents tels que Python et Go TypeScript. Cette approche basée sur le code s'inscrit dans le cadre de l' GitOps accent mis sur les configurations déclaratives versionnées.
Git comme source unique de vérité	Le code d'infrastructure de Pulumi peut être stocké et contrôlé par version dans des référentiels Git. Cela garantit que Git est la source unique de vérité pour les définitions d'infrastructure.
État souhaité déclaratif	Bien que Pulumi utilise des langages de programmation, il décrit toujours l'état souhaité de l'infrastructure de manière déclarative. Le code définit à quoi doit ressembler l'infrast ructure, et non le step-by-step processus de création de celle-ci.

Area	Capacités des outils
Synchronisation automatisée	Pulumi peut être intégré à des CI/CD pipelines pour appliquer automatiquement les modificat ions lorsque le code est mis à jour dans Git. Cela permet le déploiement continu des modifications de l'infrastructure, ce qui est un GitOps principe clé.
Support multi-cloud et Kubernetes	Pulumi prend en charge un large éventail de fournisseurs de cloud et Kubernetes, afin que vous puissiez suivre les GitOps pratiques dans divers environnements. L'outil permet une gestion cohérente des ressources sur différent es plateformes.
Gestion des états	Pulumi gère l'état de l'infrastructure, qui peut être stockée à distance et en toute sécurité. Cette gestion de l'état est cruciale pour GitOps les pratiques et garantit la cohérence entre l'état défini et l'état réel de votre infrastructure.
Détection des dérives et réconciliation	Pulumi peut détecter les différences entre l'état souhaité (dans le code) et l'état réel de l'infrastructure. Il concilie ces différences conformément au GitOps principe de réconcili ation continue.
La politique en tant que code	Vous pouvez définir et appliquer des politique s sous forme de code à l'aide de Pulumi CrossGuard. Cela permet une gestion des politiques de conformité et de sécurité de GitOps type contrôlé par version.

Area	Capacités des outils
Gestion des secrets	Pulumi fournit des moyens sécurisés de gérer les informations sensibles dans le code de l'infrastructure. Il prend en charge l'intégration avec des systèmes de gestion des secrets externes, ce qui est crucial pour les pratiques GitOps de sécurité.
Composants modulaires et réutilisables	Pulumi soutient la création de composants et de modules réutilisables. Cette modularité s'aligne sur les GitOps pratiques de gestion de déploiements complexes dans plusieurs environnements.
Aperçu et planification	Pulumi offre la possibilité de prévisualiser les modifications avant de les appliquer. Cela soutient le GitOps principe de modifications sûres et prévisibles de l'infrastructure.
Annulations et historique	Pulumi conserve un historique des déploieme nts et prend en charge les annulations vers les états précédents. Cela est conforme GitOps aux principes de traçabilité et de réversibilité.
Livraison continue pour l'infrastructure	Pulumi peut être intégré dans des CI/CD pipelines pour une mise en œuvre continue des modifications de l'infrastructure. Il prend en charge les tests automatisés et la validation du code d'infrastructure.
RBAC et contrôle d'accès	Pulumi fournit un contrôle d'accès basé sur les rôles pour gérer les personnes autorisées à apporter des modifications à l'infrastructure. Cela soutient les pratiques GitOps de sécurité et de gouvernance.

Area	Capacités des outils
Observabilité et journalisation	Pulumi propose des fonctionnalités de journalisation et de surveillance pour les modifications de l'infrastructure. Ces capacités soutiennent l'aspect observabilité des GitOps pratiques.
Intégration avec d'autres outils	Pulumi peut s'intégrer à différents outils dans le cloud. Cette flexibilité permet des GitOps flux de travail complets.
Gestion de l'environnement	Pulumi prend en charge la gestion de plusieurs environnements (développement, mise en scène, production) en utilisant la même base de code avec différentes configurations. Cela correspond aux GitOps pratiques de gestion cohérente de plusieurs environnements.
Gestion des dépendances	Pulumi gère les dépendances entre les ressources et garantit le bon ordre des opérations. Cela est crucial pour les GitOps déploiements complexes impliquant des composants interdépendants.
Fournisseurs de ressources personnalisés	Pulumi vous permet de créer des fournisseurs personnalisés pour gérer n'importe quel service piloté par API. Cela étend GitOps les pratiques à un large éventail de ressources au-delà des offres cloud standard.
Fonctionnalités de collaboration	Pulumi soutient la collaboration en équipe grâce à des contrôles d'état et d'accès partagés. Cela facilite GitOps les flux de travail dans les environnements d'équipe.

En utilisant ces fonctionnalités de Pulumi, les entreprises peuvent mettre en œuvre GitOps des pratiques pour leur infrastructure, en particulier dans les scénarios où elles ont besoin d'un contrôle

précis ou d'une logique complexe, ou lorsqu'elles souhaitent gérer un ensemble diversifié de ressources cloud et sur site au sein d'un cadre unique et cohérent.

L'approche de Pulumi GitOps est unique car elle apporte la puissance et la flexibilité des langages de programmation à usage général à la gestion de l'infrastructure tout en respectant les principes. GitOps Cela peut être particulièrement avantageux pour les équipes qui préfèrent travailler avec des langages de programmation familiers et qui souhaitent appliquer les meilleures pratiques d'ingénierie logicielle à la gestion de l'infrastructure.

Le principal facteur de différenciation de Pulumi GitOps réside dans son utilisation de langages de programmation standard pour définir l'infrastructure. Les GitOps outils traditionnels utilisent souvent le YAML ou des langages spécifiques à un domaine, tandis que Pulumi permet une logique plus complexe, une meilleure réutilisation du code et une intégration plus facile aux flux de travail de développement existants.

Pour plus d'informations, consultez la documentation de Pulumi.

GitOps comparaison des outils

Voici une comparaison des neuf GitOps outils dont il a été question dans les sections précédentes. Lorsque vous choisissez un outil, tenez compte de vos exigences spécifiques, de l'infrastructure existante, de l'expertise de votre équipe et du niveau de contrôle et de personnalisation souhaité.

Facilité d'utilisation

- Argo CD, Flux et Rancher Fleet sont généralement plus faciles à configurer.
- Spinnaker et Jenkins X ont des courbes d'apprentissage plus abruptes.
- Weave GitOps peut nécessiter une configuration supplémentaire pour les fonctionnalités avancées.
- GitLab CI/CD et Codefresh proposent des expériences intégrées.

Intégration avec Kubernetes

- Argo CD, Flux et Rancher Fleet sont très centrés sur Kubernetes.
- Jenkins X et Weave GitOps offrent des fonctionnalités plus étendues DevOps .
- Les autres outils prennent en charge Kubernetes sans se concentrer exclusivement sur celui-ci.

Capacités CI/CD

- Jenkins X, GitLab CI/CD, and Codefresh offer complete CI/CD des solutions.
- Argo CD, Flux et Weave GitOps se concentrent davantage sur l'aspect CD du flux de travail et nécessitent souvent une intégration avec des outils CI distincts.

GitOps pureté

- Argo CD et Flux sont des outils qui se concentrent spécifiquement sur GitOps.
- Les autres outils intègrent des GitOps principes à des degrés divers.

Support multicloud

- Spinnaker et Pulumi excellent dans les scénarios multicloud.
- Les autres outils peuvent fonctionner sur différents clouds, mais peuvent nécessiter une configuration supplémentaire.

Prise en charge de plusieurs clusters

- Tous les outils prennent en charge les déploiements multi-clusters.
- Argo CD et Weave GitOps disposent de fonctionnalités de gestion multi-clusters plus avancées.

Integration

- Flux bénéficie du solide soutien de la Cloud Native Computing Foundation (CNCF).
- Argo CD possède une communauté importante et active.
- Argo CD et Flux intègrent parfaitement Kubernetes.
- Jenkins X utilise le système Jenkins au sens large.
- Weave GitOps est une entreprise plus récente, mais elle se développe grâce à un solide soutien commercial.
- GitLab CI/CD s'intègre étroitement à. GitLab
- Rancher Fleet fonctionne bien dans le système Rancher.

Communauté et soutien

- Flux bénéficie d'un solide soutien de la CNCF.
- Argo CD et Spinnaker ont de grandes communautés. GitLab
- Un support commercial est disponible pour la plupart des outils.

Fonctionnalités d'entreprise

- Weave GitOps et Jenkins X proposent par défaut davantage de fonctionnalités axées sur les entreprises.

- Argo CD et Flux proposent des offres d'entreprise ou peuvent être étendus pour une utilisation en entreprise.

Flexibilité et extensibilité

- Flux est hautement modulaire et extensible.
- Argo CD offre de bonnes options de personnalisation.
- Jenkins X est très extensible mais peut nécessiter plus d'efforts.
- Weave GitOps vise à fournir une solution complète nécessitant moins d'extensibilité.

Evolutivité

- Spinnaker et GitLab CI/CD sont réputés pour leur évolutivité en entreprise.
- Argo CD et Flux gèrent bien les déploiements Kubernetes à grande échelle.

Gestion de l'infrastructure

- Pulumi se concentre sur la gestion des infrastructures.
- Weave GitOps et Flux offrent de bonnes capacités IaC.

Support du modèle de programmation et du langage

- Dans Pulumi, vous pouvez définir une infrastructure en utilisant des langages de programmation polyvalents tels que Python, Go TypeScript, C# et Java. L'utilisation par Pulumi de langages standard permet d'intégrer le code d'infrastructure aux flux de travail de développement habituels, aux pratiques de test et à une logique complexe.
- Terraform utilise le langage HashiCorp de configuration (HCL).
- CloudFormation utilise des modèles JSON et YAML.
- Argo CD, Flux, Rancher Fleet, Weave GitOps, Spinnaker et GitLab CI/CD gèrent principalement les fichiers de configuration YAML ou déclaratifs.
- Jenkins X gère le YAML et les pipelines basés sur des scripts, mais ne propose pas nativement de programmation générale pour IaC.

Cas d'utilisation d'Argo CD et de Flux

Cette section se concentre sur deux outils, Argo CD et Flux, qui fournissent des GitOps fonctionnalités pures. Dans ce contexte, pure GitOps fait référence à un modèle dans lequel un référentiel Git sert de source fiable unique pour l'état souhaité des applications et de l'infrastructure. Toutes les modifications sont effectuées par le biais de validations Git, et le système synchronise automatiquement l'environnement réel pour qu'il corresponde à l'état défini dans le référentiel. Aucune intervention manuelle n'est requise en dehors des opérations Git.

Considérations d’ordre général

- Vous préférerez peut-être utiliser Argo CD dans des environnements où la gestion visuelle et les flux de travail centrés sur les applications sont importants.
- Vous pouvez choisir Flux si vous avez besoin de solutions plus légères, d'une mutualisation robuste ou d'une intégration approfondie avec le réseau plus large des Cloud Native Computing Foundations (CNCF).
- Argo CD plaît souvent aux équipes qui passent du CI/CD traditionnel à son interface utilisateur GitOps intuitive.
- Le flux est souvent privilégié dans les environnements cloud natifs où les flux de travail basés sur la CLI et les pratiques IaC sont déjà établis.

En fin de compte, le choix entre Argo CD et Flux dépend souvent des besoins spécifiques de votre organisation, des outils existants et des préférences de l'équipe. Les deux outils sont capables de gérer la plupart des GitOps scénarios. Nous vous recommandons donc de les évaluer en fonction de vos cas d'utilisation et de vos exigences spécifiques.

Cas d'utilisation d'Argo CD

Management visuel :

- Lorsque vous avez besoin d'une interface utilisateur conviviale pour gérer les déploiements et visualiser l'état des applications.
- Pour les équipes qui préfèrent une interface graphique pour la surveillance et le dépannage.

Approche centrée sur les applications :

- Lorsque vous souhaitez gérer des déploiements au niveau de l'application plutôt que de gérer des ressources individuelles.
- Pour les entreprises qui structurent leurs déploiements autour de concepts d'applications.

Gestion de plusieurs clusters :

- Lorsque la gestion des déploiements sur plusieurs clusters est une exigence essentielle.
- Pour les environnements distribués complexes comportant de nombreux clusters.

Annulation et synchronisation des vagues :

- Lorsque vous avez besoin d'un contrôle précis du processus de déploiement, y compris les ondes de synchronisation et les interventions manuelles.
- Pour les scénarios nécessitant des stratégies de réduction complexes.

Intégration avec les outils existants :

- Lorsque vous utilisez déjà d'autres outils dans le projet Argo, tels que Argo Workflows et Argo Events.

Environnements d'entreprise :

- Pour les grandes entreprises qui ont besoin d'une intégration RBAC robuste et d'une authentification unique par défaut.

Cas d'utilisation de Flux

Déploiements légers :

- Lorsque vous avez besoin d'une solution plus légère et moins gourmande en ressources GitOps.
- Pour les scénarios d'informatique de pointe ou d'IoT où les ressources peuvent être limitées.

Mises à jour automatisées des images :

- Lorsque la détection automatique et le déploiement de nouvelles images de conteneurs constituent une exigence essentielle.

- Pour les équipes qui se concentrent sur le déploiement continu avec des mises à jour fréquentes des images.

Multi-location :

- Lorsqu'un support multi-tenant solide est nécessaire, en particulier dans les environnements de clusters partagés.
- Pour les fournisseurs de services ou les grandes organisations qui ont des séparations strictes entre les équipes ou les projets.

IAC :

- Lorsque vous gérez à la fois les applications et l'infrastructure via le même GitOps flux de travail, il est important.
- Pour les équipes fortement investies dans le paradigme IaC.

Intégration au casque :

- Lorsque l'utilisation intensive des diagrammes Helm fait partie de votre stratégie de déploiement.
- Pour les environnements comportant des déploiements complexes basés sur HELM.

Intégration du projet CNCF :

- Lorsqu'une intégration étroite avec d'autres projets de la CNCF est importante.
- Pour les organisations qui s'alignent sur les technologies et les principes de la CNCF.

Architecture modulaire :

- Lorsque vous avez besoin de flexibilité pour utiliser uniquement des composants spécifiques de la GitOps boîte à outils.
- Pour les équipes qui souhaitent créer des GitOps flux de travail personnalisés à l'aide de composants modulaires.

Livraison progressive :

- Lorsque des stratégies de déploiement avancées telles que les versions ou les A/B tests de Canary sont des exigences fondamentales.

Comparaison des fonctionnalités

Area	Argo CD	Flux
Support aux GitOps principes fondamentaux	 Oui	 Oui
Architecture	End-to-end application pour implémenter des flux de travail Kubernetes GitOps	Fournit Kubernetes CRDs et des contrôleurs pour GitOps
Configuration	Simplicité	Complexe
Support de casque	 Oui	 Oui
Personnaliser le support	 Oui	 Oui
Interface graphique intégrée	CLI et interface utilisateur Web complète	CLI et interface Web légère en option
Support RBAC	Contrôle granulaire	RBAC natif de Kubernetes
Support multi-locataires et multi-clusters	Excellente prise en charge des clusters multiples	Excellente prise en charge pour les locations multiples
Authentification par connexion unique	 Oui	 Oui

Area	Argo CD	Flux
Automatisation de la synchronisation	Possibilité de synchroniser les fenêtres	Possibilité de définir des intervalles de réconciliation
Synchronisation partielle	 Oui	 Non
Processus de réconciliation	Supporte les synchronisations manuelles et automatiques. Plusieurs stratégies différentes sont disponibles.	Supporte les synchronisations manuelles et automatiques.
Extensibilité	Supporte les plugins personnalisés. Options de personnalisation limitées.	Supporte le contrôleur personnalisé. Bonne extensibilité et intégrations tierces.
Soutien communautaire	Une communauté vaste et active.	Communauté plus petite mais en pleine croissance.
Evolutivité	Bonne évolutivité, mais limitée par le taux de récupération de données de l'interface utilisateur Web. L'analyse communautaire suggère que des dizaines de milliers de demandes sont prises en charge.	Des guides clairs pour une évolutivité horizontale et verticale, jusqu'à des dizaines de milliers d'applications.

Bonnes pratiques pour choisir un GitOps outil

Cette section fournit des considérations, des conseils et les meilleures pratiques pour choisir un GitOps outil pour votre cluster EKS. Le bon choix dépend de votre contexte spécifique, de vos exigences et de votre stratégie à long terme. Il est souvent avantageux de réaliser une preuve de concept avec vos meilleurs choix avant de prendre une décision finale.

Évaluez les besoins et les capacités de votre organisation :

- Tenez compte des compétences actuelles de votre équipe et de sa volonté d'apprendre de nouveaux outils.
- Évaluez la complexité de votre environnement Amazon EKS. (Par exemple, utilisez-vous un ou plusieurs clusters ?)
- Déterminez vos exigences spécifiques en matière de conformité, de sécurité et d'évolutivité.

> **Bonne pratique**
>
> Créez un document d'exigences détaillé qui décrit les fonctionnalités requises et les fonctionnalités utiles, mais non obligatoires.

Évaluez la maturité et l'adoption des outils :

- Étudiez la maturité des GitOps outils potentiels et leur taux d'adoption dans le secteur.
- Recherchez des outils qui ont fait leurs preuves dans les environnements Amazon EKS.

> **Bonne pratique**
>
> Priorisez les outils largement adoptés et fortement présents dans le réseau de la Cloud Native Computing Foundation (CNCF).

Envisagez l'intégration à votre chaîne d'outils existante :

- Évaluez dans quelle mesure l' GitOps outil s'intègre à votre CI/CD pipeline actuel, à vos solutions de surveillance et à d'autres outils opérationnels.
- Recherchez des intégrations natives Services AWS telles que IAM, Amazon ECR et. CloudWatch

> **Bonne pratique**
>
> Créez une preuve de concept pour tester les capacités d'intégration avant de prendre une décision finale.

Évaluez les fonctionnalités de sécurité :

- Priorisez les outils dotés de solides fonctionnalités de contrôle d'accès basé sur les rôles (RBAC) et qui s'intègrent bien à l'IAM.
- Recherchez des fonctionnalités qui prennent en charge la gestion sécurisée des secrets et l'application des politiques.

> **Bonne pratique**
>
> Choisissez un outil qui prend en charge les pratiques de sécurité GitOps basées, notamment les politiques sous forme de code et les contrôles de conformité automatisés.

Évaluez l'évolutivité et les performances :

- Examinez les performances de l'outil avec un grand nombre d'applications et de clusters.
- Évaluez son impact sur les performances du cluster et la consommation de ressources.

> **Bonne pratique**
>
> Effectuez des tests de performance avec des charges de travail similaires à celles de votre environnement de production pour vous assurer que l'outil peut gérer votre échelle.

Envisagez la prise en charge de plusieurs clusters et environnements :

- Si vous possédez ou prévoyez d'avoir plusieurs clusters EKS, privilégiez les outils dotés de solides capacités de gestion multi-clusters.
- Recherchez des fonctionnalités qui prennent en charge des déploiements cohérents dans différents environnements (tels que le développement, la mise en scène et la production).

> **Bonne pratique**
>
> Choisissez un outil qui permet la gestion centralisée de plusieurs clusters tout en conservant des configurations spécifiques à l'environnement.

Évaluez les capacités d'observabilité et de surveillance :

- Recherchez des outils offrant une visibilité claire sur l'état de vos déploiements et sur l'état de santé de votre cluster.
- Déterminez dans quelle mesure l'outil s'intègre à vos solutions de surveillance et de journalisation existantes.

> **Bonne pratique**
>
> Priorisez les outils qui proposent des tableaux de bord personnalisables et des mécanismes d'alerte pour une détection proactive des problèmes.

Évaluez la courbe d'apprentissage et la documentation :

- Évaluez la qualité et l'exhaustivité de la documentation de l'outil.
- Tenez compte de la disponibilité des ressources de formation et du soutien communautaire.

> **Bonne pratique**
>
> Choisissez un outil doté d'une documentation bien tenue, de forums communautaires actifs et de programmes de formation ou de certifications officiels.

Tenez compte des coûts et de l'utilisation des ressources :

- Évaluez à la fois les coûts directs (tels que les licences et le support) et les coûts indirects (tels que les frais généraux d'exploitation et les coûts de formation) liés à l'adoption de l'outil.
- Évaluez l'efficacité de l'outil en termes de consommation de ressources de calcul et de stockage.

> **Bonne pratique**
>
> Réalisez une analyse du coût total de possession (TCO) qui inclut les coûts à court et à long terme.

Évaluez la flexibilité et les options de personnalisation :

- Recherchez des outils qui vous permettent de personnaliser les flux de travail en fonction de vos besoins spécifiques.
- Tenez compte de l'extensibilité de l'outil via des plugins ou APIs.

> **Bonne pratique**
>
> Choisissez un outil qui équilibre les fonctionnalités par défaut avec la possibilité de le personnaliser en fonction de vos besoins uniques.

Évaluez les capacités de livraison continue et de déploiement progressif :

- Recherchez des outils qui prennent en charge les stratégies de déploiement avancées, telles que les versions et les blue/green déploiements de Canary.
- Évaluez la facilité de mise en œuvre et de gestion de ces stratégies.

> **Bonne pratique**
>
> Priorisez les outils qui offrent un support intégré pour les modèles de livraison progressifs afin de minimiser les risques liés à vos déploiements.

Pensez à la dépendance à un fournisseur et à la portabilité :

- Évaluez les dépendances de l'outil vis-à-vis de fournisseurs ou de technologies cloud spécifiques.
- Pensez à la facilité de migration vers un autre outil à l'avenir, si nécessaire.

> **Bonne pratique**
>
> Privilégiez les outils qui utilisent des standards ouverts et fournissent des fonctionnalités d'exportation pour vos GitOps configurations.

Évaluez le soutien et les extensions de la communauté :

- Examinez la taille et l'activité de la communauté d'utilisateurs.
- Évaluez la disponibilité des intégrations et des plugins tiers.

> **Bonne pratique**
>
> Rejoignez des forums communautaires ou des groupes d'utilisateurs pour bénéficier d'expériences directes auprès d'autres utilisateurs avant de prendre une décision.

Tenez compte des exigences de conformité et d'audit :

- Évaluez dans quelle mesure l'outil répond à vos besoins de conformité, y compris les pistes d'audit et les rapports.
- Recherchez les fonctionnalités qui aident à maintenir et à démontrer la conformité.

> **Bonne pratique**
>
> Choisissez un outil qui fournit des journaux d'audit complets et prend en charge la génération de rapports de conformité.

Évaluez les capacités de restauration et de reprise après sinistre :

- Évaluez la facilité et la fiabilité des mécanismes de réduction.
- Déterminez comment l'outil prend en charge les scénarios de reprise après sinistre.

Bonne pratique

Testez minutieusement les processus de restauration et de restauration dans le cadre de votre évaluation.

FAQ

Q : Quels sont les GitOps outils les plus populaires pour Amazon EKS ?

R : Les GitOps outils les plus populaires pour Amazon EKS incluent Argo CD, Flux, Jenkins X et GitLab CI/CD. Chaque outil a ses points forts, mais Argo CD et Flux sont particulièrement appréciés pour leur approche native de Kubernetes et leur solide soutien communautaire.

Q : Comment GitOps améliorer la gestion des clusters EKS ?

R : GitOps améliore la gestion des clusters EKS en fournissant un contrôle de version pour l'infrastructure, des déploiements automatisés, une sécurité améliorée grâce à des configurations déclaratives, des annulations simplifiées et une meilleure auditabilité. Il améliore également la collaboration et réduit les erreurs humaines lors des déploiements.

Q : Quelles fonctionnalités clés dois-je rechercher dans un GitOps outil pour Amazon EKS ?

R : Les principales fonctionnalités à rechercher incluent : une intégration fluide avec Amazon EKS, un RBAC robuste, la prise en charge de plusieurs clusters, des fonctionnalités d'observabilité, la prise en charge de stratégies de livraison progressives, l'évolutivité et l'intégration avec des solutions Services AWS telles que IAM et Amazon ECR.

Q : Comment garantir la sécurité lors de la mise en œuvre GitOps dans Amazon EKS ?

R : Pour garantir la sécurité, choisissez un outil doté d'une forte intégration RBAC avec IAM, d'une gestion sécurisée des secrets, de la prise en charge des référentiels Git chiffrés et de la capacité d'implémenter des politiques de sécurité sous forme de code. Vérifiez également que l'outil fournit des journaux d'audit complets.

Q : Les GitOps outils peuvent-ils gérer des environnements Amazon EKS à clusters multiples ?

R : Oui, GitOps des outils tels qu'Argo CD et Flux possèdent de solides capacités de gestion multi-clusters. Ils vous permettent de gérer plusieurs clusters EKS à partir d'un seul plan de contrôle, ce qui garantit la cohérence entre les environnements.

Q : Comment les GitOps outils s'intégrent-ils aux pipelines CI/CD existants ?

R : GitOps les outils s'intègrent généralement aux pipelines CI/CD existants en agissant comme phase de déploiement du pipeline. Ils peuvent être déclenchés par les outils CI lorsque des

modifications sont apportées au référentiel Git, et ils automatisent le processus de déploiement vers les clusters EKS.

Q : Quels sont les défis liés à la mise en œuvre GitOps dans Amazon EKS ?

R : Les défis courants incluent la gestion sécurisée des secrets, la garantie de contrôles d'accès appropriés, la gestion des applications dynamiques, la gestion du décalage entre Git et l'état du cluster et l'adaptation des flux de travail d'équipe au GitOps modèle.

Q : Comment les GitOps outils gèrent-ils les annulations dans Amazon EKS ?

R : GitOps les outils gèrent généralement les annulations en revenant à un commit précédent dans le référentiel Git. Cela déclenche automatiquement le déploiement du bon état connu précédent, ce qui se traduit par des annulations rapides et fiables.

Q : Les GitOps outils peuvent-ils gérer les modules complémentaires Amazon EKS et d'autres AWS ressources ?

R : De nombreux GitOps outils peuvent gérer les modules complémentaires Amazon EKS et certaines AWS ressources, en particulier lorsqu'ils sont combinés à des outils IaC tels que Terraform ou. CloudFormation Toutefois, l'étendue de cette fonctionnalité peut varier ; consultez la section sur les GitOps outils pour obtenir des informations spécifiques sur chaque outil.

Q : Comment les GitOps outils répondent-ils aux exigences de conformité dans Amazon EKS ?

R : GitOps les outils favorisent la conformité en fournissant une piste d'audit claire de toutes les modifications, en appliquant les processus d'approbation, en mettant en œuvre la politique sous forme de code pour les contrôles de conformité automatisés et en proposant des fonctionnalités de journalisation et de reporting détaillées.

Q : Quelle est la courbe d'apprentissage nécessaire à la mise en œuvre GitOps dans Amazon EKS ?

R : La courbe d'apprentissage peut varier en fonction de l'outil et des connaissances existantes de votre équipe. En général, les équipes familiarisées avec Git, Kubernetes et Amazon EKS s'adapteront plus rapidement que les autres. Les outils les plus populaires proposent une documentation complète et des ressources de formation pour faciliter l'adoption.

Q : Comment les GitOps outils gèrent-ils la gestion des secrets dans Amazon EKS ?

R : GitOps les outils s'intègrent généralement à des solutions de gestion des secrets externes telles que AWS Secrets Manager HashiCorp Vault. Certains outils proposent également un chiffrement intégré pour les secrets stockés dans les référentiels Git.

Q : Les GitOps outils peuvent-ils fonctionner à la fois avec des applications statiques et statiques dans Amazon EKS ?

R : Oui, les GitOps outils peuvent fonctionner à la fois avec des applications avec ou sans état. Cependant, la gestion des applications dynamiques nécessite souvent des considérations supplémentaires, telles que la gestion des volumes persistants et la garantie de la cohérence des données lors des mises à jour.

Q : Comment les GitOps outils sont-ils compatibles avec Canary ou blue/green les déploiements dans Amazon EKS ?

R : De nombreux GitOps outils offrent un support intégré pour les stratégies de déploiement avancées. Ils peuvent gérer le déploiement progressif des nouvelles versions, surveiller les problèmes et revenir automatiquement en arrière si des problèmes sont détectés. Toutes ces opérations sont définies sous forme de code dans le référentiel Git.

Q : Quelle est la différence entre l'utilisation d'un GitOps outil et l'utilisation **`kubectl apply`** d'un CI/CD pipeline ?

R : les GitOps outils offrent des avantages par rapport aux `kubectl apply` commandes simples, notamment la détection automatique des dérives et la réconciliation, une sécurité améliorée grâce à des déploiements basés sur des pulls, une meilleure auditabilité et des stratégies de déploiement plus sophistiquées. Ils fournissent également une approche plus complète pour gérer l'état complet du cluster.

Ressources

Les ressources suivantes fournissent de la documentation officielle, des guides pratiques, des études de cas et des analyses approfondies qui peuvent vous aider à prendre une décision éclairée lors du choix d'un GitOps outil pour votre cluster EKS. Ils couvrent divers aspects GitOps, notamment les stratégies de mise en œuvre, les meilleures pratiques, les comparaisons entre différents outils et les expériences du monde réel.

AWS ressources :

- Documentation Amazon EKS
- Automatiser Amazon EKS avec GitOps (article de AWS blog)
- Présentation d' GitOps EKS avec Weaveworks (atelier)AWS
- Flux lab (atelier Amazon EKS)
- Laboratoire Argo CD (atelier Amazon EKS)

GitOps et documentation de l'outil :

- GitOps Meilleures pratiques pour un déploiement continu et une sécurité progressive (webinaire DevOps .com à la demande)
- Documentation Kubernetes
- Documentation sur Argo CD
- Documentation sur les flux
- Documentation sur le tissage GitOps
- Documentation de Jenkins X
- GitLab CI/CD documentation
- Documentation Spinnaker
- Documentation sur la flotte de ranchers
- Documentation Codefresh

Historique du document

Le tableau suivant décrit les modifications importantes apportées à ce guide. Pour être averti des mises à jour à venir, abonnez-vous à un fil RSS.

Modification	Description	Date
Publication initiale	—	30 avril 2025

AWS Glossaire des directives prescriptives

Les termes suivants sont couramment utilisés dans les stratégies, les guides et les modèles fournis par les directives AWS prescriptives. Pour suggérer des entrées, veuillez utiliser le lien Faire un commentaire à la fin du glossaire.

Nombres

7 R

Sept politiques de migration courantes pour transférer des applications vers le cloud. Ces politiques s'appuient sur les 5 R identifiés par Gartner en 2011 et sont les suivantes :

- Refactorisation/réarchitecture : transférez une application et modifiez son architecture en tirant pleinement parti des fonctionnalités natives cloud pour améliorer l'agilité, les performances et la capacité de mise à l'échelle. Cela implique généralement le transfert du système d'exploitation et de la base de données. Exemple : migrez votre base de données Oracle sur site vers l'édition compatible avec Amazon Aurora PostgreSQL.
- Replateformer (déplacer et remodeler) : transférez une application vers le cloud et introduisez un certain niveau d'optimisation pour tirer parti des fonctionnalités du cloud. Exemple : migrez votre base de données Oracle sur site vers Amazon Relational Database Service (Amazon RDS) pour Oracle dans le. AWS Cloud
- Racheter (rachat) : optez pour un autre produit, généralement en passant d'une licence traditionnelle à un modèle SaaS. Exemple : migrez votre système de gestion de la relation client (CRM) vers Salesforce.com.
- Réhéberger (lift and shift) : transférez une application vers le cloud sans apporter de modifications pour tirer parti des fonctionnalités du cloud. Exemple : migrez votre base de données Oracle locale vers Oracle sur une EC2 instance du AWS Cloud.
- Relocaliser (lift and shift au niveau de l'hyperviseur) : transférez l'infrastructure vers le cloud sans acheter de nouveau matériel, réécrire des applications ou modifier vos opérations existantes. Vous migrez des serveurs d'une plateforme sur site vers un service cloud pour la même plateforme. Exemple : migrer une Microsoft Hyper-V application vers AWS.
- Retenir : conservez les applications dans votre environnement source. Il peut s'agir d'applications nécessitant une refactorisation majeure, que vous souhaitez retarder, et d'applications existantes que vous souhaitez retenir, car rien ne justifie leur migration sur le plan commercial.

- Retirer : mettez hors service ou supprimez les applications dont vous n'avez plus besoin dans votre environnement source.

A

ABAC

Voir contrôle d'accès basé sur les attributs.

services abstraits

Consultez la section Services gérés.

ACIDE

Voir atomicité, consistance, isolation, durabilité.

migration active-active

Méthode de migration de base de données dans laquelle la synchronisation des bases de données source et cible est maintenue (à l'aide d'un outil de réplication bidirectionnelle ou d'opérations d'écriture double), tandis que les deux bases de données gèrent les transactions provenant de la connexion d'applications pendant la migration. Cette méthode prend en charge la migration par petits lots contrôlés au lieu d'exiger un basculement ponctuel. Elle est plus flexible mais demande plus de travail qu'une migration active-passive.

migration active-passive

Méthode de migration de base de données dans laquelle les bases de données source et cible sont synchronisées, mais seule la base de données source gère les transactions liées à la connexion des applications pendant que les données sont répliquées vers la base de données cible. La base de données cible n'accepte aucune transaction pendant la migration.

fonction d'agrégation

Fonction SQL qui agit sur un groupe de lignes et calcule une valeur de retour unique pour le groupe. Des exemples de fonctions d'agrégation incluent `SUM` et`MAX`.

AI

Voir intelligence artificielle.

AIOps

Voir les opérations d'intelligence artificielle.

anonymisation

Processus de suppression définitive d'informations personnelles dans un ensemble de données. L'anonymisation peut contribuer à protéger la vie privée. Les données anonymisées ne sont plus considérées comme des données personnelles.

anti-motif

Solution fréquemment utilisée pour un problème récurrent lorsque la solution est contre-productive, inefficace ou moins efficace qu'une alternative.

contrôle des applications

Une approche de sécurité qui permet d'utiliser uniquement des applications approuvées afin de protéger un système contre les logiciels malveillants.

portefeuille d'applications

Ensemble d'informations détaillées sur chaque application utilisée par une organisation, y compris le coût de génération et de maintenance de l'application, ainsi que sa valeur métier. Ces informations sont essentielles pour le processus de découverte et d'analyse du portefeuille et permettent d'identifier et de prioriser les applications à migrer, à moderniser et à optimiser.

intelligence artificielle (IA)

Domaine de l'informatique consacré à l'utilisation des technologies de calcul pour exécuter des fonctions cognitives généralement associées aux humains, telles que l'apprentissage, la résolution de problèmes et la reconnaissance de modèles. Pour plus d'informations, veuillez consulter Qu'est-ce que l'intelligence artificielle ?

opérations d'intelligence artificielle (AIOps)

Processus consistant à utiliser des techniques de machine learning pour résoudre les problèmes opérationnels, réduire les incidents opérationnels et les interventions humaines, mais aussi améliorer la qualité du service. Pour plus d'informations sur son AIOps utilisation dans la stratégie de AWS migration, consultez le guide d'intégration des opérations.

chiffrement asymétrique

Algorithme de chiffrement qui utilise une paire de clés, une clé publique pour le chiffrement et une clé privée pour le déchiffrement. Vous pouvez partager la clé publique, car elle n'est pas utilisée pour le déchiffrement, mais l'accès à la clé privée doit être très restreint.

atomicité, cohérence, isolement, durabilité (ACID)

Ensemble de propriétés logicielles garantissant la validité des données et la fiabilité opérationnelle d'une base de données, même en cas d'erreur, de panne de courant ou d'autres problèmes.

contrôle d'accès par attributs (ABAC)

Pratique qui consiste à créer des autorisations détaillées en fonction des attributs de l'utilisateur, tels que le service, le poste et le nom de l'équipe. Pour plus d'informations, consultez ABAC pour AWS dans la documentation Gestion des identités et des accès AWS (IAM).

source de données faisant autorité

Emplacement où vous stockez la version principale des données, considérée comme la source d'information la plus fiable. Vous pouvez copier les données de la source de données officielle vers d'autres emplacements à des fins de traitement ou de modification des données, par exemple en les anonymisant, en les expurgant ou en les pseudonymisant.

Zone de disponibilité

Un emplacement distinct au sein d'un Région AWS réseau isolé des défaillances dans d'autres zones de disponibilité et fournissant une connectivité réseau peu coûteuse et à faible latence aux autres zones de disponibilité de la même région.

AWS Cadre d'adoption du cloud (AWS CAF)

Un cadre de directives et de meilleures pratiques visant AWS à aider les entreprises à élaborer un plan efficace pour réussir leur migration vers le cloud. AWS La CAF organise ses conseils en six domaines prioritaires appelés perspectives : les affaires, les personnes, la gouvernance, les plateformes, la sécurité et les opérations. Les perspectives d'entreprise, de personnes et de gouvernance mettent l'accent sur les compétences et les processus métier, tandis que les perspectives relatives à la plateforme, à la sécurité et aux opérations se concentrent sur les compétences et les processus techniques. Par exemple, la perspective liée aux personnes cible les parties prenantes qui s'occupent des ressources humaines (RH), des fonctions de dotation en personnel et de la gestion des personnes. Dans cette perspective, la AWS CAF fournit des conseils pour le développement du personnel, la formation et les communications afin de préparer l'organisation à une adoption réussie du cloud. Pour plus d'informations, veuillez consulter le site Web AWS CAF et le livre blanc AWS CAF.

AWS Cadre de qualification de la charge de travail (AWS WQF)

Outil qui évalue les charges de travail liées à la migration des bases de données, recommande des stratégies de migration et fournit des estimations de travail. AWS Le WQF est inclus avec

AWS Schema Conversion Tool (AWS SCT). Il analyse les schémas de base de données et les objets de code, le code d'application, les dépendances et les caractéristiques de performance, et fournit des rapports d'évaluation.

B

mauvais bot

Un bot destiné à perturber ou à nuire à des individus ou à des organisations.

BCP

Consultez la section Planification de la continuité des activités.

graphique de comportement

Vue unifiée et interactive des comportements des ressources et des interactions au fil du temps. Vous pouvez utiliser un graphique de comportement avec Amazon Detective pour examiner les tentatives de connexion infructueuses, les appels d'API suspects et les actions similaires. Pour plus d'informations, veuillez consulter Data in a behavior graph dans la documentation Detective.

système de poids fort

Système qui stocke d'abord l'octet le plus significatif. Voir aussi endianité.

classification binaire

Processus qui prédit un résultat binaire (l'une des deux classes possibles). Par exemple, votre modèle de machine learning peut avoir besoin de prévoir des problèmes tels que « Cet e-mail est-il du spam ou non ? » ou « Ce produit est-il un livre ou une voiture ? ».

filtre de Bloom

Structure de données probabiliste et efficace en termes de mémoire qui est utilisée pour tester si un élément fait partie d'un ensemble.

déploiement bleu/vert

Stratégie de déploiement dans laquelle vous créez deux environnements distincts mais identiques. Vous exécutez la version actuelle de l'application dans un environnement (bleu) et la nouvelle version de l'application dans l'autre environnement (vert). Cette stratégie vous permet de revenir rapidement en arrière avec un impact minimal.

bot

Application logicielle qui exécute des tâches automatisées sur Internet et simule l'activité ou l'interaction humaine. Certains robots sont utiles ou bénéfiques, comme les robots d'exploration Web qui indexent des informations sur Internet. D'autres robots, appelés « bots malveillants », sont destinés à perturber ou à nuire à des individus ou à des organisations.

botnet

Réseaux de robots infectés par des logiciels malveillants et contrôlés par une seule entité, connue sous le nom d'herder ou d'opérateur de bots. Les botnets sont le mécanisme le plus connu pour faire évoluer les bots et leur impact.

branche

Zone contenue d'un référentiel de code. La première branche créée dans un référentiel est la branche principale. Vous pouvez créer une branche à partir d'une branche existante, puis développer des fonctionnalités ou corriger des bogues dans la nouvelle branche. Une branche que vous créez pour générer une fonctionnalité est communément appelée branche de fonctionnalités. Lorsque la fonctionnalité est prête à être publiée, vous fusionnez à nouveau la branche de fonctionnalités dans la branche principale. Pour plus d'informations, consultez À propos des branches (GitHub documentation).

accès par brise-vitre

Dans des circonstances exceptionnelles et par le biais d'un processus approuvé, il s'agit d'un moyen rapide pour un utilisateur d'accéder à un accès auquel Compte AWS il n'est généralement pas autorisé. Pour plus d'informations, consultez l'indicateur Implementation break-glass procedures dans le guide Well-Architected AWS .

stratégie existante (brownfield)

L'infrastructure existante de votre environnement. Lorsque vous adoptez une stratégie existante pour une architecture système, vous concevez l'architecture en fonction des contraintes des systèmes et de l'infrastructure actuels. Si vous étendez l'infrastructure existante, vous pouvez combiner des politiques brownfield (existantes) et greenfield (inédites).

cache de tampon

Zone de mémoire dans laquelle sont stockées les données les plus fréquemment consultées.

capacité métier

Ce que fait une entreprise pour générer de la valeur (par exemple, les ventes, le service client ou le marketing). Les architectures de microservices et les décisions de développement

peuvent être dictées par les capacités métier. Pour plus d'informations, veuillez consulter la section Organisation en fonction des capacités métier du livre blanc Exécution de microservices conteneurisés sur AWS.

planification de la continuité des activités (BCP)

Plan qui tient compte de l'impact potentiel d'un événement perturbateur, tel qu'une migration à grande échelle, sur les opérations, et qui permet à une entreprise de reprendre ses activités rapidement.

C

CAF

Voir le cadre d'adoption du AWS cloud.

déploiement de Canary

Diffusion lente et progressive d'une version pour les utilisateurs finaux. Lorsque vous êtes sûr, vous déployez la nouvelle version et remplacez la version actuelle dans son intégralité.

CCo E

Voir le Centre d'excellence du cloud.

CDC

Consultez la section Capture des données de modification.

capture des données de modification (CDC)

Processus de suivi des modifications apportées à une source de données, telle qu'une table de base de données, et d'enregistrement des métadonnées relatives à ces modifications. Vous pouvez utiliser la CDC à diverses fins, telles que l'audit ou la réplication des modifications dans un système cible afin de maintenir la synchronisation.

ingénierie du chaos

Introduire intentionnellement des défaillances ou des événements perturbateurs pour tester la résilience d'un système. Vous pouvez utiliser AWS Fault Injection Service (AWS FIS) pour effectuer des expériences qui stressent vos AWS charges de travail et évaluer leur réponse.

CI/CD

Découvrez l'intégration continue et la livraison continue.

classification

Processus de catégorisation qui permet de générer des prédictions. Les modèles de ML pour les problèmes de classification prédisent une valeur discrète. Les valeurs discrètes se distinguent toujours les unes des autres. Par exemple, un modèle peut avoir besoin d'évaluer la présence ou non d'une voiture sur une image.

chiffrement côté client

Chiffrement des données localement, avant que la cible ne les Service AWS reçoive.

Centre d'excellence du cloud (CCoE)

Une équipe multidisciplinaire qui dirige les efforts d'adoption du cloud au sein d'une organisation, notamment en développant les bonnes pratiques en matière de cloud, en mobilisant des ressources, en établissant des délais de migration et en guidant l'organisation dans le cadre de transformations à grande échelle. Pour plus d'informations, consultez les CCoarticles électroniques du blog sur la stratégie AWS Cloud d'entreprise.

cloud computing

Technologie cloud généralement utilisée pour le stockage de données à distance et la gestion des appareils IoT. Le cloud computing est généralement associé à la technologie informatique de pointe.

modèle d'exploitation du cloud

Dans une organisation informatique, modèle d'exploitation utilisé pour créer, faire évoluer et optimiser un ou plusieurs environnements cloud. Pour plus d'informations, consultez la section Création de votre modèle d'exploitation cloud.

étapes d'adoption du cloud

Les quatre phases que les entreprises traversent généralement lorsqu'elles migrent vers AWS Cloud :

- Projet : exécution de quelques projets liés au cloud à des fins de preuve de concept et d'apprentissage
- Base : réaliser des investissements fondamentaux pour accélérer votre adoption du cloud (par exemple, créer une zone de landing zone, définir un CCo E, établir un modèle opérationnel)
- Migration : migration d'applications individuelles
- Réinvention : optimisation des produits et services et innovation dans le cloud

Ces étapes ont été définies par Stephen Orban dans le billet de blog The Journey Toward Cloud-First & the Stages of Adoption publié sur le blog AWS Cloud Enterprise Strategy. Pour plus d'informations sur leur lien avec la stratégie de AWS migration, consultez le guide de préparation à la migration.

CMDB

Consultez la base de données de gestion des configurations.

référentiel de code

Emplacement où le code source et d'autres ressources, comme la documentation, les exemples et les scripts, sont stockés et mis à jour par le biais de processus de contrôle de version. Les référentiels cloud courants incluent GitHub ouBitbucket Cloud. Chaque version du code est appelée branche. Dans une structure de microservice, chaque référentiel est consacré à une seule fonctionnalité. Un seul pipeline CI/CD peut utiliser plusieurs référentiels.

cache passif

Cache tampon vide, mal rempli ou contenant des données obsolètes ou non pertinentes. Cela affecte les performances, car l'instance de base de données doit lire à partir de la mémoire principale ou du disque, ce qui est plus lent que la lecture à partir du cache tampon.

données gelées

Données rarement consultées et généralement historiques. Lorsque vous interrogez ce type de données, les requêtes lentes sont généralement acceptables. Le transfert de ces données vers des niveaux ou classes de stockage moins performants et moins coûteux peut réduire les coûts.

vision par ordinateur (CV)

Domaine de l'IA qui utilise l'apprentissage automatique pour analyser et extraire des informations à partir de formats visuels tels que des images numériques et des vidéos. Par exemple, Amazon SageMaker AI fournit des algorithmes de traitement d'image pour les CV.

dérive de configuration

Pour une charge de travail, une modification de configuration par rapport à l'état attendu. Cela peut entraîner une non-conformité de la charge de travail, et cela est généralement progressif et involontaire.

base de données de gestion des configurations (CMDB)

Référentiel qui stocke et gère les informations relatives à une base de données et à son environnement informatique, y compris les composants matériels et logiciels ainsi que leurs

configurations. Vous utilisez généralement les données d'une CMDB lors de la phase de découverte et d'analyse du portefeuille de la migration.

pack de conformité

Ensemble de AWS Config règles et d'actions correctives que vous pouvez assembler pour personnaliser vos contrôles de conformité et de sécurité. Vous pouvez déployer un pack de conformité en tant qu'entité unique dans une région Compte AWS et, ou au sein d'une organisation, à l'aide d'un modèle YAML. Pour plus d'informations, consultez la section Packs de conformité dans la AWS Config documentation.

intégration continue et livraison continue (CI/CD)

Processus d'automatisation des étapes de source, de construction, de test, de préparation et de production du processus de publication du logiciel. CI/CD est communément décrit comme un pipeline. CI/CD peut vous aider à automatiser les processus, à améliorer la productivité, à améliorer la qualité du code et à accélérer les livraisons. Pour plus d'informations, veuillez consulter Avantages de la livraison continue. CD peut également signifier déploiement continu. Pour plus d'informations, veuillez consulter Livraison continue et déploiement continu.

CV

Voir vision par ordinateur.

D

données au repos

Données stationnaires dans votre réseau, telles que les données stockées.

classification des données

Processus permettant d'identifier et de catégoriser les données de votre réseau en fonction de leur sévérité et de leur sensibilité. Il s'agit d'un élément essentiel de toute stratégie de gestion des risques de cybersécurité, car il vous aide à déterminer les contrôles de protection et de conservation appropriés pour les données. La classification des données est une composante du pilier de sécurité du AWS Well-Architected Framework. Pour plus d'informations, veuillez consulter Classification des données.

dérive des données

Une variation significative entre les données de production et les données utilisées pour entraîner un modèle ML, ou une modification significative des données d'entrée au fil du temps. La dérive

des données peut réduire la qualité, la précision et l'équité globales des prédictions des modèles ML.

données en transit

Données qui circulent activement sur votre réseau, par exemple entre les ressources du réseau.

maillage de données

Un cadre architectural qui fournit une propriété des données distribuée et décentralisée avec une gestion et une gouvernance centralisées.

minimisation des données

Le principe de collecte et de traitement des seules données strictement nécessaires. La pratique de la minimisation des données AWS Cloud peut réduire les risques liés à la confidentialité, les coûts et l'empreinte carbone de vos analyses.

périmètre de données

Ensemble de garde-fous préventifs dans votre AWS environnement qui permettent de garantir que seules les identités fiables accèdent aux ressources fiables des réseaux attendus. Pour plus d'informations, voir Création d'un périmètre de données sur AWS.

prétraitement des données

Pour transformer les données brutes en un format facile à analyser par votre modèle de ML. Le prétraitement des données peut impliquer la suppression de certaines colonnes ou lignes et le traitement des valeurs manquantes, incohérentes ou en double.

provenance des données

Le processus de suivi de l'origine et de l'historique des données tout au long de leur cycle de vie, par exemple la manière dont les données ont été générées, transmises et stockées.

sujet des données

Personne dont les données sont collectées et traitées.

entrepôt des données

Un système de gestion des données qui prend en charge les informations commerciales, telles que les analyses. Les entrepôts de données contiennent généralement de grandes quantités de données historiques et sont généralement utilisés pour les requêtes et les analyses.

langage de définition de base de données (DDL)

Instructions ou commandes permettant de créer ou de modifier la structure des tables et des objets dans une base de données.

langage de manipulation de base de données (DML)

Instructions ou commandes permettant de modifier (insérer, mettre à jour et supprimer) des informations dans une base de données.

DDL

Voir langage de définition de base de données.

ensemble profond

Sert à combiner plusieurs modèles de deep learning à des fins de prédiction. Vous pouvez utiliser des ensembles profonds pour obtenir une prévision plus précise ou pour estimer l'incertitude des prédictions.

deep learning

Un sous-champ de ML qui utilise plusieurs couches de réseaux neuronaux artificiels pour identifier le mappage entre les données d'entrée et les variables cibles d'intérêt.

defense-in-depth

Approche de la sécurité de l'information dans laquelle une série de mécanismes et de contrôles de sécurité sont judicieusement répartis sur l'ensemble d'un réseau informatique afin de protéger la confidentialité, l'intégrité et la disponibilité du réseau et des données qu'il contient. Lorsque vous adoptez cette stratégie AWS, vous ajoutez plusieurs contrôles à différentes couches de la AWS Organizations structure afin de sécuriser les ressources. Par exemple, une defense-in-depth approche peut combiner l'authentification multifactorielle, la segmentation du réseau et le chiffrement.

administrateur délégué

Dans AWS Organizations, un service compatible peut enregistrer un compte AWS membre pour administrer les comptes de l'organisation et gérer les autorisations pour ce service. Ce compte est appelé administrateur délégué pour ce service. Pour plus d'informations et une liste des services compatibles, veuillez consulter la rubrique Services qui fonctionnent avec AWS Organizations dans la documentation AWS Organizations .

déploiement

Processus de mise à disposition d'une application, de nouvelles fonctionnalités ou de corrections de code dans l'environnement cible. Le déploiement implique la mise en œuvre de modifications dans une base de code, puis la génération et l'exécution de cette base de code dans les environnements de l'application.

environnement de développement

Voir environnement.

contrôle de détection

Contrôle de sécurité conçu pour détecter, journaliser et alerter après la survenue d'un événement. Ces contrôles constituent une deuxième ligne de défense et vous alertent en cas d'événements de sécurité qui ont contourné les contrôles préventifs en place. Pour plus d'informations, veuillez consulter la rubrique Contrôles de détection dans Implementing security controls on AWS.

cartographie de la chaîne de valeur du développement (DVSM)

Processus utilisé pour identifier et hiérarchiser les contraintes qui nuisent à la rapidité et à la qualité du cycle de vie du développement logiciel. DVSM étend le processus de cartographie de la chaîne de valeur initialement conçu pour les pratiques de production allégée. Il met l'accent sur les étapes et les équipes nécessaires pour créer et transférer de la valeur tout au long du processus de développement logiciel.

jumeau numérique

Représentation virtuelle d'un système réel, tel qu'un bâtiment, une usine, un équipement industriel ou une ligne de production. Les jumeaux numériques prennent en charge la maintenance prédictive, la surveillance à distance et l'optimisation de la production.

tableau des dimensions

Dans un schéma en étoile, table plus petite contenant les attributs de données relatifs aux données quantitatives d'une table de faits. Les attributs des tables de dimensions sont généralement des champs de texte ou des nombres discrets qui se comportent comme du texte. Ces attributs sont couramment utilisés pour la contrainte des requêtes, le filtrage et l'étiquetage des ensembles de résultats.

catastrophe

Un événement qui empêche une charge de travail ou un système d'atteindre ses objectifs commerciaux sur son site de déploiement principal. Ces événements peuvent être des

catastrophes naturelles, des défaillances techniques ou le résultat d'actions humaines, telles qu'une mauvaise configuration involontaire ou une attaque de logiciel malveillant.

reprise après sinistre (DR)

La stratégie et le processus que vous utilisez pour minimiser les temps d'arrêt et les pertes de données causés par un sinistre. Pour plus d'informations, consultez Disaster Recovery of Workloads on AWS : Recovery in the Cloud in the AWS Well-Architected Framework.

DML

Voir langage de manipulation de base de données.

conception axée sur le domaine

Approche visant à développer un système logiciel complexe en connectant ses composants à des domaines évolutifs, ou objectifs métier essentiels, que sert chaque composant. Ce concept a été introduit par Eric Evans dans son ouvrage Domain-Driven Design: Tackling Complexity in the Heart of Software (Boston : Addison-Wesley Professional, 2003). Pour plus d'informations sur l'utilisation du design piloté par domaine avec le modèle de figuier étrangleur, veuillez consulter Modernizing legacy Microsoft ASP.NET (ASMX) web services incrementally by using containers and Amazon API Gateway.

DR

Voir reprise après sinistre.

détection de dérive

Suivi des écarts par rapport à une configuration de référence. Par exemple, vous pouvez l'utiliser AWS CloudFormation pour détecter la dérive des ressources du système ou AWS Control Tower pour détecter les modifications de votre zone d'atterrissage susceptibles d'affecter le respect des exigences de gouvernance.

DVSM

Voir la cartographie de la chaîne de valeur du développement.

E

EDA

Voir analyse exploratoire des données.

EDI

Voir échange de données informatisé.

informatique de périphérie

Technologie qui augmente la puissance de calcul des appareils intelligents en périphérie d'un réseau IoT. Comparé au cloud computing, l'informatique de pointe peut réduire la latence des communications et améliorer le temps de réponse.

échange de données informatisé (EDI)

L'échange automatique de documents commerciaux entre les organisations. Pour plus d'informations, voir Qu'est-ce que l'échange de données informatisé ?

chiffrement

Processus informatique qui transforme des données en texte clair, lisibles par l'homme, en texte chiffré.

clé de chiffrement

Chaîne cryptographique de bits aléatoires générée par un algorithme cryptographique. La longueur des clés peut varier, et chaque clé est conçue pour être imprévisible et unique.

endianisme

Ordre selon lequel les octets sont stockés dans la mémoire de l'ordinateur. Les systèmes de poids fort stockent d'abord l'octet le plus significatif. Les systèmes de poids faible stockent d'abord l'octet le moins significatif.

point de terminaison

Voir point de terminaison de service.

service de point de terminaison

Service que vous pouvez héberger sur un cloud privé virtuel (VPC) pour le partager avec d'autres utilisateurs. Vous pouvez créer un service de point de terminaison avec AWS PrivateLink et accorder des autorisations à d'autres Comptes AWS ou à Gestion des identités et des accès AWS (IAM) principaux. Ces comptes ou principaux peuvent se connecter à votre service de point de terminaison de manière privée en créant des points de terminaison d'un VPC d'interface. Pour plus d'informations, veuillez consulter Création d'un service de point de terminaison dans la documentation Amazon Virtual Private Cloud (Amazon VPC).

planification des ressources d'entreprise (ERP)

Système qui automatise et gère les principaux processus métier (tels que la comptabilité, le MES et la gestion de projet) pour une entreprise.

chiffrement d'enveloppe

Processus de chiffrement d'une clé de chiffrement à l'aide d'une autre clé de chiffrement. Pour plus d'informations, consultez la section Chiffrement des enveloppes dans la documentation AWS Key Management Service (AWS KMS).

environment

Instance d'une application en cours d'exécution. Les types d'environnement les plus courants dans le cloud computing sont les suivants :

- Environnement de développement : instance d'une application en cours d'exécution à laquelle seule l'équipe principale chargée de la maintenance de l'application peut accéder. Les environnements de développement sont utilisés pour tester les modifications avant de les promouvoir dans les environnements supérieurs. Ce type d'environnement est parfois appelé environnement de test.
- Environnements inférieurs : tous les environnements de développement d'une application, tels que ceux utilisés pour les générations et les tests initiaux.
- Environnement de production : instance d'une application en cours d'exécution à laquelle les utilisateurs finaux peuvent accéder. Dans un CI/CD pipeline, l'environnement de production est le dernier environnement de déploiement.
- Environnements supérieurs : tous les environnements accessibles aux utilisateurs autres que l'équipe de développement principale. Ils peuvent inclure un environnement de production, des environnements de préproduction et des environnements pour les tests d'acceptation par les utilisateurs.

épopée

Dans les méthodologies agiles, catégories fonctionnelles qui aident à organiser et à prioriser votre travail. Les épopées fournissent une description détaillée des exigences et des tâches d'implémentation. Par exemple, les points forts de la AWS CAF en matière de sécurité incluent la gestion des identités et des accès, les contrôles de détection, la sécurité des infrastructures, la protection des données et la réponse aux incidents. Pour plus d'informations sur les épopées dans la stratégie de migration AWS , veuillez consulter le guide d'implémentation du programme.

ERP

Voir Planification des ressources d'entreprise.

analyse exploratoire des données (EDA)

Processus d'analyse d'un jeu de données pour comprendre ses principales caractéristiques. Vous collectez ou agrégez des données, puis vous effectuez des enquêtes initiales pour trouver des modèles, détecter des anomalies et vérifier les hypothèses. L'EDA est réalisée en calculant des statistiques récapitulatives et en créant des visualisations de données.

F

tableau des faits

La table centrale dans un schéma en étoile. Il stocke des données quantitatives sur les opérations commerciales. Généralement, une table de faits contient deux types de colonnes : celles qui contiennent des mesures et celles qui contiennent une clé étrangère pour une table de dimensions.

échouer rapidement

Une philosophie qui utilise des tests fréquents et progressifs pour réduire le cycle de vie du développement. C'est un élément essentiel d'une approche agile.

limite d'isolation des défauts

Dans le AWS Cloud, une limite telle qu'une zone de disponibilité Région AWS, un plan de contrôle ou un plan de données qui limite l'effet d'une panne et contribue à améliorer la résilience des charges de travail. Pour plus d'informations, consultez la section Limites d'isolation des AWS pannes.

branche de fonctionnalités

Voir succursale.

fonctionnalités

Les données d'entrée que vous utilisez pour faire une prédiction. Par exemple, dans un contexte de fabrication, les fonctionnalités peuvent être des images capturées périodiquement à partir de la ligne de fabrication.

importance des fonctionnalités

Le niveau d'importance d'une fonctionnalité pour les prédictions d'un modèle. Il s'exprime généralement sous la forme d'un score numérique qui peut être calculé à l'aide de différentes techniques, telles que la méthode Shapley Additive Explanations (SHAP) et les gradients intégrés. Pour plus d'informations, voir Interprétabilité du modèle d'apprentissage automatique avec AWS.

transformation de fonctionnalité

Optimiser les données pour le processus de ML, notamment en enrichissant les données avec des sources supplémentaires, en mettant à l'échelle les valeurs ou en extrayant plusieurs ensembles d'informations à partir d'un seul champ de données. Cela permet au modèle de ML de tirer parti des données. Par exemple, si vous décomposez la date « 2021-05-27 00:15:37 » en « 2021 », « mai », « jeudi » et « 15 », vous pouvez aider l'algorithme d'apprentissage à apprendre des modèles nuancés associés à différents composants de données.

invitation en quelques coups

Fournir à un LLM un petit nombre d'exemples illustrant la tâche et le résultat souhaité avant de lui demander d'effectuer une tâche similaire. Cette technique est une application de l'apprentissage contextuel, dans le cadre de laquelle les modèles apprennent à partir d'exemples (prises de vue) intégrés dans des instructions. Les instructions en quelques clics peuvent être efficaces pour les tâches qui nécessitent un formatage, un raisonnement ou des connaissances de domaine spécifiques. Voir également l'invite Zero-Shot.

FGAC

Découvrez le contrôle d'accès détaillé.

contrôle d'accès détaillé (FGAC)

Utilisation de plusieurs conditions pour autoriser ou refuser une demande d'accès.

migration instantanée (flash-cut)

Méthode de migration de base de données qui utilise la réplication continue des données par le biais de la capture des données de modification afin de migrer les données dans les plus brefs délais, au lieu d'utiliser une approche progressive. L'objectif est de réduire au maximum les temps d'arrêt.

FM

Voir le modèle de fondation.

modèle de fondation (FM)

Un vaste réseau neuronal d'apprentissage profond qui s'est entraîné sur d'énormes ensembles de données généralisées et non étiquetées. FMs sont capables d'effectuer une grande variété de tâches générales, telles que comprendre le langage, générer du texte et des images et converser en langage naturel. Pour plus d'informations, voir Que sont les modèles de base ?

G

IA générative

Sous-ensemble de modèles d'IA qui ont été entraînés sur de grandes quantités de données et qui peuvent utiliser une simple invite textuelle pour créer de nouveaux contenus et artefacts, tels que des images, des vidéos, du texte et du son. Pour plus d'informations, consultez Qu'est-ce que l'IA générative.

blocage géographique

Voir les restrictions géographiques.

restrictions géographiques (blocage géographique)

Sur Amazon CloudFront, option permettant d'empêcher les utilisateurs de certains pays d'accéder aux distributions de contenu. Vous pouvez utiliser une liste d'autorisation ou une liste de blocage pour spécifier les pays approuvés et interdits. Pour plus d'informations, consultez la section Restreindre la distribution géographique de votre contenu dans la CloudFront documentation.

Flux de travail Gitflow

Approche dans laquelle les environnements inférieurs et supérieurs utilisent différentes branches dans un référentiel de code source. Le flux de travail Gitflow est considéré comme existant, et le flux de travail basé sur les troncs est l'approche moderne préférée.

image dorée

Un instantané d'un système ou d'un logiciel utilisé comme modèle pour déployer de nouvelles instances de ce système ou logiciel. Par exemple, dans le secteur de la fabrication, une image dorée peut être utilisée pour fournir des logiciels sur plusieurs appareils et contribue à améliorer la vitesse, l'évolutivité et la productivité des opérations de fabrication des appareils.

stratégie inédite

L'absence d'infrastructures existantes dans un nouvel environnement. Lorsque vous adoptez une stratégie inédite pour une architecture système, vous pouvez sélectionner toutes les nouvelles technologies sans restriction de compatibilité avec l'infrastructure existante, également appelée brownfield. Si vous étendez l'infrastructure existante, vous pouvez combiner des politiques brownfield (existantes) et greenfield (inédites).

barrière de protection

Règle de haut niveau qui permet de régir les ressources, les politiques et la conformité au sein des unités organisationnelles (OUs). Les barrières de protection préventives appliquent des politiques pour garantir l'alignement sur les normes de conformité. Elles sont mises en œuvre à l'aide de politiques de contrôle des services et de limites des autorisations IAM. Les barrières de protection de détection détectent les violations des politiques et les problèmes de conformité, et génèrent des alertes pour y remédier. Ils sont implémentés à l'aide d'Amazon AWS Config AWS Security Hub CSPM GuardDuty AWS Trusted Advisor, d'Amazon Inspector et de AWS Lambda contrôles personnalisés.

H

HA

Découvrez la haute disponibilité.

migration de base de données hétérogène

Migration de votre base de données source vers une base de données cible qui utilise un moteur de base de données différent (par exemple, Oracle vers Amazon Aurora). La migration hétérogène fait généralement partie d'un effort de réarchitecture, et la conversion du schéma peut s'avérer une tâche complexe. AWS propose AWS SCT qui facilite les conversions de schémas.

haute disponibilité (HA)

Capacité d'une charge de travail à fonctionner en continu, sans intervention, en cas de difficultés ou de catastrophes. Les systèmes HA sont conçus pour basculer automatiquement, fournir constamment des performances de haute qualité et gérer différentes charges et défaillances avec un impact minimal sur les performances.

modernisation des historiens

Approche utilisée pour moderniser et mettre à niveau les systèmes de technologie opérationnelle (OT) afin de mieux répondre aux besoins de l'industrie manufacturière. Un historien est un type de base de données utilisé pour collecter et stocker des données provenant de diverses sources dans une usine.

données de rétention

Partie de données historiques étiquetées qui n'est pas divulguée dans un ensemble de données utilisé pour entraîner un modèle d'apprentissage automatique. Vous pouvez utiliser les données de blocage pour évaluer les performances du modèle en comparant les prévisions du modèle aux données de blocage.

migration de base de données homogène

Migration de votre base de données source vers une base de données cible qui partage le même moteur de base de données (par exemple, Microsoft SQL Server vers Amazon RDS for SQL Server). La migration homogène s'inscrit généralement dans le cadre d'un effort de réhébergement ou de replateforme. Vous pouvez utiliser les utilitaires de base de données natifs pour migrer le schéma.

données chaudes

Données fréquemment consultées, telles que les données en temps réel ou les données translationnelles récentes. Ces données nécessitent généralement un niveau ou une classe de stockage à hautes performances pour fournir des réponses rapides aux requêtes.

correctif

Solution d'urgence à un problème critique dans un environnement de production. En raison de son urgence, un correctif est généralement créé en dehors du flux de travail de DevOps publication habituel.

période de soins intensifs

Immédiatement après le basculement, période pendant laquelle une équipe de migration gère et surveille les applications migrées dans le cloud afin de résoudre les problèmes éventuels. En règle générale, cette période dure de 1 à 4 jours. À la fin de la période de soins intensifs, l'équipe de migration transfère généralement la responsabilité des applications à l'équipe des opérations cloud.

I

IaC

Considérez l'infrastructure comme un code.

politique basée sur l’identité

Politique attachée à un ou plusieurs principaux IAM qui définit leurs autorisations au sein de l' AWS Cloud environnement.

application inactive

Application dont l'utilisation moyenne du processeur et de la mémoire se situe entre 5 et 20 % sur une période de 90 jours. Dans un projet de migration, il est courant de retirer ces applications ou de les retenir sur site.

IIo T

Voir Internet industriel des objets.

infrastructure immuable

Modèle qui déploie une nouvelle infrastructure pour les charges de travail de production au lieu de mettre à jour, d'appliquer des correctifs ou de modifier l'infrastructure existante. Les infrastructures immuables sont intrinsèquement plus cohérentes, fiables et prévisibles que les infrastructures mutables. Pour plus d'informations, consultez les meilleures pratiques de déploiement à l'aide d'une infrastructure immuable dans le AWS Well-Architected Framework.

VPC entrant (d'entrée)

Dans une architecture AWS multi-comptes, un VPC qui accepte, inspecte et achemine les connexions réseau depuis l'extérieur d'une application. L'architecture AWS de référence de sécurité recommande de configurer votre compte réseau avec les fonctions entrantes, sortantes et d'inspection VPCs afin de protéger l'interface bidirectionnelle entre votre application et l'Internet en général.

migration incrémentielle

Stratégie de basculement dans le cadre de laquelle vous migrez votre application par petites parties au lieu d'effectuer un basculement complet unique. Par exemple, il se peut que vous ne transfériez que quelques microservices ou utilisateurs vers le nouveau système dans un premier temps. Après avoir vérifié que tout fonctionne correctement, vous pouvez transférer

progressivement des microservices ou des utilisateurs supplémentaires jusqu'à ce que vous puissiez mettre hors service votre système hérité. Cette stratégie réduit les risques associés aux migrations de grande ampleur.

Industry 4.0

Un terme introduit par Klaus Schwab en 2016 pour désigner la modernisation des processus de fabrication grâce aux avancées en matière de connectivité, de données en temps réel, d'automatisation, d'analyse et d'IA/ML.

infrastructure

Ensemble des ressources et des actifs contenus dans l'environnement d'une application.

infrastructure en tant que code (IaC)

Processus de mise en service et de gestion de l'infrastructure d'une application via un ensemble de fichiers de configuration. IaC est conçue pour vous aider à centraliser la gestion de l'infrastructure, à normaliser les ressources et à mettre à l'échelle rapidement afin que les nouveaux environnements soient reproductibles, fiables et cohérents.

Internet industriel des objets (IIoT)

L'utilisation de capteurs et d'appareils connectés à Internet dans les secteurs industriels tels que la fabrication, l'énergie, l'automobile, les soins de santé, les sciences de la vie et l'agriculture. Pour plus d'informations, voir Élaboration d'une stratégie de transformation numérique de l'Internet des objets (IIoT) industriel.

VPC d'inspection

Dans une architecture AWS multi-comptes, un VPC centralisé qui gère les inspections du trafic réseau VPCs entre (identique ou Régions AWS différent), Internet et les réseaux locaux. L'architecture AWS de référence de sécurité recommande de configurer votre compte réseau avec les fonctions entrantes, sortantes et d'inspection VPCs afin de protéger l'interface bidirectionnelle entre votre application et l'Internet en général.

Internet des objets (IoT)

Réseau d'objets physiques connectés dotés de capteurs ou de processeurs intégrés qui communiquent avec d'autres appareils et systèmes via Internet ou via un réseau de communication local. Pour plus d'informations, veuillez consulter la section Qu'est-ce que l'IoT ?.

interprétabilité

Caractéristique d'un modèle de machine learning qui décrit dans quelle mesure un être humain peut comprendre comment les prédictions du modèle dépendent de ses entrées. Pour plus d'informations, voir Interprétabilité du modèle d'apprentissage automatique avec AWS.

IoT

Voir Internet des objets.

Bibliothèque d'informations informatiques (ITIL)

Ensemble de bonnes pratiques pour proposer des services informatiques et les aligner sur les exigences métier. L'ITIL constitue la base de l'ITSM.

gestion des services informatiques (ITSM)

Activités associées à la conception, à la mise en œuvre, à la gestion et à la prise en charge de services informatiques d'une organisation. Pour plus d'informations sur l'intégration des opérations cloud aux outils ITSM, veuillez consulter le guide d'intégration des opérations.

ITIL

Consultez la bibliothèque d'informations informatiques.

ITSM

Voir Gestion des services informatiques.

L

contrôle d'accès basé sur des étiquettes (LBAC)

Une implémentation du contrôle d'accès obligatoire (MAC) dans laquelle une valeur d'étiquette de sécurité est explicitement attribuée aux utilisateurs et aux données elles-mêmes. L'intersection entre l'étiquette de sécurité utilisateur et l'étiquette de sécurité des données détermine les lignes et les colonnes visibles par l'utilisateur.

zone de destination

Une zone d'atterrissage est un AWS environnement multi-comptes bien conçu, évolutif et sécurisé. Il s'agit d'un point de départ à partir duquel vos entreprises peuvent rapidement lancer et déployer des charges de travail et des applications en toute confiance dans leur environnement de sécurité et d'infrastructure. Pour plus d'informations sur les zones de destination, veuillez consulter Setting up a secure and scalable multi-account AWS environment.

grand modèle de langage (LLM)

Un modèle d'intelligence artificielle basé sur le deep learning qui est préentraîné sur une grande quantité de données. Un LLM peut effectuer plusieurs tâches, telles que répondre à des questions, résumer des documents, traduire du texte dans d'autres langues et compléter des phrases. Pour plus d'informations, voir Que sont LLMs.

migration de grande envergure

Migration de 300 serveurs ou plus.

LBAC

Voir contrôle d'accès basé sur des étiquettes.

principe de moindre privilège

Bonne pratique de sécurité qui consiste à accorder les autorisations minimales nécessaires à l'exécution d'une tâche. Pour plus d'informations, veuillez consulter la rubrique Accorder les autorisations de moindre privilège dans la documentation IAM.

lift and shift

Voir 7 Rs.

système de poids faible

Système qui stocke d'abord l'octet le moins significatif. Voir aussi endianité.

LLM

Voir le grand modèle de langage.

environnements inférieurs

Voir environnement.

M

machine learning (ML)

Type d'intelligence artificielle qui utilise des algorithmes et des techniques pour la reconnaissance et l'apprentissage de modèles. Le ML analyse et apprend à partir de données enregistrées, telles que les données de l'Internet des objets (IoT), pour générer un modèle statistique basé sur des modèles. Pour plus d'informations, veuillez consulter Machine Learning.

branche principale

Voir succursale.

malware

Logiciel conçu pour compromettre la sécurité ou la confidentialité de l'ordinateur. Les logiciels malveillants peuvent perturber les systèmes informatiques, divulguer des informations sensibles ou obtenir un accès non autorisé. Parmi les malwares, on peut citer les virus, les vers, les rançongiciels, les chevaux de Troie, les logiciels espions et les enregistreurs de frappe.

services gérés

Services AWS pour lequel AWS fonctionnent la couche d'infrastructure, le système d'exploitation et les plateformes, et vous accédez aux points de terminaison pour stocker et récupérer des données. Amazon Simple Storage Service (Amazon S3) et Amazon DynamoDB sont des exemples de services gérés. Ils sont également appelés services abstraits.

système d'exécution de la fabrication (MES)

Un système logiciel pour le suivi, la surveillance, la documentation et le contrôle des processus de production qui convertissent les matières premières en produits finis dans l'atelier.

MAP

Voir Migration Acceleration Program.

mécanisme

Processus complet au cours duquel vous créez un outil, favorisez son adoption, puis inspectez les résultats afin de procéder aux ajustements nécessaires. Un mécanisme est un cycle qui se renforce et s'améliore lorsqu'il fonctionne. Pour plus d'informations, voir Création de mécanismes dans le cadre AWS Well-Architected.

compte membre

Tous, à l' Comptes AWS exception du compte de gestion, qui font partie d'une organisation dans AWS Organizations. Un compte ne peut être membre que d'une seule organisation à la fois.

MAILLES

Voir le système d'exécution de la fabrication.

Transport télémétrique en file d'attente de messages (MQTT)

Protocole de communication léger machine-to-machine (M2M), basé sur le modèle de publication/d'abonnement, pour les appareils IoT aux ressources limitées.

microservice

Un petit service indépendant qui communique via un réseau bien défini APIs et qui est généralement détenu par de petites équipes autonomes. Par exemple, un système d'assurance peut inclure des microservices qui mappent à des capacités métier, telles que les ventes ou le marketing, ou à des sous-domaines, tels que les achats, les réclamations ou l'analytique. Les avantages des microservices incluent l'agilité, la flexibilité de la mise à l'échelle, la facilité de déploiement, la réutilisation du code et la résilience. Pour plus d'informations, consultez la section Intégration de microservices à l'aide de services AWS sans serveur.

architecture de microservices

Approche de création d'une application avec des composants indépendants qui exécutent chaque processus d'application en tant que microservice. Ces microservices communiquent via une interface bien définie en utilisant Lightweight. APIs Chaque microservice de cette architecture peut être mis à jour, déployé et mis à l'échelle pour répondre à la demande de fonctions spécifiques d'une application. Pour plus d'informations, consultez la section Implémentation de microservices sur AWS.

Programme d'accélération des migrations (MAP)

Un AWS programme qui fournit un support de conseil, des formations et des services pour aider les entreprises à établir une base opérationnelle solide pour passer au cloud, et pour aider à compenser le coût initial des migrations. MAP inclut une méthodologie de migration pour exécuter les migrations héritées de manière méthodique, ainsi qu'un ensemble d'outils pour automatiser et accélérer les scénarios de migration courants.

migration à grande échelle

Processus consistant à transférer la majeure partie du portefeuille d'applications vers le cloud par vagues, un plus grand nombre d'applications étant déplacées plus rapidement à chaque vague. Cette phase utilise les bonnes pratiques et les enseignements tirés des phases précédentes pour implémenter une usine de migration d'équipes, d'outils et de processus en vue de rationaliser la migration des charges de travail grâce à l'automatisation et à la livraison agile. Il s'agit de la troisième phase de la stratégie de migration AWS.

usine de migration

Équipes interfonctionnelles qui rationalisent la migration des charges de travail grâce à des approches automatisées et agiles. Les équipes de Migration Factory comprennent généralement des responsables des opérations, des analystes commerciaux et des propriétaires, des ingénieurs de migration, des développeurs et DevOps des professionnels travaillant dans le cadre de sprints.

Entre 20 et 50 % du portefeuille d'applications d'entreprise est constitué de modèles répétés qui peuvent être optimisés par une approche d'usine. Pour plus d'informations, veuillez consulter la rubrique discussion of migration factories et le guide Cloud Migration Factory dans cet ensemble de contenus.

métadonnées de migration

Informations relatives à l'application et au serveur nécessaires pour finaliser la migration. Chaque modèle de migration nécessite un ensemble de métadonnées de migration différent. Les exemples de métadonnées de migration incluent le sous-réseau cible, le groupe de sécurité et le AWS compte.

modèle de migration

Tâche de migration reproductible qui détaille la stratégie de migration, la destination de la migration et l'application ou le service de migration utilisé. Exemple : réorganisez la migration vers Amazon EC2 avec le service de migration AWS d'applications.

Évaluation du portefeuille de migration (MPA)

Outil en ligne qui fournit des informations pour valider l'analyse de rentabilisation en faveur de la migration vers le. AWS Cloud La MPA propose une évaluation détaillée du portefeuille (dimensionnement approprié des serveurs, tarification, comparaison du coût total de possession, analyse des coûts de migration), ainsi que la planification de la migration (analyse et collecte des données d'applications, regroupement des applications, priorisation des migrations et planification des vagues). L'outil MPA (connexion requise) est disponible gratuitement pour tous les AWS consultants et consultants APN Partner.

Évaluation de la préparation à la migration (MRA)

Processus qui consiste à obtenir des informations sur l'état de préparation d'une organisation au cloud, à identifier les forces et les faiblesses et à élaborer un plan d'action pour combler les lacunes identifiées, à l'aide du AWS CAF. Pour plus d'informations, veuillez consulter le guide de préparation à la migration. La MRA est la première phase de la stratégie de migration AWS.

stratégie de migration

L'approche utilisée pour migrer une charge de travail vers le AWS Cloud. Pour plus d'informations, reportez-vous aux 7 R de ce glossaire et à Mobiliser votre organisation pour accélérer les migrations à grande échelle.

ML

Voir apprentissage automatique.

modernisation

Transformation d'une application obsolète (héritée ou monolithique) et de son infrastructure en un système agile, élastique et hautement disponible dans le cloud afin de réduire les coûts, de gagner en efficacité et de tirer parti des innovations. Pour plus d'informations, consultez la section Stratégie de modernisation des applications dans le AWS Cloud.

évaluation de la préparation à la modernisation

Évaluation qui permet de déterminer si les applications d'une organisation sont prêtes à être modernisées, d'identifier les avantages, les risques et les dépendances, et qui détermine dans quelle mesure l'organisation peut prendre en charge l'état futur de ces applications. Le résultat de l'évaluation est un plan de l'architecture cible, une feuille de route détaillant les phases de développement et les étapes du processus de modernisation, ainsi qu'un plan d'action pour combler les lacunes identifiées. Pour plus d'informations, consultez la section Évaluation de l'état de préparation à la modernisation des applications dans le AWS Cloud.

applications monolithiques (monolithes)

Applications qui s'exécutent en tant que service unique avec des processus étroitement couplés. Les applications monolithiques ont plusieurs inconvénients. Si une fonctionnalité de l'application connaît un pic de demande, l'architecture entière doit être mise à l'échelle. L'ajout ou l'amélioration des fonctionnalités d'une application monolithique devient également plus complexe lorsque la base de code s'élargit. Pour résoudre ces problèmes, vous pouvez utiliser une architecture de microservices. Pour plus d'informations, veuillez consulter Decomposing monoliths into microservices.

MPA

Voir Évaluation du portefeuille de migration.

MQTT

Voir Message Queuing Telemetry Transport.

classification multi-classes

Processus qui permet de générer des prédictions pour plusieurs classes (prédiction d'un résultat parmi plus de deux). Par exemple, un modèle de ML peut demander « Ce produit est-il un livre, une voiture ou un téléphone ? » ou « Quelle catégorie de produits intéresse le plus ce client ? ».

infrastructure mutable

Modèle qui met à jour et modifie l'infrastructure existante pour les charges de travail de production. Pour améliorer la cohérence, la fiabilité et la prévisibilité, le AWS Well-Architected Framework recommande l'utilisation d'une infrastructure immuable comme meilleure pratique.

O

OAC

Voir Contrôle d'accès à l'origine.

OAI

Voir l'identité d'accès à l'origine.

OCM

Voir gestion du changement organisationnel.

migration hors ligne

Méthode de migration dans laquelle la charge de travail source est supprimée au cours du processus de migration. Cette méthode implique un temps d'arrêt prolongé et est généralement utilisée pour de petites charges de travail non critiques.

OI

Consultez la section Intégration des opérations.

OLA

Voir l'accord au niveau opérationnel.

migration en ligne

Méthode de migration dans laquelle la charge de travail source est copiée sur le système cible sans être mise hors ligne. Les applications connectées à la charge de travail peuvent continuer à fonctionner pendant la migration. Cette méthode implique un temps d'arrêt nul ou minimal et est généralement utilisée pour les charges de travail de production critiques.

OPC-UA

Voir Open Process Communications - Architecture unifiée.

Communications par processus ouvert - Architecture unifiée (OPC-UA)

Un protocole de communication machine-to-machine (M2M) pour l'automatisation industrielle. L'OPC-UA fournit une norme d'interopérabilité avec des schémas de cryptage, d'authentification et d'autorisation des données.

accord au niveau opérationnel (OLA)

Accord qui précise ce que les groupes informatiques fonctionnels s'engagent à fournir les uns aux autres, afin de prendre en charge un contrat de niveau de service (SLA).

examen de l'état de préparation opérationnelle (ORR)

Une liste de questions et de bonnes pratiques associées qui vous aident à comprendre, à évaluer, à prévenir ou à réduire l'ampleur des incidents et des défaillances possibles. Pour plus d'informations, voir Operational Readiness Reviews (ORR) dans le AWS Well-Architected Framework.

technologie opérationnelle (OT)

Systèmes matériels et logiciels qui fonctionnent avec l'environnement physique pour contrôler les opérations, les équipements et les infrastructures industriels. Dans le secteur manufacturier, l'intégration des systèmes OT et des technologies de l'information (IT) est au cœur des transformations de l'industrie 4.0.

intégration des opérations (OI)

Processus de modernisation des opérations dans le cloud, qui implique la planification de la préparation, l'automatisation et l'intégration. Pour en savoir plus, veuillez consulter le guide d'intégration des opérations.

journal de suivi d'organisation

Un parcours créé par AWS CloudTrail qui enregistre tous les événements pour tous les membres Comptes AWS d'une organisation dans AWS Organizations. Ce journal de suivi est créé dans chaque Compte AWS qui fait partie de l'organisation et suit l'activité de chaque compte. Pour plus d'informations, consultez la section Création d'un suivi pour une organisation dans la CloudTrail documentation.

gestion du changement organisationnel (OCM)

Cadre pour gérer les transformations métier majeures et perturbatrices du point de vue des personnes, de la culture et du leadership. L'OCM aide les organisations à se préparer et à effectuer la transition vers de nouveaux systèmes et de nouvelles politiques en accélérant

l'adoption des changements, en abordant les problèmes de transition et en favorisant des changements culturels et organisationnels. Dans la stratégie de AWS migration, ce cadre est appelé accélération du personnel, en raison de la rapidité du changement requise dans les projets d'adoption du cloud. Pour plus d'informations, veuillez consulter le guide OCM.

contrôle d'accès d'origine (OAC)

Dans CloudFront, une option améliorée pour restreindre l'accès afin de sécuriser votre contenu Amazon Simple Storage Service (Amazon S3). L'OAC prend en charge tous les compartiments S3 dans leur ensemble Régions AWS, le chiffrement côté serveur avec AWS KMS (SSE-KMS) et les requêtes dynamiques PUT adressées au compartiment S3. DELETE

identité d'accès d'origine (OAI)

Dans CloudFront, une option permettant de restreindre l'accès afin de sécuriser votre contenu Amazon S3. Lorsque vous utilisez OAI, il CloudFront crée un principal auprès duquel Amazon S3 peut s'authentifier. Les principaux authentifiés ne peuvent accéder au contenu d'un compartiment S3 que par le biais d'une distribution spécifique CloudFront . Voir également OAC, qui fournit un contrôle d'accès plus précis et amélioré.

ORR

Voir l'examen de l'état de préparation opérationnelle.

DE

Voir technologie opérationnelle.

VPC sortant (de sortie)

Dans une architecture AWS multi-comptes, un VPC qui gère les connexions réseau initiées depuis une application. L'architecture AWS de référence de sécurité recommande de configurer votre compte réseau avec les fonctions entrantes, sortantes et d'inspection VPCs afin de protéger l'interface bidirectionnelle entre votre application et l'Internet en général.

P

limite des autorisations

Politique de gestion IAM attachée aux principaux IAM pour définir les autorisations maximales que peut avoir l'utilisateur ou le rôle. Pour plus d'informations, veuillez consulter la rubrique Limites des autorisations dans la documentation IAM.

informations personnelles identifiables (PII)

Informations qui, lorsqu'elles sont consultées directement ou associées à d'autres données connexes, peuvent être utilisées pour déduire raisonnablement l'identité d'une personne. Les exemples d'informations personnelles incluent les noms, les adresses et les coordonnées.

PII

Voir les informations personnelles identifiables.

manuel stratégique

Ensemble d'étapes prédéfinies qui capturent le travail associé aux migrations, comme la fourniture de fonctions d'opérations de base dans le cloud. Un manuel stratégique peut revêtir la forme de scripts, de runbooks automatisés ou d'un résumé des processus ou des étapes nécessaires au fonctionnement de votre environnement modernisé.

PLC

Voir contrôleur logique programmable.

PLM

Consultez la section Gestion du cycle de vie des produits.

politique

Objet capable de définir les autorisations (voir la politique basée sur l'identité), de spécifier les conditions d'accès (voir la politique basée sur les ressources) ou de définir les autorisations maximales pour tous les comptes d'une organisation dans AWS Organizations (voir la politique de contrôle des services).

persistance polyglotte

Choix indépendant de la technologie de stockage de données d'un microservice en fonction des modèles d'accès aux données et d'autres exigences. Si vos microservices utilisent la même technologie de stockage de données, ils peuvent rencontrer des difficultés d'implémentation ou présenter des performances médiocres. Les microservices sont plus faciles à mettre en œuvre, atteignent de meilleures performances, ainsi qu'une meilleure capacité de mise à l'échelle s'ils utilisent l'entrepôt de données le mieux adapté à leurs besoins. Pour plus d'informations, veuillez consulter Enabling data persistence in microservices.

évaluation du portefeuille

Processus de découverte, d'analyse et de priorisation du portefeuille d'applications afin de planifier la migration. Pour plus d'informations, veuillez consulter Evaluating migration readiness.

predicate

Une condition de requête qui renvoie `true` ou`false`, généralement située dans une `WHERE` clause.

prédicat pushdown

Technique d'optimisation des requêtes de base de données qui filtre les données de la requête avant le transfert. Cela réduit la quantité de données qui doivent être extraites et traitées à partir de la base de données relationnelle et améliore les performances des requêtes.

contrôle préventif

Contrôle de sécurité conçu pour empêcher qu'un événement ne se produise. Ces contrôles constituent une première ligne de défense pour empêcher tout accès non autorisé ou toute modification indésirable de votre réseau. Pour plus d'informations, veuillez consulter Preventative controls dans Implementing security controls on AWS.

principal

Entité AWS capable d'effectuer des actions et d'accéder aux ressources. Cette entité est généralement un utilisateur root pour un Compte AWS rôle IAM ou un utilisateur. Pour plus d'informations, veuillez consulter la rubrique Principal dans Termes et concepts relatifs aux rôles, dans la documentation IAM.

confidentialité dès la conception

Une approche d'ingénierie système qui prend en compte la confidentialité tout au long du processus de développement.

zones hébergées privées

Conteneur contenant des informations sur la manière dont vous souhaitez qu'Amazon Route 53 réponde aux requêtes DNS pour un domaine et ses sous-domaines au sein d'un ou de plusieurs VPCs domaines. Pour plus d'informations, veuillez consulter Working with private hosted zones dans la documentation Route 53.

contrôle proactif

Contrôle de sécurité conçu pour empêcher le déploiement de ressources non conformes. Ces contrôles analysent les ressources avant qu'elles ne soient provisionnées. Si la ressource n'est pas conforme au contrôle, elle n'est pas provisionnée. Pour plus d'informations, consultez le guide de référence sur les contrôles dans la AWS Control Tower documentation et consultez la section Contrôles proactifs dans Implémentation des contrôles de sécurité sur AWS.

gestion du cycle de vie des produits (PLM)

Gestion des données et des processus d'un produit tout au long de son cycle de vie, depuis la conception, le développement et le lancement, en passant par la croissance et la maturité, jusqu'au déclin et au retrait.

environnement de production

Voir environnement.

contrôleur logique programmable (PLC)

Dans le secteur manufacturier, un ordinateur hautement fiable et adaptable qui surveille les machines et automatise les processus de fabrication.

chaînage rapide

Utiliser le résultat d'une invite LLM comme entrée pour l'invite suivante afin de générer de meilleures réponses. Cette technique est utilisée pour décomposer une tâche complexe en sous-tâches ou pour affiner ou développer de manière itérative une réponse préliminaire. Cela permet d'améliorer la précision et la pertinence des réponses d'un modèle et permet d'obtenir des résultats plus précis et personnalisés.

pseudonymisation

Processus de remplacement des identifiants personnels dans un ensemble de données par des valeurs fictives. La pseudonymisation peut contribuer à protéger la vie privée. Les données pseudonymisées sont toujours considérées comme des données personnelles.

publish/subscribe (pub/sub)

Modèle qui permet des communications asynchrones entre les microservices afin d'améliorer l'évolutivité et la réactivité. Par exemple, dans un MES basé sur des microservices, un microservice peut publier des messages d'événements sur un canal auquel d'autres microservices peuvent s'abonner. Le système peut ajouter de nouveaux microservices sans modifier le service de publication.

Q

plan de requête

Série d'étapes, telles que des instructions, utilisées pour accéder aux données d'un système de base de données relationnelle SQL.

régression du plan de requêtes

Le cas où un optimiseur de service de base de données choisit un plan moins optimal qu'avant une modification donnée de l'environnement de base de données. Cela peut être dû à des changements en termes de statistiques, de contraintes, de paramètres d'environnement, de liaisons de paramètres de requêtes et de mises à jour du moteur de base de données.

R

Matrice RACI

Voir responsable, responsable, consulté, informé (RACI).

CHIFFON

Voir Retrieval Augmented Generation.

rançongiciel

Logiciel malveillant conçu pour bloquer l'accès à un système informatique ou à des données jusqu'à ce qu'un paiement soit effectué.

Matrice RASCI

Voir responsable, responsable, consulté, informé (RACI).

RCAC

Voir contrôle d'accès aux lignes et aux colonnes.

réplica en lecture

Copie d'une base de données utilisée en lecture seule. Vous pouvez acheminer les requêtes vers le réplica de lecture pour réduire la charge sur votre base de données principale.

réarchitecte

Voir 7 Rs.

objectif de point de récupération (RPO)

Durée maximale acceptable depuis le dernier point de récupération des données. Il détermine ce qui est considéré comme étant une perte de données acceptable entre le dernier point de reprise et l’interruption du service.

objectif de temps de récupération (RTO)

Le délai maximum acceptable entre l'interruption du service et le rétablissement du service.

refactoriser

Voir 7 Rs.

Région

Ensemble de AWS ressources dans une zone géographique. Chacun Région AWS est isolé et indépendant des autres pour garantir tolérance aux pannes, stabilité et résilience. Pour plus d'informations, voir Spécifier ce que Régions AWS votre compte peut utiliser.

régression

Technique de ML qui prédit une valeur numérique. Par exemple, pour résoudre le problème « Quel sera le prix de vente de cette maison ? », un modèle de ML pourrait utiliser un modèle de régression linéaire pour prédire le prix de vente d'une maison sur la base de faits connus à son sujet (par exemple, la superficie en mètres carrés).

réhéberger

Voir 7 Rs.

version

Dans un processus de déploiement, action visant à promouvoir les modifications apportées à un environnement de production.

déplacer

Voir 7 Rs.

replateforme

Voir 7 Rs.

rachat

Voir 7 Rs.

résilience

La capacité d'une application à résister aux perturbations ou à s'en remettre. La haute disponibilité et la reprise après sinistre sont des considérations courantes lors de la planification de la résilience dans le AWS Cloud. Pour plus d'informations, consultez AWS Cloud Résilience.

politique basée sur les ressources

Politique attachée à une ressource, comme un compartiment Amazon S3, un point de terminaison ou une clé de chiffrement. Ce type de politique précise les principaux auxquels l'accès est autorisé, les actions prises en charge et toutes les autres conditions qui doivent être remplies.

matrice responsable, redevable, consulté et informé (RACI)

Une matrice qui définit les rôles et les responsabilités de toutes les parties impliquées dans les activités de migration et les opérations cloud. Le nom de la matrice est dérivé des types de responsabilité définis dans la matrice : responsable (R), responsable (A), consulté (C) et informé (I). Le type de support (S) est facultatif. Si vous incluez le support, la matrice est appelée matrice RASCI, et si vous l'excluez, elle est appelée matrice RACI.

contrôle réactif

Contrôle de sécurité conçu pour permettre de remédier aux événements indésirables ou aux écarts par rapport à votre référence de sécurité. Pour plus d'informations, veuillez consulter la rubrique Responsive controls dans Implementing security controls on AWS.

retain

Voir 7 Rs.

se retirer

Voir 7 Rs.

Génération augmentée de récupération (RAG)

Technologie d'IA générative dans laquelle un LLM fait référence à une source de données faisant autorité qui se trouve en dehors de ses sources de données de formation avant de générer une réponse. Par exemple, un modèle RAG peut effectuer une recherche sémantique dans la base de connaissances ou dans les données personnalisées d'une organisation. Pour plus d'informations, voir Qu'est-ce que RAG ?

rotation

Processus de mise à jour périodique d'un secret pour empêcher un attaquant d'accéder aux informations d'identification.

contrôle d'accès aux lignes et aux colonnes (RCAC)

Utilisation d'expressions SQL simples et flexibles dotées de règles d'accès définies. Le RCAC comprend des autorisations de ligne et des masques de colonnes.

RPO

Voir l'objectif du point de récupération.

RTO

Voir l'objectif en matière de temps de rétablissement.

runbook

Ensemble de procédures manuelles ou automatisées nécessaires à l'exécution d'une tâche spécifique. Elles visent généralement à rationaliser les opérations ou les procédures répétitives présentant des taux d'erreur élevés.

S

SAML 2.0

Un standard ouvert utilisé par de nombreux fournisseurs d'identité (IdPs). Cette fonctionnalité permet l'authentification unique fédérée (SSO), afin que les utilisateurs puissent se connecter AWS Management Console ou appeler les opérations de l' AWS API sans que vous ayez à créer un utilisateur dans IAM pour tous les membres de votre organisation. Pour plus d'informations sur la fédération SAML 2.0, veuillez consulter À propos de la fédération SAML 2.0 dans la documentation IAM.

SCADA

Voir Contrôle de supervision et acquisition de données.

SCP

Voir la politique de contrôle des services.

secret

Dans AWS Secrets Manager des informations confidentielles ou restreintes, telles qu'un mot de passe ou des informations d'identification utilisateur, que vous stockez sous forme cryptée. Il comprend la valeur secrète et ses métadonnées. La valeur secrète peut être binaire, une chaîne unique ou plusieurs chaînes. Pour plus d'informations, voir Que contient le secret d'un Secrets Manager ? dans la documentation de Secrets Manager.

sécurité dès la conception

Une approche d'ingénierie système qui prend en compte la sécurité tout au long du processus de développement.

contrôle de sécurité

Barrière de protection technique ou administrative qui empêche, détecte ou réduit la capacité d'un assaillant d'exploiter une vulnérabilité de sécurité. Il existe quatre principaux types de contrôles de sécurité : préventifs, détectifs, réactifs et proactifs.

renforcement de la sécurité

Processus qui consiste à réduire la surface d'attaque pour la rendre plus résistante aux attaques. Cela peut inclure des actions telles que la suppression de ressources qui ne sont plus requises, la mise en œuvre des bonnes pratiques de sécurité consistant à accorder le moindre privilège ou la désactivation de fonctionnalités inutiles dans les fichiers de configuration.

système de gestion des informations et des événements de sécurité (SIEM)

Outils et services qui associent les systèmes de gestion des informations de sécurité (SIM) et de gestion des événements de sécurité (SEM). Un système SIEM collecte, surveille et analyse les données provenant de serveurs, de réseaux, d'appareils et d'autres sources afin de détecter les menaces et les failles de sécurité, mais aussi de générer des alertes.

automatisation des réponses de sécurité

Action prédéfinie et programmée conçue pour répondre automatiquement à un événement de sécurité ou y remédier. Ces automatisations servent de contrôles de sécurité détectifs ou réactifs qui vous aident à mettre en œuvre les meilleures pratiques AWS de sécurité. Parmi les actions de réponse automatique, citons la modification d'un groupe de sécurité VPC, l'application de correctifs à une EC2 instance Amazon ou la rotation des informations d'identification.

chiffrement côté serveur

Chiffrement des données à destination, par celui Service AWS qui les reçoit.

Politique de contrôle des services (SCP)

Politique qui fournit un contrôle centralisé des autorisations pour tous les comptes d'une organisation dans AWS Organizations. SCPs définissez des garde-fous ou des limites aux actions qu'un administrateur peut déléguer à des utilisateurs ou à des rôles. Vous pouvez les utiliser SCPs comme listes d'autorisation ou de refus pour spécifier les services ou les actions autorisés ou interdits. Pour plus d'informations, consultez la section Politiques de contrôle des services dans la AWS Organizations documentation.

point de terminaison du service

URL du point d'entrée pour un Service AWS. Pour vous connecter par programmation au service cible, vous pouvez utiliser un point de terminaison. Pour plus d'informations, veuillez consulter la rubrique Service AWS endpoints dans Références générales AWS.

contrat de niveau de service (SLA)

Accord qui précise ce qu'une équipe informatique promet de fournir à ses clients, comme le temps de disponibilité et les performances des services.

indicateur de niveau de service (SLI)

Mesure d'un aspect des performances d'un service, tel que son taux d'erreur, sa disponibilité ou son débit.

objectif de niveau de service (SLO)

Mesure cible qui représente l'état d'un service, tel que mesuré par un indicateur de niveau de service.

modèle de responsabilité partagée

Un modèle décrivant la responsabilité que vous partagez en matière AWS de sécurité et de conformité dans le cloud. AWS est responsable de la sécurité du cloud, alors que vous êtes responsable de la sécurité dans le cloud. Pour de plus amples informations, veuillez consulter Modèle de responsabilité partagée.

SIEM

Consultez les informations de sécurité et le système de gestion des événements.

point de défaillance unique (SPOF)

Défaillance d'un seul composant critique d'une application susceptible de perturber le système.

SLA

Voir le contrat de niveau de service.

SLI

Voir l'indicateur de niveau de service.

SLO

Voir l'objectif de niveau de service.

split-and-seed modèle

Modèle permettant de mettre à l'échelle et d'accélérer les projets de modernisation. Au fur et à mesure que les nouvelles fonctionnalités et les nouvelles versions de produits sont définies, l'équipe principale se divise pour créer des équipes de produit. Cela permet de mettre à l'échelle les capacités et les services de votre organisation, d'améliorer la productivité des développeurs et de favoriser une innovation rapide. Pour plus d'informations, consultez la section Approche progressive de la modernisation des applications dans le. AWS Cloud

SPOF

Voir point de défaillance unique.

schéma en étoile

Structure organisationnelle de base de données qui utilise une grande table de faits pour stocker les données transactionnelles ou mesurées et utilise une ou plusieurs tables dimensionnelles plus petites pour stocker les attributs des données. Cette structure est conçue pour être utilisée dans un entrepôt de données ou à des fins de business intelligence.

modèle de figuier étrangleur

Approche de modernisation des systèmes monolithiques en réécrivant et en remplaçant progressivement les fonctionnalités du système jusqu'à ce que le système hérité puisse être mis hors service. Ce modèle utilise l'analogie d'un figuier de vigne qui se développe dans un arbre existant et qui finit par supplanter son hôte. Le schéma a été présenté par Martin Fowler comme un moyen de gérer les risques lors de la réécriture de systèmes monolithiques. Pour obtenir un exemple d'application de ce modèle, veuillez consulter Modernizing legacy Microsoft ASP.NET (ASMX) web services incrementally by using containers and Amazon API Gateway.

sous-réseau

Plage d'adresses IP dans votre VPC. Un sous-réseau doit se trouver dans une seule zone de disponibilité.

contrôle de supervision et acquisition de données (SCADA)

Dans le secteur manufacturier, un système qui utilise du matériel et des logiciels pour surveiller les actifs physiques et les opérations de production.

chiffrement symétrique

Algorithme de chiffrement qui utilise la même clé pour chiffrer et déchiffrer les données.

tests synthétiques

Tester un système de manière à simuler les interactions des utilisateurs afin de détecter les problèmes potentiels ou de surveiller les performances. Vous pouvez utiliser Amazon CloudWatch Synthetics pour créer ces tests.

invite du système

Technique permettant de fournir un contexte, des instructions ou des directives à un LLM afin d'orienter son comportement. Les instructions du système aident à définir le contexte et à établir des règles pour les interactions avec les utilisateurs.

T

balises

Des paires clé-valeur qui agissent comme des métadonnées pour organiser vos AWS ressources. Les balises peuvent vous aider à gérer, identifier, organiser, rechercher et filtrer des ressources. Pour plus d'informations, veuillez consulter la rubrique Balisage de vos AWS ressources.

variable cible

La valeur que vous essayez de prédire dans le cadre du ML supervisé. Elle est également qualifiée de variable de résultat. Par exemple, dans un environnement de fabrication, la variable cible peut être un défaut du produit.

liste de tâches

Outil utilisé pour suivre les progrès dans un runbook. Liste de tâches qui contient une vue d'ensemble du runbook et une liste des tâches générales à effectuer. Pour chaque tâche générale, elle inclut le temps estimé nécessaire, le propriétaire et l'avancement.

environnement de test

Voir environnement.

entraînement

Pour fournir des données à partir desquelles votre modèle de ML peut apprendre. Les données d'entraînement doivent contenir la bonne réponse. L'algorithme d'apprentissage identifie des modèles dans les données d'entraînement, qui mettent en correspondance les attributs des données d'entrée avec la cible (la réponse que vous souhaitez prédire). Il fournit un modèle de ML

qui capture ces modèles. Vous pouvez alors utiliser le modèle de ML pour obtenir des prédictions sur de nouvelles données pour lesquelles vous ne connaissez pas la cible.

passerelle de transit

Un hub de transit réseau que vous pouvez utiliser pour interconnecter vos réseaux VPCs et ceux sur site. Pour plus d'informations, voir Qu'est-ce qu'une passerelle de transit dans la AWS Transit Gateway documentation.

flux de travail basé sur jonction

Approche selon laquelle les développeurs génèrent et testent des fonctionnalités localement dans une branche de fonctionnalités, puis fusionnent ces modifications dans la branche principale. La branche principale est ensuite intégrée aux environnements de développement, de préproduction et de production, de manière séquentielle.

accès sécurisé

Accorder des autorisations à un service que vous spécifiez pour effectuer des tâches au sein de votre organisation AWS Organizations et dans ses comptes en votre nom. Le service de confiance crée un rôle lié au service dans chaque compte, lorsque ce rôle est nécessaire, pour effectuer des tâches de gestion à votre place. Pour plus d'informations, consultez la section Utilisation AWS Organizations avec d'autres AWS services dans la AWS Organizations documentation.

réglage

Pour modifier certains aspects de votre processus d'entraînement afin d'améliorer la précision du modèle de ML. Par exemple, vous pouvez entraîner le modèle de ML en générant un ensemble d'étiquetage, en ajoutant des étiquettes, puis en répétant ces étapes plusieurs fois avec différents paramètres pour optimiser le modèle.

équipe de deux pizzas

Une petite DevOps équipe que vous pouvez nourrir avec deux pizzas. Une équipe de deux pizzas garantit les meilleures opportunités de collaboration possible dans le développement de logiciels.

U

incertitude

Un concept qui fait référence à des informations imprécises, incomplètes ou inconnues susceptibles de compromettre la fiabilité des modèles de ML prédictifs. Il existe deux types

d'incertitude : l'incertitude épistémique est causée par des données limitées et incomplètes, alors que l'incertitude aléatoire est causée par le bruit et le caractère aléatoire inhérents aux données. Pour plus d'informations, veuillez consulter le guide Quantifying uncertainty in deep learning systems.

tâches indifférenciées

Également connu sous le nom de « levage de charges lourdes », ce travail est nécessaire pour créer et exploiter une application, mais qui n'apporte pas de valeur directe à l'utilisateur final ni d'avantage concurrentiel. Les exemples de tâches indifférenciées incluent l'approvisionnement, la maintenance et la planification des capacités.

environnements supérieurs

Voir environnement.

V

mise à vide

Opération de maintenance de base de données qui implique un nettoyage après des mises à jour incrémentielles afin de récupérer de l'espace de stockage et d'améliorer les performances.

contrôle de version

Processus et outils permettant de suivre les modifications, telles que les modifications apportées au code source dans un référentiel.

Appairage de VPC

Une connexion entre deux VPCs qui vous permet d'acheminer le trafic en utilisant des adresses IP privées. Pour plus d'informations, veuillez consulter la rubrique Qu'est-ce que l'appairage de VPC ? dans la documentation Amazon VPC.

vulnérabilités

Défaut logiciel ou matériel qui compromet la sécurité du système.

W

cache actif

Cache tampon qui contient les données actuelles et pertinentes fréquemment consultées. L'instance de base de données peut lire à partir du cache tampon, ce qui est plus rapide que la lecture à partir de la mémoire principale ou du disque.

données chaudes

Données rarement consultées. Lorsque vous interrogez ce type de données, des requêtes modérément lentes sont généralement acceptables.

fonction de fenêtre

Fonction SQL qui effectue un calcul sur un groupe de lignes liées d'une manière ou d'une autre à l'enregistrement en cours. Les fonctions de fenêtre sont utiles pour traiter des tâches, telles que le calcul d'une moyenne mobile ou l'accès à la valeur des lignes en fonction de la position relative de la ligne en cours.

charge de travail

Ensemble de ressources et de code qui fournit une valeur métier, par exemple une application destinée au client ou un processus de backend.

flux de travail

Groupes fonctionnels d'un projet de migration chargés d'un ensemble de tâches spécifique. Chaque flux de travail est indépendant, mais prend en charge les autres flux de travail du projet. Par exemple, le flux de travail du portefeuille est chargé de prioriser les applications, de planifier les vagues et de collecter les métadonnées de migration. Le flux de travail du portefeuille fournit ces actifs au flux de travail de migration, qui migre ensuite les serveurs et les applications.

VER

Voir écrire une fois, lire plusieurs.

WQF

Voir le cadre AWS de qualification de la charge de travail.

écrire une fois, lire plusieurs (WORM)

Modèle de stockage qui écrit les données une seule fois et empêche leur suppression ou leur modification. Les utilisateurs autorisés peuvent lire les données autant de fois que nécessaire,

mais ils ne peuvent pas les modifier. Cette infrastructure de stockage de données est considérée comme immuable.

Z

exploit Zero-Day

Une attaque, généralement un logiciel malveillant, qui tire parti d'une vulnérabilité de type « jour zéro ».

vulnérabilité « jour zéro »

Une faille ou une vulnérabilité non atténuée dans un système de production. Les acteurs malveillants peuvent utiliser ce type de vulnérabilité pour attaquer le système. Les développeurs prennent souvent conscience de la vulnérabilité à la suite de l'attaque.

invite Zero-Shot

Fournir à un LLM des instructions pour effectuer une tâche, mais aucun exemple (plans) pouvant aider à la guider. Le LLM doit utiliser ses connaissances pré-entraînées pour gérer la tâche. L'efficacité de l'invite zéro dépend de la complexité de la tâche et de la qualité de l'invite. Voir également les instructions en quelques clics.

application zombie

Application dont l'utilisation moyenne du processeur et de la mémoire est inférieure à 5 %. Dans un projet de migration, il est courant de retirer ces applications.

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.