Tester des applications sans serveur sur AWS

AWS Conseils prescriptifs

AWS Conseils prescriptifs: Tester des applications sans serveur sur AWS

Table of Contents

Tester des applications sans serveur sur AWS

Amazon Web Services (???contributeurs)

Mars 2026 (historique du document)

Ce guide traite des méthodologies de test des applications sans serveur, décrit les défis que vous pourriez rencontrer lors des tests et présente les bonnes pratiques. Ces techniques de test ont pour but de vous aider à itérer plus rapidement et à publier votre code en toute confiance.

Ce guide s'adresse aux développeurs qui cherchent à établir des politiques de test pour leurs applications sans serveur. Vous pouvez utiliser le guide comme point de départ pour en savoir plus sur les politiques de test et consulter le référentiel Serverless Test Samples pour découvrir des exemples de tests conformes aux modèles et aux bonnes pratiques décrits dans ce guide. Ce guide décrit les méthodologies de test sans serveur, décrit les défis auxquels les clients sont confrontés lorsqu'ils testent des applications sans serveur et présente les meilleures pratiques pour tester des applications sans serveur. Ces techniques ont pour but d'aider les développeurs à itérer plus rapidement et à publier en toute confiance.

Présentation de

Les tests automatisés constituent des investissements essentiels qui contribuent à garantir la qualité des applications et la vitesse de développement. Les tests accélèrent également les commentaires pour les développeurs. En tant que développeur, vous souhaitez pouvoir itérer rapidement sur votre application et obtenir des commentaires sur la qualité de votre code. De nombreux développeurs ont l'habitude d'écrire des applications qu'ils déploient dans un environnement sur leur bureau, soit directement sur leur système d'exploitation, soit dans un environnement basé sur des conteneurs. Lorsque vous travaillez dans des environnements de bureau ou basés sur des conteneurs, vous écrivez généralement des tests par rapport à du code entièrement hébergé sur votre bureau. Toutefois, dans les applications sans serveur, les composants de l'architecture peuvent ne pas être déployables dans un environnement de bureau, mais n'exister que dans le cloud. Une architecture basée sur le cloud peut inclure des couches de persistance, des systèmes de messagerie, des structures de sécurité et d'autres composants. APIs Lorsque vous écrivez du code d'application qui repose sur ces composants, il peut être difficile de déterminer le meilleur moyen de concevoir et d'exécuter des tests.

Ce guide vous aide à adopter une stratégie de test qui réduit les frictions et les confusions, et qui améliore la qualité du code.

Conditions préalables

Ce guide part du principe que vous connaissez les bases des tests automatisés, notamment la manière dont les tests logiciels automatisés sont utilisés pour garantir la qualité des logiciels. Le guide fournit une introduction de haut niveau à une stratégie de test d'applications sans serveur et ne nécessite aucune expérience pratique en matière de rédaction de tests.

Définitions

Ce guide utilise les termes suivants :

- Les tests unitaires sont des tests exécutés par rapport au code d'un seul composant architectural de manière isolée.
- Les tests d'intégration sont exécutés sur deux composants architecturaux ou plus, généralement dans un environnement cloud.
- End-to-end les tests vérifient les comportements dans l'ensemble des applications ou des flux de travail.
- Les émulateurs sont des applications (souvent fournies par un tiers) conçues pour imiter un service cloud sans fournir ni invoquer de ressources cloud.
- Les simulations (également appelées faux) sont des implémentations dans une application de test qui remplacent une dépendance par une simulation de cette dépendance.

Objectifs

Les bonnes pratiques présentées dans ce guide ont pour but de vous aider à atteindre deux objectifs principaux :

- Améliorer la qualité des applications sans serveur
  - Tester aux limites de l'architecture
  - Tester aux limites du code
- Diminuer le temps d'implémentation ou de modification des fonctionnalités

Améliorer la qualité de votre logiciel

La qualité d'une application dépend dans une large mesure de la capacité des développeurs à tester une variété de scénarios pour vérifier les fonctionnalités. Lorsque vous n'implémentez pas de tests automatisés ou, plus généralement, si vos tests ne couvrent pas correctement les scénarios requis, la qualité de votre application ne peut être ni déterminée ni garantie.

Dans une architecture basée sur des serveurs, les équipes sont en mesure de définir facilement une portée pour les tests. Ainsi, tout code qui s'exécute sur le serveur d'applications doit être testé. Les autres composants qui font appel au serveur, ou les dépendances que le serveur appelle, sont souvent considérés comme externes et hors de portée des tests par l'équipe responsable de l'application sur le serveur.

Les applications sans serveur consistent souvent en de petites unités de travail, telles que des fonctions AWS Lambda , qui s'exécutent dans leur propre environnement. Les équipes seront probablement responsables de plusieurs de ces plus petites unités au sein d'une même application. Certaines fonctionnalités de l'application peuvent être entièrement déléguées à des services gérés tels qu'Amazon Simple Storage Service (Amazon S3) ou Amazon Simple Queue Service (Amazon SQS) sans utiliser de code développé en interne. Les modèles traditionnels basés sur des serveurs pour les tests de logiciels peuvent exclure les services gérés en les considérant comme externes à l'application. Cela peut entraîner une couverture inadéquate, les scénarios critiques pouvant se limiter à des tests exploratoires manuels ou à quelques cas de test d'intégration dont les résultats varient en fonction de l'environnement. Par conséquent, l'adoption de politiques de test qui incluent les comportements des services gérés et les configurations cloud peut améliorer la qualité des logiciels.

Tester aux limites de l'architecture

Au fur et à mesure que les applications sans serveur se développent, elles se répartissent naturellement sur plusieurs composants architecturaux. Bien que cela utilise des fonctionnalités AWS distribuées, cela peut rendre end-to-end le comportement difficile à comprendre.

Identifier les limites naturelles

Lorsque vous concevez votre architecture conformément aux meilleures pratiques sans serveur (une fonction = une tâche, découplage), vous remarquerez des limites naturelles autour des sous-systèmes. Ces limites représentent des points de séparation logiques dans votre application.

Les limites en tant que contrats de test

Ces limites architecturales sont d'excellentes candidates pour tester les arêtes. Traitez chaque limite comme un contrat et vérifiez qu'elle se comporte conformément aux spécifications définies. Considérez ces limites comme des joints dans votre application où vous pouvez insérer une validation de test.

Principaux avantages

Les principaux avantages des tests aux limites de l'architecture sont les suivants :

- Périmètre de test ciblé — Testez les sous-systèmes de manière indépendante sans avoir à comprendre l'application dans son intégralité.
- Validation du contrat — Assurez-vous que chaque limite conserve le comportement attendu au fur et à mesure de l'évolution du système
- Instrumentation à double usage : ces mêmes limites constituent d'excellents points d'observabilité en production
- Harnais de test : permet de tester des systèmes asynchrones sans serveur. Il vous aide à tester les architectures pilotées par les événements en capturant et en validant les événements au fur et à mesure qu'ils circulent dans votre sous-système.

Tester aux limites du code

Définissez des limites de code claires en séparant le code d'infrastructure, tel que le code Lambda, de votre logique métier principale. Cette séparation crée des étendues de test distinctes qui simplifient votre stratégie de test.

Le schéma des limites

Définissez deux limites de code claires dans vos fonctions Lambda :

- Limite extérieure (gestionnaire Lambda) : couche d'adaptation fine qui gère les problèmes spécifiques à AWS Lambda
- Limite interne (logique métier) : méthodes de logique métier pures indépendantes de l'environnement d'exécution Lambda

Gestionnaire en tant qu'adaptateur (scope externe)

Votre gestionnaire de fonctions Lambda doit être une fine couche qui :

- Extrait les données des objets entrants `event` et `context` des objets
- Valide les données extraites
- Transmet uniquement les détails pertinents aux méthodes de logique métier
- Renvoie les résultats dans le format attendu pour Lambda

Logique métier (périmètre interne)

Votre logique métier de base doit :

- Fonctionne indépendamment des détails spécifiques à Lambda
- Acceptez des entrées simples et validées
- Renvoie des résultats prévisibles
- Nécessite un minimum de dépendances pour l'initialisation

Avantages des tests par périmètre

- Tests des limites internes — Tests unitaires complets autour de la logique métier sans complexité Lambda ni configuration de l'environnement
- Tests des limites extérieures — Tests d'intégration ciblés validant la gestion des événements et l'extraction des données par la couche adaptatrice
- Frais de test minimaux : aucun environnement complexe ou dépendance étendue n'est nécessaire pour la majorité de vos tests

Cette approche basée sur les limites vous permet de tester la majeure partie de votre code sous forme de fonctions pures tout en minimisant et en ciblant les tests Lambda.

Diminuer le temps d'implémentation ou de modification des fonctionnalités

Vous pouvez minimiser l'effet des bogues logiciels et des problèmes de configuration sur les coûts et les délais en détectant ces problèmes au cours d'un cycle de développement itératif. Lorsqu'un développeur ne parvient pas à détecter ces problèmes, davantage de personnes doivent investir des efforts supplémentaires pour identifier les problèmes.

Une architecture sans serveur peut comprendre des services gérés qui fournissent des fonctionnalités d'applications critiques par le biais d'appels d'API. C'est pourquoi votre cycle de développement doit inclure des tests qui valident à la fois le chemin heureux (où les interactions avec

ces services se comportent comme prévu) et le chemin triste (où les appels échouent, renvoient des réponses inattendues ou se comportent différemment selon les environnements). Sans ces tests, vous pouvez rencontrer des problèmes liés aux différences entre votre environnement local et l'environnement déployé. Dans ce cas, vous devez passer plus de temps à essayer de reproduire et de vérifier un correctif, car chaque itération nécessite désormais de valider les modifications par rapport à un environnement différent de votre configuration préférée.

Une stratégie de test sans serveur appropriée améliore le temps d'itération en fournissant des résultats précis pour les tests qui incluent des appels à d'autres services.

Techniques de tests pour les applications sans serveur

Il existe trois approches principales pour exécuter des tests sur le code d'une application sans serveur : les tests fictifs, les tests d'émulation et les tests dans le cloud. Vous pouvez généralement trouver une combinaison de ces approches utilisées dans le cadre d'un seul projet. Par exemple, vous pouvez utiliser des tests fictifs lorsque vous développez votre code localement, des tests d'émulation pour reproduire plus fidèlement le comportement du service avant le déploiement, et des tests cloud dans le cadre d'un processus d'intégration et de livraison continues (CI/CD) nocturnes.

Tests simulés

Les tests simulés sont une technique qui consiste à créer des objets de remplacement dans votre code afin de simuler le comportement des services cloud. Par exemple, vous pouvez écrire un test qui utilise une maquette du service Amazon S3, et vous pouvez configurer cette maquette pour qu'elle renvoie un objet de réponse spécifique chaque fois que la `PutObject` méthode est appelée. Lorsque le test s'exécute, la simulation renvoie la réponse sans appeler Amazon S3 ni aucun autre point de terminaison de service.

Ces objets de remplacement sont souvent générés par un cadre simulé afin de réduire le nombre d'implémentations nécessaires pour un test donné. Certains frameworks fictifs sont génériques tandis que d'autres sont conçus spécifiquement pour AWS SDKs.

Les simulations, tout comme les stubs, entrent dans la catégorie plus large des faux. Les simulacres diffèrent des émulateurs (abordés plus loin dans cette section) en ce sens que les maquettes sont généralement créées ou configurées par un développeur dans le cadre du code de test, tandis que les émulateurs sont des applications autonomes qui s'exposent APIs (comme REST APIs) de la même manière que les systèmes qu'ils émulent.

Les avantages de l’utilisation de simulations incluent les avantages suivants :

- Mocks peut simuler des services tiers échappant au contrôle de votre application, tels que APIs des fournisseurs de logiciels en tant que service (SaaS), sans avoir besoin d'un accès direct à ces services.
- Par ailleurs, les simulations sont utiles pour tester les conditions d'échec, en particulier lorsque ces conditions (comme dans le cas d'une panne de service) sont difficiles à simuler.
- Comme les émulateurs, les frameworks fictifs peuvent permettre un développement local rapide une fois qu'ils ont été configurés.

- Les simulations peuvent fournir un comportement de substitution pour pratiquement n’importe quel type d’objet. Les stratégies de simulations peuvent donc couvrir une plus grande variété de services que les émulateurs.
- Lorsque de nouvelles fonctionnalités ou de nouveaux comportements sont disponibles, les tests fictifs peuvent réagir plus rapidement en utilisant (ou en recourant à) un framework de simulation générique, capable de simuler les nouvelles fonctionnalités dès que le AWS SDK mis à jour est disponible.

Les tests simulés présentent les inconvénients suivants :

- Les simulations nécessitent généralement un effort d’installation et de configuration non négligeable, en particulier lorsqu’il s’agit de déterminer les valeurs de retour de différents services afin de simuler correctement les réponses.
- Comme les simulations sont écrites ou configurées par les développeurs qui rédigent les tests, elles représentent une responsabilité supplémentaire pour eux.
- Il se peut que vous deviez avoir accès au cloud afin de comprendre les valeurs des services APIs et de les rentabiliser.
- Les maquettes peuvent également être difficiles à gérer, car elles nécessitent des mises à jour chaque fois que les signatures d'API cloud simulées changent, que les schémas de valeur de retour évoluent ou que la logique d'application testée est étendue pour faire des appels à de nouvelles. APIs Ces modifications créent une charge de travail supplémentaire en matière de développement de tests pour les développeurs.
- Les tests fictifs peuvent réussir localement mais échouer dans le cloud, car ils simulent, au lieu de reproduire, le comportement de services réels, sans détecter les problèmes spécifiques à l'environnement.
- Les frameworks fictifs, tels que les émulateurs, ne peuvent pas détecter les violations des politiques Gestion des identités et des accès AWS (IAM), les limites de quotas de service ou les contraintes de taille de charge utile, et ils ne déclenchent pas de services auxiliaires tels GuardDuty qu'Amazon CloudWatch AWS CloudTrail ou Amazon susceptibles d'affecter le comportement des applications dans un environnement déployé.
- Bien que les simulations soient plus efficaces pour simuler ce que fera une application lorsque l'autorisation échoue ou qu'un quota est dépassé, les tests simulés ne permettent pas de déterminer le résultat réel qui se produira lorsque le code sera déployé dans l'environnement de production.

Tests d'émulation

Les tests d'émulation sont activés par des applications exécutées localement connues sous le nom d'émulateurs qui ressemblent à. Services AWS Les émulateurs sont APIs similaires à leurs homologues du cloud et fournissent des valeurs de retour similaires. Ils peuvent également simuler des changements d'état initiés par des appels d'API. Par exemple, un émulateur Amazon S3 peut stocker un objet sur le disque local lorsque la `PutObject` méthode est appelée. Lorsqu'il `GetObject` est appelé avec la même clé, l'émulateur lit le même objet sur le disque et le renvoie.

Les avantages des tests d'émulation sont les suivants :

- Ils constituent l'environnement le plus familier pour les développeurs qui ont l'habitude de développer du code exclusivement dans un environnement local. Par exemple, si vous êtes habitué au développement d'une application à n niveaux, vous pouvez avoir un moteur de base de données et un serveur Web, similaires à ceux exécutés en production, exécutés sur votre machine locale pour fournir une capacité de test rapide, locale et isolée.
- Il ne nécessite aucune modification de l'infrastructure cloud (comme les comptes cloud des développeurs), il est donc facile à mettre en œuvre avec les modèles de test existants. Les tests d'émulation offrent les avantages des itérations de développement rapides et locales.

Cependant, les émulateurs présentent plusieurs inconvénients :

- Ils sont souvent difficiles à configurer et à reproduire, en particulier lorsqu'ils sont utilisés dans le cadre de CI/CD pipelines. Cela risque d'augmenter la charge de travail et la maintenance pour le personnel informatique ou pour les développeurs qui gèrent leurs propres installations de logiciels.
- Les fonctionnalités émulées APIs sont généralement en retard par rapport aux modifications des services et peuvent entraver l'adoption de nouvelles fonctionnalités jusqu'à ce que le support soit ajouté.
- Les émulateurs peuvent nécessiter une prise en charge, des mises à jour, des corrections de bogues ou des améliorations de parité des fonctionnalités, qui relèvent de la responsabilité de l'auteur de l'émulateur (qui est souvent une entreprise tierce).
- Les tests basés sur des émulateurs peuvent fournir des résultats positifs localement, mais ils peuvent échouer dans le cloud en raison d'interactions avec d'autres aspects AWS tels que les politiques IAM et les quotas, qui ne sont généralement pas émulés.

- Certains Services AWS ne disposent pas d'émulateurs. Par conséquent, si vous vous fiez uniquement à l'émulation, il se peut que vous ne disposiez pas d'une option de test satisfaisante pour certaines parties de votre application.

Tests dans le cloud

Les tests dans le cloud sont le processus qui consiste à exécuter des tests par rapport à du code déployé dans un environnement cloud, plutôt que dans un environnement de bureau. La valeur des tests dans le cloud augmente avec le développement d'applications natives cloud. Par exemple :

- Vous pouvez exécuter des tests dans le cloud en fonction des fonctionnalités les plus récentes APIs , afin de vous assurer que vos tests reflètent les valeurs de retour et les comportements les plus récents, tout en AWS continuant à lancer de nouveaux services et fonctionnalités.
- Vos tests peuvent couvrir les politiques IAM, les quotas de service, les configurations et tous les services.
- Votre environnement de test cloud ressemble le plus étroitement à votre environnement d'exécution de production. Les tests que vous exécutez dans le cloud sont donc susceptibles d'obtenir les résultats les plus cohérents à mesure de leur progression dans les environnements.

Les inconvénients des tests dans le cloud sont les suivants :

- Les déploiements vers des environnements cloud prennent généralement plus de temps que les déploiements vers des environnements de bureau. Des outils tels qu'AWS Serverless Application Model (AWS SAM) Accelerate, le mode de surveillance AWS Cloud Development Kit (AWS CDK) et SST contribuent à réduire la latence associée aux itérations de déploiement cloud.
- Les tests dans le cloud peuvent entraîner des coûts de service supplémentaires, contrairement aux tests locaux, qui utilisent votre environnement de développement local.
- Les tests dans le cloud peuvent être moins réalisables si vous ne disposez pas d'un accès Internet haut débit.
- Dans les secteurs réglementés, les politiques de sécurité des entreprises peuvent restreindre l'accès des développeurs aux environnements cloud, ce qui rend difficile, voire impossible, l'exécution de tests cloud dans le cadre d'un flux de développement local.
- Les limites de l'environnement sont souvent tracées au niveau de la pile dans les comptes partagés destinés aux environnements de développement, parfois en utilisant des politiques de type espace de noms telles que l'utilisation de préfixes pour identifier le propriétaire. Pour les

environnements de pré-production et de production, les limites sont généralement définies au niveau du compte afin d'isoler les charges de travail de leurs bruyants voisins, de prendre en charge des contrôles de sécurité du moindre privilège et de protéger les données sensibles. L'obligation de créer des environnements isolés peut imposer une charge supplémentaire aux DevOps équipes, en particulier si elles font partie d'une entreprise qui applique des contrôles stricts sur les comptes et l'infrastructure.

Difficultés liées au test des applications sans serveur

Lorsque vous utilisez des émulateurs et des appels simulés pour tester votre application sans serveur sur votre poste de travail local, vous pouvez rencontrer des incohérences lors des tests au fur et à mesure que votre code progresse d'un environnement à l'autre dans votre pipeline. CI/CD Les tests unitaires que vous créez sur votre poste de travail pour valider la logique métier de votre application peuvent ne pas inclure ou ne pas représenter avec précision les aspects critiques des services cloud. Les tests complets ne peuvent pas être effectués localement de manière isolée. Ils nécessitent de vérifier les autorisations et les configurations entre plusieurs services.

Les sections suivantes décrivent les problèmes que vous pourriez rencontrer lors de l'implémentation d'une stratégie de test cloud. Les sections suivantes décrivent les défis auxquels les clients sont confrontés lorsqu'ils essaient de mettre en œuvre une stratégie de test dans le cloud, ainsi que nos conseils sur les meilleures pratiques pour obtenir une couverture de test efficace.

Exemple : fonction Lambda qui crée un compartiment Amazon S3

Si la logique d'une fonction Lambda dépend de la création d'un compartiment Amazon S3, un test complet doit confirmer qu'Amazon S3 a bien été appelé et que le compartiment a été créé avec succès. Dans le cas d’un test simulé, vous pouvez simuler une réponse de réussite et éventuellement ajouter un scénario de test pour gérer une réponse d’échec. Dans un scénario de test d'émulation, l'`CreateBucket`API peut être appelée, mais l'identité à l'origine de l'appel ne provient pas du fait que le service Lambda assume un rôle, et une authentification par espace réservé sera utilisée à la place. Il s'agit souvent de votre rôle ou de votre identité d'utilisateur le plus permissif.

Les configurations de simulation et d'émulation décrites précédemment vont très probablement tester ce que fera la fonction Lambda si elle appelle Amazon S3 avec succès (ou sans succès). Cependant, ces tests ne permettront pas de déterminer si la fonction Lambda est capable de créer correctement le bucket, compte tenu de la configuration de la fonction. Cette configuration est probablement représentée par l'infrastructure sous forme de code (IaC) pour des produits et services tels que AWS CloudFormation Terraform AWS SAM ou HashiCorp Terraform. L'un des problèmes possibles est que le rôle attribué à la fonction n'est pas associé à une politique autorisant l'`s3:CreateBucket`action, et que la fonction échouera donc toujours lorsqu'elle est déployée dans un environnement cloud.

Exemple : fonction Lambda qui traite les messages provenant d'Amazon SQS

Si une file d'attente Amazon Simple Queue Service (Amazon SQS) est la source d'une fonction Lambda, un test complet devrait vérifier que la fonction Lambda est correctement invoquée lorsqu'un message est placé dans une file d'attente. Les tests d'émulation et les tests fictifs sont généralement configurés pour exécuter directement le code de la fonction Lambda et pour simuler l'intégration Amazon SQS en transmettant une charge utile d'événement JSON (ou un objet désérialisé) comme entrée du gestionnaire de fonctions.

Les tests locaux qui simulent l'intégration Amazon SQS testeront ce que fera la fonction Lambda lorsqu'elle est appelée par Amazon SQS avec une charge utile donnée, mais ils ne garantissent pas qu'Amazon SQS invoquera correctement la fonction Lambda lorsqu'elle est déployée dans un environnement cloud.

Voici quelques exemples de problèmes de configuration que vous pouvez rencontrer avec Amazon SQS et Lambda :

- Le délai de visibilité d’Amazon SQS est trop court, ce qui entraîne des invocations multiples alors qu’une seule était prévue.
- Le rôle d'exécution de la fonction Lambda n'autorise pas la lecture de messages depuis la file d'attente (via `sqs:ReceiveMessage`sqs:DeleteMessage, ou`sqs:GetQueueAttributes`).
- L’exemple d’événement transmis à la fonction Lambda dépasse le quota de taille de message Amazon SQS. Par conséquent, le test n’est pas valide, car Amazon SQS ne sera jamais en mesure d’envoyer un message de cette taille.

Comme le montrent ces exemples, les tests qui couvrent la logique métier mais pas les configurations entre les services cloud sont susceptibles de fournir des résultats peu fiables.

Bonnes pratiques pour tester les applications sans serveur

Les sections suivantes décrivent les bonnes pratiques pour obtenir une couverture efficace lors du test d'applications sans serveur.

Prioriser les tests dans le cloud

Pour des applications bien conçues, vous pouvez utiliser diverses techniques de test pour répondre à un éventail d'exigences et de conditions. Cependant, sur la base des outils actuels, nous vous recommandons de vous concentrer autant que possible sur les tests dans le cloud. Bien que les tests dans le cloud puissent créer de la latence pour les développeurs, augmenter les coûts et parfois nécessiter des investissements dans DevOps des contrôles supplémentaires, cette technique fournit la couverture de test la plus fiable, précise et complète.

Vous devez avoir accès à des environnements isolés dans lesquels vous pouvez effectuer les tests. Idéalement, chaque développeur devrait disposer d'un service dédié Compte AWS afin d'éviter tout problème de dénomination des ressources qui peut survenir lorsque plusieurs développeurs travaillant sur le même code tentent de déployer ou d'invoquer des appels d'API sur des ressources portant des noms identiques. Ces environnements doivent être configurés avec les alertes et les contrôles appropriés afin d'éviter des dépenses inutiles. Par exemple, vous pouvez limiter le type, le niveau ou la taille des ressources qui peuvent être créées, et configurer des alertes par e-mail lorsque les coûts estimés dépassent un seuil donné.

Si vous devez partager un single Compte AWS avec d'autres développeurs, les processus de test automatisés doivent nommer les ressources de manière à ce qu'elles soient uniques pour chaque développeur. Par exemple, les scripts de mise à jour ou les fichiers de configuration TOML à l'origine des commandes AWS SAM CLI sam deploy ou sam sync peuvent spécifier automatiquement un nom de pile incluant le nom d'utilisateur du développeur local.

Les tests dans le cloud sont utiles pour toutes les phases de test, y compris les tests unitaires, les tests d'intégration et les end-to-end tests.

Utiliser des simulations si nécessaire

Les cadres simulés sont un outil précieux pour écrire des tests unitaires rapides. Ils sont particulièrement utiles lorsque les tests doivent couvrir une logique interne complexe, comme

des calculs mathématiques ou financiers ou des simulations. Recherchez les tests unitaires qui comportent un grand nombre de cas de test ou de variations d'entrée, dans lesquels les entrées ne modifient ni le modèle ni le contenu des appels vers d'autres services cloud. La création de tests de simulation pour ces scénarios peut améliorer les temps d'itération des développeurs.

Le code couvert par des tests unitaires qui utilisent des tests de simulation doit également être couvert par des tests dans le cloud. Cela est nécessaire car les maquettes fonctionnent toujours sur l'ordinateur portable ou sur la machine de construction d'un développeur, et l'environnement peut être configuré différemment de ce qu'il sera dans le cloud. Par exemple, votre code peut inclure des AWS Lambda fonctions qui utilisent plus de mémoire ou prennent plus de temps que ce que Lambda est configuré pour allouer lorsqu'il est exécuté avec certains paramètres d'entrée. Votre code peut également inclure des variables d'environnement qui ne sont pas configurées de la même manière (ou pas du tout), et les différences peuvent entraîner un comportement différent ou un échec du code.

N'utilisez pas de simulacres de services cloud pour valider la bonne mise en œuvre de ces intégrations de services. Bien qu'il soit acceptable de simuler un service cloud lorsque vous testez d'autres fonctionnalités, vous devez tester les appels de service cloud dans le cloud pour valider la configuration et l'implémentation fonctionnelle correctes.

Les simulations peuvent apporter une valeur ajoutée aux tests unitaires, en particulier lorsque vous testez fréquemment un grand nombre de cas. Cet avantage est réduit pour les tests d'intégration, car le niveau d'effort nécessaire pour implémenter les simulations nécessaires augmente avec le nombre de points de connexion. End-to-endles tests ne doivent pas utiliser de simulations, car ces tests portent généralement sur des états et une logique complexe qui ne peuvent pas être facilement simulés avec des frameworks fictifs.

Comprenez les inconvénients des tests d'émulation

Les émulateurs peuvent être un choix pratique pour des cas d'utilisation spécifiques. Par exemple, une équipe de développement dont l'accès à Internet est limité, incohérent ou lent peut trouver que les tests d'émulation constituent le moyen le plus fiable d'itérer sur le code avant de passer à un environnement cloud.

Dans la plupart des autres cas, utilisez les émulateurs de manière sélective. Lorsque vous comptez beaucoup sur un émulateur, il peut s'avérer difficile d'intégrer de nouvelles fonctionnalités de AWS service à vos tests jusqu'à ce que le fournisseur d'émulation publie une mise à jour garantissant la parité des fonctionnalités. Les émulateurs nécessitent également un investissement initial et continu pour l'installation et la configuration des systèmes de développement et des machines de

construction. En outre, de nombreux services cloud ne disposent pas d'émulateurs ; le choix d'une stratégie privilégiant l'émulation peut soit empêcher l'utilisation de ces services, soit produire du code et des configurations qui ne sont pas correctement testés par rapport au comportement réel des services.

Si vous utilisez des tests d'émulation, complétez-les autant que possible par des tests sur le cloud afin de valider que les configurations cloud appropriées sont en place et de tester les interactions avec des services qui ne peuvent être simulés ou simulés que dans un environnement émulé.

Les tests d'émulation peuvent fournir des informations rapides pour les tests unitaires et, en fonction des fonctionnalités et de la parité comportementale du logiciel d'émulation, peuvent également prendre en charge certaines intégrations et certains end-to-end tests.

Tests de portée à travers les limites naturelles

À mesure que les applications sans serveur intègrent de plus en plus de composants architecturaux, des limites naturelles apparaissent autour des sous-systèmes, en particulier lorsque l'on suit les meilleures pratiques telles que les fonctions à usage unique et le découplage piloté par les événements. Ces limites constituent des zones de test efficaces où vous pouvez valider les contrats entre les composants.

Identifier les limites de l'architecture

Recherchez des coutures naturelles dans la conception de votre application :

- Entre les services, comme une EventBridge règle Amazon reliant un éditeur à un consommateur
- À la périphérie des API, tels que les points de terminaison Amazon API Gateway qui font face aux fonctions Lambda
- Autour des flux de travail, tels que l' AWS Step Functions orchestration de plusieurs services
- Au niveau des couches de stockage, telles que les flux Amazon DynamoDB déclenchant un traitement en aval

Séparer le code Lambda de la logique métier

Simplifiez vos tests en isolant le code Lambda de la logique métier principale. Votre gestionnaire Lambda doit agir comme un adaptateur léger entre le AWS runtime et la logique de votre application. Il doit extraire et valider les données des événements, puis les déléguer à une fonction testable qui

n'a aucune dépendance Lambda. Cela rend votre logique métier portable, plus facile à raisonner et simple à tester sans vous moquer des objets Lambda ou configurer des environnements complexes.

Traitez les limites comme des contrats

Testez à la limite, pas à travers elle. Validez ce qui franchit la limite sans avoir besoin de l'ensemble du système en aval. Ces mêmes limites servent également de points d'observabilité dans la production. Les joints architecturaux que vous testez peuvent être instrumentés pour être surveillés à l'aide d'Amazon CloudWatch Logs, de AWS X-Ray traces et d' EventBridgeévénements.

Utiliser des harnais de test pour les flux de travail asynchrones

Les applications sans serveur reposent souvent sur des modèles asynchrones, dans lesquels les événements déclenchent le traitement, les messages circulent dans les files d'attente et les flux de travail couvrent plusieurs services sans réponse immédiate. Vous ne pouvez pas simplement invoquer une fonction et inspecter une valeur de retour. Le résultat peut apparaître ultérieurement dans une base de données, un flux de journal ou un autre service.

Un harnais de test consiste à tester l'infrastructure que vous déployez parallèlement à votre application afin d'observer et de valider ce comportement asynchrone. Les harnais de test incluent généralement :

- Des écouteurs d'événements qui s'abonnent aux mêmes événements que ceux produits par votre application
- Mécanismes de stockage (tels que des tables DynamoDB ou des compartiments Amazon S3) permettant de capturer les résultats des tests
- Logique d'interrogation dans votre code de test qui attend l'apparition des résultats attendus

Votre code de test déclenche un événement, attend la fin du flux de travail, puis interroge le harnais de test pour vérifier que le résultat attendu s'est produit.

Voici les bonnes pratiques associées :

- Définissez clairement SLAs les opérations asynchrones : déterminez la durée que devraient prendre les flux de travail et utilisez-les comme délais d'expiration des interrogations dans vos tests
- Utiliser des identifiants uniques pour isoler les tests : générez des noms de fichiers, des messages IDs ou des jetons de corrélation uniques par test afin d'éviter toute interférence entre les tests

- Déployez une infrastructure de test parallèlement à votre application : incluez des ressources de harnais de test dans vos infrastructure-as-code modèles afin qu'elles restent synchronisées au fur et à mesure de l'évolution de votre application
- Nettoyez les données de test après les tests : cela évite l'accumulation d'artefacts de test dans votre environnement cloud

Les harnais de test sont particulièrement utiles pour les tests d'intégration qui valident les flux de travail sur plusieurs services, les end-to-end tests qui vérifient les parcours utilisateurs complets et les architectures axées sur les événements dans lesquelles les services communiquent via Amazon EventBridge SNS, Amazon SQS ou Amazon Kinesis.

Organisez les environnements cloud pour isoler les développeurs

Les tests dans le cloud nécessitent des environnements isolés les uns des autres. Lorsque les développeurs partagent un compte unique Compte AWS, tel qu'un compte de développement d'équipe, envisagez de créer une pile d'applications distincte pour chaque développeur ou branche de fonctionnalité. Cela permet d'isoler les ressources, d'éviter les collisions de noms et d'éviter les conflits de quotas ou les problèmes de voisinage bruyants lors des tests.

Utilisez AWS Systems Manager le Parameter Store ou un outil similaire pour gérer les configurations spécifiques aux piles, telles que les points de terminaison d'API et les noms de files d'attente. Pour des raisons de rentabilité, partagez des ressources coûteuses telles que les clusters Amazon Relational Database Service (Amazon RDS) entre les équipes de développeurs tout en maintenant les ressources légères sans serveur (telles que les fonctions Lambda, les stages API Gateway et les tables DynamoDB) isolées par pile.

Dans les secteurs réglementés, les politiques de sécurité des entreprises peuvent restreindre l'accès des développeurs aux environnements cloud, ce qui complique l'exécution de tests cloud dans le cadre d'un flux de travail de développement local. Dans ces cas, les tests d'émulation peuvent combler le vide entre les tests simulés locaux et la validation complète dans le cloud, mais ils doivent être complétés par des tests dans le cloud chaque fois que l'accès le permet.

Accélérer les boucles de rétroaction

Lorsque vous effectuez des tests dans le cloud, utilisez les outils et les techniques qui vous permettent d'accélérer les boucles de rétroaction de développement. Par exemple, utilisez le mode AWS SAM Accelerate et le mode AWS CDK Watch pour réduire le temps nécessaire pour

transférer les modifications de code vers un environnement cloud. Les exemples du référentiel GitHub Serverless Test Samples explorent certaines de ces techniques.

Nous vous recommandons également de créer et de tester les ressources cloud à partir de votre machine locale le plus tôt possible pendant le développement, et pas seulement après l'enregistrement auprès du contrôle de source. Cette pratique permet d’accélérer l’exploration et l’expérimentation lors du développement de solutions. En outre, la possibilité d'automatiser le déploiement à partir d'une machine de développement vous permet de découvrir plus rapidement les problèmes de configuration du cloud et de réduire les efforts inutiles liés aux mises à jour et à l'approbation des modifications du contrôle de source.

FAQ

J'ai une fonction Lambda qui effectue des calculs et renvoie un résultat sans appeler aucun autre service. Dois-je vraiment la tester dans le cloud ?

Oui AWS Lambda les fonctions ont des paramètres de configuration susceptibles de modifier le résultat du test. Tout le code de fonction Lambda dépend du délai d'expiration et des paramètres de mémoire, ce qui peut entraîner l'échec de la fonction s'ils ne sont pas définis correctement. Les politiques Lambda permettent également la journalisation des sorties standard sur Amazon. CloudWatch Même si votre code n'appelle pas CloudWatch directement, une autorisation est requise pour activer la journalisation, et cette autorisation ne peut pas être simulée ou émulée avec précision.

Comment les tests dans le cloud peuvent-ils faciliter les tests unitaires ? S'il se trouve dans le cloud et qu'il se connecte à d'autres ressources, n'est-ce pas un test d'intégration ?

Nous définissons les tests unitaires comme des tests qui opèrent sur des composants architecturaux de manière isolée. Cette définition n'exclut pas nécessairement l'utilisation d'appels de service ou d'autres communications réseau.

De nombreuses applications sans serveur possèdent des composants architecturaux qui peuvent être testés de manière isolée, même dans le cloud. Un exemple simple est une fonction Lambda qui prend des entrées, les interprète et envoie un message à une file d'attente Amazon Simple Queue Service (Amazon SQS). Un test unitaire d'une telle fonction devrait permettre de vérifier si les valeurs d'entrée entraînent la présence de certaines valeurs dans le message en file d'attente. Prenons l'exemple d'un test écrit en utilisant le modèle arrange, act, assert :

- Organiser — Allouez des ressources (une file d'attente pour recevoir les messages et la fonction testée).
- Agir — Appelez la fonction en cours de test.
- Assert — Récupère le message envoyé par la fonction et valide le résultat.

Une approche de test simulé consiste à simuler la file d’attente avec un objet simulé en cours de traitement et à créer une instance en cours de traitement de la classe ou du module contenant le code de la fonction Lambda. Pendant la phase d'assertion, le message en file d'attente serait extrait de l'objet simulé.

Dans une approche basée sur le cloud, le test crée une file d’attente Amazon SQS pour les besoins du test et déploie la fonction Lambda avec des variables d’environnement configurées pour utiliser la file d’attente Amazon SQS isolée comme destination de sortie. Après avoir exécuté la fonction Lambda, le test récupère le message dans la file d’attente Amazon SQS.

Le test basé sur le cloud exécuterait le même code, affirmerait le même comportement et validerait l'exactitude fonctionnelle de l'application. Cependant, cela aurait l'avantage supplémentaire de pouvoir valider les paramètres suivants de la fonction Lambda : le rôle Gestion des identités et des accès AWS (IAM), les politiques IAM, le délai d'expiration et les paramètres de mémoire de la fonction.

Prochaines étapes et ressources

Utilisez les ressources suivantes pour en savoir plus et pour consulter d'autres exemples concrets.

Exemples d’implémentations

Le référentiel Serverless Test Samples GitHub contient des exemples concrets de tests qui suivent les modèles et les meilleures pratiques décrits dans ce guide. Le référentiel contient des exemples de code et des descriptions guidées des processus de simulation, d’émulation et de test dans le cloud décrits dans les sections précédentes. Utilisez ce référentiel pour vous familiariser avec les derniers conseils de test sans serveur publiés par AWS.

Suggestions de lecture

Visitez Serverless Land pour accéder aux derniers blogs, vidéos et formations sur les technologies AWS sans serveur.

Références

- Accélérer le développement sans serveur avec AWS SAM Accelerate (article de AWS blog)
- Accroître la vitesse de développement avec CDK Watch (article de AWS blog)
- Simulation d'intégrations de services avec AWS Step Functions Local (AWS article de blog)
- Commencer à tester des applications sans serveur (article de AWS blog)
- Accélérez les tests sans serveur grâce à LocalStack l'intégration dans VS Code IDE (article de AWS blog)
- Déboguer localement les fonctions avec AWS SAM (AWS documentation)

Outils

- AWS SAM — Tester et déboguer des applications sans serveur
- AWS SAM — Intégration aux tests automatisés
- AWS Lambda — Tester les fonctions Lambda dans la console
- LocalStack Cloud Emulator — Améliorez l'expérience de test local pour les applications sans serveur avec LocalStack

Collaborateurs

Conception

- Dan Fox, AWS
- Leslie Raj, AWS
- Rohan Mehta, AWS
- Rob Hill, AWS

Révision

- Brian Krygsman, AWS

Rédaction technique

- Lilly AbouHarb, AWS

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
Mises à jour	Nous avons ajouté des conseils sur les tests aux limites de l'architecture et du code, les harnais de test pour les flux de travail asynchrones et l'isolation de l'environnement des développeurs. Nous avons également mis à jour les recommandations relatives aux tests d'émulation.	18 mars 2026
Publication initiale	—	9 décembre 2022

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 sur site vers Oracle sur une instance EC2 dans le. 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, c'est 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

Voir 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 étapes 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 connus sous le nom de 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 AWS avec le service de migration 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 peuvent accéder au contenu d'un compartiment S3 uniquement via 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 informations de contact.

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.

policy

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.

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).

RAG

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

Un 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 en matière AWS de sécurité. Parmi les actions de réponse automatique, citons la modification d'un groupe de sécurité VPC, l'application de correctifs à une instance Amazon EC2 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, voir 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

tags

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.