Cadre du cycle de vie de résilience

AWS Directives prescriptives

AWS Directives prescriptives: Cadre du cycle de vie de résilience

Table of Contents

Introduction 1
Termes et définitions 2
Résilience continue 3
Étape 1 : Fixer des objectifs 4
Cartographie des applications critiques 4
Cartographie des témoignages d'utilisateurs 5
Définition des mesures 6
Création de mesures supplémentaires 7
Étape 2 : Conception et mise en œuvre 8
AWS Framework Well-Architected 8
Comprendre les dépendances 9
Stratégies de reprise après sinistre 9
Définition des stratégies CI/CD 10
Conduite des ORR 12
Comprendre les limites d'isolation des AWS pannes 12
Sélection des réponses 12
Modélisation de résilience 13
Échouer en toute 14
Étape 3 : Évaluer et tester 15
Activités préalables au déploiement 15
Conception de l'environnement 15
Tests d'intégration 16
Pipelines de déploiement automatisés 16
Test de charge 17
Activités postérieures au déploiement 17
Réalisation d'évaluations de résilience 18
tests de reprise après sinistre 18
Détection des écarts 18
Tests synthétiques 19
Ingénierie du chaos 19
Étape 4 : opérer 21
Observabilité 21
Gestion d'événements 22
Résilience continue 22

Étape 5 : Réagir et apprendre 24
Création de rapports d'analyse des incidents 24
Réalisation d'examens opérationnels 25
Examen des performances des alarmes 26
Précision de l'alarme 26
Faux positifs 27
Faux négatifs 27
Alertes dupliquées 27
Réalisation d'examens des métriques 27
Fournir des formations et des habilitations 28
Création d'une base de connaissances sur les incidents 28
Mettre en œuvre la résilience en profondeur 29
Conclusion et ressources 30
Collaborateurs 31
Historique du document 32
Glossaire 33
..... 33
A 34
B 37
C 39
D 42
E 47
F 49
G 50
H 51
I 52
L 55
M 56
O 60
P 63
Q 66
R 66
S 69
T 73
U 74
V 75

W 75
Z 77
..... lxxviii

Cadre du cycle de vie de la résilience : une approche continue de l'amélioration de la résilience

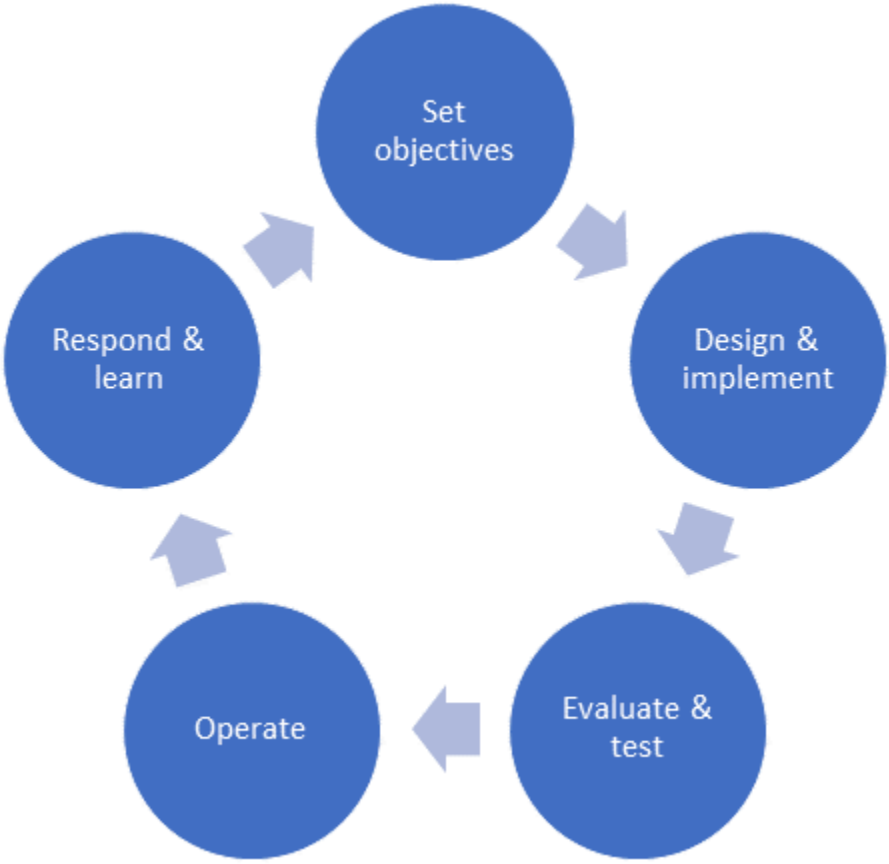
Amazon Web Services (contributeurs)

Octobre 2023 (historique du document)

Les entreprises modernes sont aujourd'hui confrontées à un nombre croissant de défis liés à la résilience, d'autant plus que les attentes des clients évoluent vers un état d'esprit toujours actif et toujours disponible. Les équipes distantes et les applications complexes et distribuées sont associées à un besoin croissant de versions fréquentes. Par conséquent, une organisation et ses applications doivent être plus résilientes que jamais.

AWS définit la résilience comme la capacité d'une application à résister aux perturbations ou à se rétablir après celles-ci, notamment celles liées à l'infrastructure, aux services dépendants, aux mauvaises configurations et aux problèmes de réseau transitoires. (Voir Resiliency et les composants de la fiabilité dans la documentation du pilier de fiabilité du AWS Well-Architected Framework.) Cependant, pour atteindre le niveau de résilience souhaité, des compromis sont souvent nécessaires. La complexité opérationnelle, la complexité de l'ingénierie et les coûts devront être évalués et ajustés en conséquence.

Sur la base d'années de travail avec les clients et les équipes internes, AWS a développé un cadre de cycle de vie de résilience qui capture les enseignements et les meilleures pratiques en matière de résilience. Le cadre décrit cinq étapes clés illustrées dans le schéma suivant. À chaque étape, vous pouvez utiliser des stratégies, des services et des mécanismes pour améliorer votre posture de résilience.



Ces étapes sont décrites dans les sections suivantes de ce guide :

- Étape 1 : Fixer des objectifs
- Étape 2 : Conception et mise en œuvre
- Étape 3 : Évaluer et tester
- Étape 4 : opérer
- Étape 5 : Réagir et apprendre

Termes et définitions

Les concepts de résilience de chaque étape sont appliqués à différents niveaux, qu'il s'agisse de composants individuels ou de systèmes complets. La mise en œuvre de ces concepts nécessite une définition claire de plusieurs termes :

- Un composant est un élément qui exécute une fonction et comprend des ressources logicielles et technologiques. Les exemples de composants incluent la configuration du code, l'infrastructure

telle que le réseau, ou même les serveurs, les magasins de données et les dépendances externes telles que les dispositifs d'authentification multifactorielle (MFA).

- Une application est un ensemble de composants qui apportent une valeur commerciale, tels qu'une vitrine Web destinée aux clients ou le processus principal qui améliore les modèles d'apprentissage automatique. Une application peut consister en un sous-ensemble de composants dans un seul AWS compte, ou il peut s'agir d'un ensemble de plusieurs composants répartis sur plusieurs Comptes AWS régions.
- Un système est un ensemble d'applications, de personnes et de processus nécessaires à la gestion d'une fonction commerciale donnée. Il englobe l'application requise pour exécuter une fonction, les processus opérationnels tels que l'intégration continue et la livraison continue (CI/CD), l'observabilité, la gestion de la configuration, la réponse aux incidents et la reprise après sinistre, ainsi que les opérateurs qui gèrent ces tâches.
- Une interruption est un événement qui empêche votre application de fonctionner correctement.
- La déficience est l'effet qu'une interruption a sur une application si elle n'est pas atténuée. Les applications peuvent être perturbées si elles subissent une série de perturbations.

Résilience continue

Le cycle de vie de la résilience est un processus continu. Même au sein d'une même organisation, vos équipes de candidature peuvent atteindre des niveaux d'exhaustivité différents au cours de chaque étape, en fonction des exigences de votre candidature. Cependant, plus chaque étape est complète, plus le niveau de résilience de votre application sera élevé.

Vous devez considérer le cycle de vie de la résilience comme un processus standard que votre organisation peut mettre en œuvre. AWS a délibérément modélisé le cycle de vie de résilience de manière à ce qu'il soit similaire au cycle de développement logiciel (SDLC), dans le but d'intégrer la planification, les tests et l'apprentissage tout au long des processus d'exploitation pendant que vous développez et exploitez vos applications. Comme c'est le cas pour de nombreux processus de développement agiles, le cycle de vie de la résilience peut être répété à chaque itération du processus de développement. Nous vous recommandons d'approfondir progressivement les pratiques à chaque étape du cycle de vie.

Étape 1 : Fixer des objectifs

Comprendre quel niveau de résilience est nécessaire et comment vous allez le mesurer constitue la base de l'étape de définition des objectifs. Il est difficile d'améliorer quelque chose si vous n'avez pas d'objectif et si vous ne pouvez pas le mesurer.

Toutes les applications n'ont pas besoin du même niveau de résilience. Lorsque vous définissez des objectifs, considérez le niveau requis afin de faire les bons investissements et de faire les bons compromis. Une bonne analogie est celle d'une voiture : elle a quatre pneus mais un seul pneu de secours. Le risque de crevaison de plusieurs pneus pendant un trajet est faible, et le fait d'avoir des pièces de rechange supplémentaires pourrait nuire à d'autres caractéristiques, telles que l'espace de chargement ou le rendement énergétique. Il s'agit donc d'un compromis raisonnable.

Après avoir défini les objectifs, vous implémentez des contrôles d'observabilité lors des étapes ultérieures (étape 2 : conception et mise en œuvre et étape 4 : exploitation) pour déterminer si les objectifs sont atteints.

Cartographie des applications critiques

La définition des objectifs de résilience ne doit pas être exclusivement une conversation technique. Commencez plutôt par vous concentrer sur les besoins de l'entreprise afin de comprendre ce que l'application doit apporter et les conséquences d'une dépréciation. Cette compréhension des objectifs commerciaux se répercute ensuite sur des domaines tels que l'architecture, l'ingénierie et les opérations. Les objectifs de résilience que vous définissez peuvent être appliqués à toutes vos applications, mais la manière dont les objectifs sont mesurés varie souvent en fonction de la fonction de l'application. Vous exécutez peut-être une application essentielle à l'entreprise, et si cette application est altérée, votre entreprise risque de perdre des revenus importants ou de porter atteinte à sa réputation. Il se peut également que vous ayez une autre application qui n'est pas aussi critique et qui peut tolérer certains temps d'arrêt sans nuire à la capacité de votre entreprise à mener ses activités.

Prenons l'exemple d'une application de gestion des commandes pour une entreprise de vente au détail. Si les composants de l'application de gestion des commandes sont défectueux et ne fonctionnent pas correctement, les nouvelles ventes ne seront pas conclues. Cette entreprise de vente au détail possède également un café pour ses employés situé dans l'un de ses bâtiments. Le café dispose d'un menu en ligne auquel les employés peuvent accéder sur une page Web statique. Si cette page Web n'est plus disponible, certains employés peuvent se plaindre, mais cela

ne causera pas nécessairement un préjudice financier à l'entreprise. Sur la base de cet exemple, l'entreprise choisira probablement de fixer des objectifs de résilience plus ambitieux pour l'application de gestion des commandes, mais n'investira pas de manière significative pour garantir la résilience de l'application Web.

Identifier les applications les plus critiques, les domaines dans lesquels déployer le plus d'efforts et les domaines dans lesquels il convient de faire des compromis est aussi important que de pouvoir mesurer la résilience d'une application en production. Pour mieux comprendre l'impact de la dépréciation, vous pouvez effectuer une analyse d'impact commercial (BIA). Un BIA fournit une approche structurée et systématique pour identifier et hiérarchiser les applications métier critiques, évaluer les risques et les impacts potentiels, et identifier les dépendances sous-jacentes. Le BIA permet de quantifier le coût des interruptions de service pour les applications les plus importantes de votre entreprise. Cette métrique permet de déterminer combien cela coûtera si une application spécifique est défectueuse et incapable de remplir sa fonction. Dans l'exemple précédent, si l'application de gestion des commandes est défectueuse, le commerce de détail pourrait perdre des revenus importants.

Cartographie des témoignages d'utilisateurs

Au cours du processus BIA, vous pouvez découvrir qu'une application est responsable de plusieurs fonctions commerciales ou qu'une fonction métier nécessite plusieurs applications. En utilisant l'exemple précédent d'une entreprise de vente au détail, la fonction de gestion des commandes peut nécessiter des applications distinctes pour le paiement, les promotions et les prix. Si une application échoue, l'impact peut être ressenti par l'entreprise et par les utilisateurs qui interagissent avec l'entreprise. Par exemple, il se peut que l'entreprise ne soit pas en mesure d'ajouter de nouvelles commandes, de donner accès à des promotions et à des remises, ou de mettre à jour le prix de ses produits. Ces différentes fonctions requises par la fonction de gestion des commandes peuvent reposer sur plusieurs applications. Ces fonctions peuvent également avoir de multiples dépendances externes, ce qui rend trop complexe le processus de mise en œuvre d'une résilience uniquement axée sur les composants. Une meilleure façon de gérer ce scénario est de se concentrer sur les user stories, qui décrivent l'expérience à laquelle les utilisateurs s'attendent lorsqu'ils interagissent avec une application ou un ensemble d'applications.

En vous concentrant sur les témoignages d'utilisateurs, vous pouvez comprendre quels aspects de l'expérience client sont les plus importants, afin de créer des mécanismes de protection contre des menaces spécifiques. Dans l'exemple précédent, l'une des histoires d'utilisateur pourrait être Checkout, qui implique l'application de paiement et dépend de l'application de tarification. Une autre

histoire d'utilisateur pourrait être la visualisation de promotions, ce qui implique l'application de promotion. Après avoir cartographié les applications les plus critiques et leurs user stories, vous pouvez commencer à définir les métriques que vous utiliserez pour mesurer la résilience de ces user stories. Ces indicateurs peuvent être appliqués à l'ensemble d'un portefeuille ou à des témoignages d'utilisateurs individuels.

Définition des mesures

Les objectifs de point de reprise (RPO), les objectifs de temps de restauration (RTO) et les objectifs de niveau de service (SLO) sont des mesures standard du secteur utilisées pour évaluer la résilience d'un système donné. Le RPO fait référence à l'ampleur des pertes de données que l'entreprise peut tolérer en cas de panne, tandis que le RTO est une mesure de la rapidité avec laquelle une application doit être à nouveau disponible après une panne. Ces deux mesures sont mesurées en unités de temps : secondes, minutes et heures. Vous pouvez également mesurer le temps pendant lequel l'application fonctionne correctement, c'est-à-dire qu'elle exécute ses fonctions comme prévu et qu'elle est accessible à ses utilisateurs. Ces SLO détaillent le niveau de service attendu des clients et sont mesurés par des indicateurs tels que le pourcentage (%) de demandes traitées sans erreur dans un délai de réponse inférieur à une seconde (par exemple, 99,99 % des demandes recevront une réponse chaque mois). Le RPO et le RTO sont liés aux stratégies de reprise après sinistre, en supposant que le fonctionnement des applications et les processus de restauration seront interrompus, qu'il s'agisse de restaurer des sauvegardes ou de rediriger le trafic utilisateur. Les SLO sont résolus par la mise en œuvre de contrôles de haute disponibilité, qui ont tendance à réduire les temps d'arrêt d'une application.

Les métriques SLO sont couramment utilisées dans la définition des accords de niveau de service (SLA), qui sont des contrats entre les fournisseurs de services et les utilisateurs finaux. Les SLA s'accompagnent généralement d'engagements financiers et décrivent les pénalités qui doivent être payées par le fournisseur si ces accords ne sont pas respectés. Cependant, un SLA n'est pas une mesure de votre posture de résilience, et l'augmentation d'un SLA ne rend pas votre application plus résiliente.

Vous pouvez commencer à définir vos objectifs en fonction des SLO, des RPO et des RTO. Une fois que vous avez défini vos objectifs de résilience et que vous avez bien compris vos objectifs de RPO et de RTO, vous pouvez effectuer une évaluation de votre architecture AWS Resilience Hubafin de découvrir les faiblesses potentielles liées à la résilience. AWS Resilience Hub évalue une architecture d'application AWS par rapport aux meilleures pratiques de Well-Architected Framework et partage des conseils de correction dans le contexte de ce qui doit être spécifiquement amélioré pour atteindre les objectifs de RTO et de RPO que vous avez définis.

Création de mesures supplémentaires

Le RPO, le RTO et les SLO sont de bons indicateurs de résilience, mais vous pouvez également réfléchir aux objectifs d'un point de vue commercial et définir des objectifs en fonction des fonctions de votre application. Par exemple, votre objectif pourrait être le suivant : les commandes réussies par minute resteront supérieures à 98 % si le temps de latence entre mon frontend et mon backend augmente de 40 %.Ou : les flux démarrés par seconde resteront dans les limites d'un écart type par rapport à la moyenne, même en cas de perte d'un composant spécifique. Vous pouvez également créer des objectifs pour réduire le temps moyen de restauration (MTTR) pour les types de défaillances connus ; par exemple : les temps de restauration seront réduits de x % si l'un de ces problèmes connus se produit. La création d'objectifs correspondant aux besoins de l'entreprise vous aide à anticiper les types de défaillances que votre application devrait tolérer. Il vous aide également à identifier les approches permettant de réduire le risque de détérioration de votre application.

Si vous pensez à l'objectif de continuer à fonctionner si vous perdez 5 % des instances qui alimentent votre application, vous pouvez déterminer que votre application doit être prédimensionnée ou qu'elle doit être capable d'évoluer suffisamment rapidement pour prendre en charge le trafic supplémentaire généré par cet événement. Vous pouvez également décider de tirer parti de différents modèles architecturaux, comme décrit dans la section Étape 2 : Conception et mise en œuvre.

Vous devez également mettre en œuvre des mesures d'observabilité pour vos objectifs commerciaux spécifiques. Par exemple, vous pouvez suivre le taux de commande moyen, le prix moyen des commandes, le nombre moyen d'abonnements ou d'autres indicateurs qui peuvent fournir des informations sur la santé de l'entreprise en fonction du comportement de votre application. En implémentant des fonctionnalités d'observabilité pour votre application, vous pouvez créer des alarmes et prendre des mesures si ces mesures dépassent les limites que vous avez définies. L'observabilité est abordée plus en détail dans la section Étape 4 : Fonctionnement.

Étape 2 : Conception et mise en œuvre

À l'étape précédente, vous avez défini vos objectifs de résilience. À présent, au stade de la conception et de la mise en œuvre, vous essayez d'anticiper les modes de défaillance et d'identifier les choix de conception, en vous basant sur les objectifs que vous avez définis à l'étape précédente. Vous définissez également des stratégies de gestion du changement et développez le code logiciel et la configuration de l'infrastructure. Les sections suivantes mettent en évidence les AWS meilleures pratiques à prendre en compte lors de la prise en compte de compromis tels que le coût, la complexité et les frais opérationnels.

AWS Framework Well-Architected

Lorsque vous concevez votre application en fonction des objectifs de résilience souhaités, vous devez évaluer plusieurs facteurs et faire des compromis sur l'architecture la plus optimale. Pour créer une application hautement résiliente, vous devez prendre en compte les aspects liés à la conception, à la création et au déploiement, à la sécurité et aux opérations. Le AWS Well-Architected Framework fournit un ensemble de meilleures pratiques, de principes de conception et de modèles architecturaux pour vous aider à concevoir des applications résilientes. AWS Les six piliers du AWS Well-Architected Framework fournissent les meilleures pratiques pour concevoir et exploiter des systèmes résilients, sécurisés, efficaces, rentables et durables. Le framework fournit un moyen de mesurer systématiquement vos architectures par rapport aux meilleures pratiques et d'identifier les domaines à améliorer.

Voici des exemples de la manière dont le AWS Well-Architected Framework peut vous aider à concevoir et à mettre en œuvre des applications répondant à vos objectifs de résilience :

- Le pilier de fiabilité : le pilier de fiabilité met l'accent sur l'importance de créer des applications capables de fonctionner correctement et de manière cohérente, même en cas de panne ou de perturbation. Par exemple, le AWS Well-Architected Framework vous recommande d'utiliser une architecture de microservices pour réduire et simplifier vos applications, afin de pouvoir différencier les besoins de disponibilité des différents composants de votre application. Vous trouverez également des descriptions détaillées des meilleures pratiques pour créer des applications en utilisant la régulation, les nouvelles tentatives avec arrêt exponentiel, l'échec rapide (délestage), l'idempuissance, le travail constant, les disjoncteurs et la stabilité statique.
- Examen complet : Le AWS Well-Architected Framework encourage un examen complet de votre architecture par rapport aux meilleures pratiques et aux principes de conception. Il permet de mesurer régulièrement vos architectures et d'identifier les domaines à améliorer.

- Gestion des risques : Le AWS Well-Architected Framework vous aide à identifier et à gérer les risques susceptibles d'avoir un impact sur la fiabilité de votre application. En abordant les scénarios de défaillance potentiels de manière proactive, vous pouvez réduire leur probabilité ou la détérioration qui en résulte.
- Amélioration continue : La résilience est un processus continu, et le AWS Well-Architected Framework met l'accent sur l'amélioration continue. En révisant et en affinant régulièrement votre architecture et vos processus en fonction des directives du AWS Well-Architected Framework, vous pouvez vous assurer que vos systèmes restent résilients face à l'évolution des défis et des exigences.

Comprendre les dépendances

Comprendre les dépendances d'un système est essentiel à la résilience. Les dépendances incluent les connexions entre les composants d'une application et les connexions aux composants extérieurs à l'application, tels que les API tierces et les services partagés appartenant à l'entreprise. La compréhension de ces connexions vous permet d'isoler et de gérer les perturbations, car une défaillance d'un composant peut affecter d'autres composants. Ces connaissances aident les ingénieurs à évaluer l'impact des déficiences, à planifier en conséquence et à garantir une utilisation efficace des ressources. Comprendre les dépendances vous aide à créer des stratégies alternatives et à coordonner les processus de restauration. Il vous aide également à déterminer les cas dans lesquels vous pouvez remplacer une dépendance matérielle par une dépendance souple, afin que votre application puisse continuer à remplir ses fonctions commerciales en cas de déficience de dépendance. Les dépendances influencent également les décisions relatives à l'équilibrage de charge et au dimensionnement des applications. Il est essentiel de comprendre les dépendances lorsque vous apportez des modifications à votre application, car cela peut vous aider à déterminer les risques et les impacts potentiels. Ces connaissances vous aident à créer des applications stables et résilientes, en participant à la gestion des défaillances, à l'évaluation de l'impact, au rétablissement des défaillances, à l'équilibrage de charge, à la mise à l'échelle et à la gestion des modifications. Vous pouvez suivre les dépendances manuellement ou utiliser des outils et des services tels que AWS X-Raypour comprendre les dépendances de vos applications distribuées.

Stratégies de reprise après sinistre

Une stratégie de reprise après sinistre (DR) joue un rôle essentiel dans la création et l'exploitation d'applications résilientes, principalement en garantissant la continuité des activités. Il garantit que les opérations commerciales cruciales peuvent se poursuivre avec le moins d'impact possible, même

en cas de catastrophe, minimisant ainsi les temps d'arrêt et les pertes potentielles de revenus. Les stratégies de reprise après sinistre sont essentielles à la protection des données car elles intègrent souvent des sauvegardes et une réplication régulières des données sur plusieurs sites, ce qui permet de protéger les informations commerciales précieuses et d'éviter toute perte totale en cas de sinistre. En outre, de nombreux secteurs sont réglementés par des politiques qui obligent les entreprises à mettre en place une stratégie de reprise après sinistre afin de protéger les données sensibles et de garantir la disponibilité des services en cas de sinistre. En garantissant une diminution minimale du service, une stratégie de reprise après sinistre renforce également la confiance et la satisfaction des clients. Une stratégie de reprise après sinistre bien mise en œuvre et fréquemment mise en œuvre réduit le temps de reprise après un sinistre et permet de garantir que les applications sont rapidement remises en ligne. En outre, les catastrophes peuvent entraîner des coûts importants, non seulement en raison de la perte de revenus due aux interruptions de service, mais également en raison des dépenses liées à la restauration des applications et des données. Une stratégie de reprise après sinistre bien conçue permet de se prémunir contre ces pertes financières.

La stratégie que vous choisissez dépend des besoins spécifiques de votre application, de vos RTO et RPO, ainsi que de votre budget. AWS Elastic Disaster Recoveryest un service de résilience spécialement conçu que vous pouvez utiliser pour vous aider à mettre en œuvre votre stratégie de reprise après sinistre pour les applications sur site et dans le cloud.

Pour plus d'informations, consultez les sections Disaster Recovery of Workloads on AWS et AWS Multi-Region Fundamentals sur le AWS site Web.

Définition des stratégies CI/CD

L'une des causes les plus fréquentes d'altération des applications est le code ou d'autres modifications qui modifient l'état de fonctionnement de l'application par rapport à un état de fonctionnement connu auparavant. Si vous n'abordez pas la gestion du changement avec soin, cela peut entraîner de fréquentes déficiences. La fréquence des changements augmente les chances d'impact. Cependant, le fait d'apporter des modifications moins fréquemment entraîne des ensembles de modifications plus importants, qui sont beaucoup plus susceptibles d'entraîner des altérations en raison de leur grande complexité. Les pratiques d'intégration continue et de livraison continue (CI/CD) sont conçues pour que les modifications soient minimes et fréquentes (ce qui se traduit par une augmentation de la productivité) tout en soumettant chaque modification à un niveau élevé d'inspection par le biais de l'automatisation. Certaines des stratégies fondamentales sont les suivantes :

- Automatisation complète : le concept fondamental du CI/CD est d'automatiser au maximum les processus de création et de déploiement. Cela inclut la création, les tests, le déploiement et même la surveillance. Les pipelines automatisés contribuent à réduire les risques d'erreur humaine, à garantir la cohérence et à rendre le processus plus fiable et plus efficace.
- Développement piloté par les tests (TDD) : rédigez des tests avant d'écrire le code de l'application. Cette pratique garantit que tout le code est associé à des tests, ce qui améliore la fiabilité du code et la qualité de l'inspection automatisée. Ces tests sont exécutés dans le pipeline CI pour valider les modifications.
- Validations et intégrations fréquentes : encouragez les développeurs à valider du code fréquemment et à effectuer régulièrement des intégrations. Les modifications mineures et fréquentes sont plus faciles à tester et à déboguer, ce qui réduit le risque de problèmes importants. L'automatisation réduit le coût de chaque validation et de chaque déploiement, ce qui permet des intégrations fréquentes.
- Infrastructure immuable : traitez vos serveurs et les autres composants de l'infrastructure comme des entités statiques et immuables. Remplacez l'infrastructure au lieu de la modifier autant que possible, et créez une nouvelle infrastructure à l'aide d'un code testé et déployé dans votre pipeline.
- Mécanisme d'annulation : disposez toujours d'un moyen simple, fiable et fréquemment testé pour annuler les modifications en cas de problème. Pour garantir la sécurité du déploiement, il est essentiel de pouvoir revenir rapidement au bon état connu antérieur. Il peut s'agir d'un simple bouton pour revenir à l'état précédent, ou il peut être entièrement automatisé et déclenché par des alarmes.
- Contrôle de version : Conservez l'ensemble du code, de la configuration et même de l'infrastructure de l'application sous forme de code dans un référentiel contrôlé par version. Cette pratique vous permet de suivre facilement les modifications et de les annuler si nécessaire.
- Déploiements Canary et déploiements bleu/vert : le fait de déployer d'abord de nouvelles versions de votre application sur un sous-ensemble de votre infrastructure, ou de gérer deux environnements (bleu/vert), vous permet de vérifier le comportement d'un changement en production et de revenir rapidement en arrière si nécessaire.

Le CI/CD ne concerne pas seulement les outils, mais aussi la culture. Il est tout aussi important de créer une culture qui valorise l'automatisation, les tests et les leçons à tirer des échecs que de mettre en œuvre les bons outils et processus. Les annulations, si elles sont effectuées très rapidement avec un impact minimal, ne doivent pas être considérées comme un échec mais comme une expérience d'apprentissage.

Conduite des ORR

Un examen de l'état de préparation opérationnelle (ORR) permet d'identifier les lacunes opérationnelles et procédurales. Chez Amazon, nous avons créé des ORR pour transformer les enseignements tirés de décennies d'expérience dans le domaine de l'exploitation de services haut de gamme en questions ciblées accompagnées de conseils sur les meilleures pratiques. Un ORR capture les leçons apprises précédemment et oblige les nouvelles équipes à s'assurer qu'elles ont pris en compte ces leçons dans leurs candidatures. Les ORR peuvent fournir une liste des modes de défaillance ou des causes de défaillance qui peuvent être intégrés à l'activité de modélisation de la résilience décrite dans la section sur la modélisation de la résilience ci-dessous. Pour plus d'informations, consultez la section Operational Readiness Reviews (ORR) sur le site Web AWS Well-Architected Framework.

Comprendre les limites d'isolation des AWS pannes

AWS fournit de multiples limites d'isolation des pannes pour vous aider à atteindre vos objectifs de résilience. Vous pouvez utiliser ces limites pour tirer parti de l'étendue prévisible de la maîtrise des impacts qu'elles fournissent. Vous devez savoir comment les AWS services sont conçus à l'aide de ces limites afin de pouvoir faire des choix intentionnels concernant les dépendances que vous sélectionnez pour votre application. Pour comprendre comment utiliser les limites dans votre application, consultez la section AWS Fault Isolation Boundaries sur le AWS site Web.

Sélection des réponses

Un système peut répondre de nombreuses manières à une alarme. Certaines alarmes peuvent nécessiter une réponse de la part de l'équipe des opérations, tandis que d'autres peuvent déclencher des mécanismes d'autoréparation au sein de l'application. Vous pouvez décider de conserver les réponses qui pourraient être automatisées sous forme d'opérations manuelles afin de contrôler les coûts de l'automatisation ou de gérer les contraintes techniques. Le type de réponse à une alarme est susceptible d'être sélectionné en fonction du coût de mise en œuvre de la réponse, de la fréquence prévue de l'alarme, de la précision de l'alarme et des pertes commerciales potentielles liées à l'absence totale de réponse à l'alarme.

Par exemple, lorsqu'un processus serveur se bloque, le processus peut être redémarré par le système d'exploitation, ou un nouveau serveur peut être provisionné et l'ancien arrêté, ou un opérateur peut être invité à se connecter à distance au serveur et à le redémarrer. Ces réponses ont

le même résultat, à savoir le redémarrage du processus du serveur d'applications, mais leurs niveaux de coûts de mise en œuvre et de maintenance varient.

> **Note**
>
> Vous pouvez sélectionner plusieurs réponses afin d'adopter une approche de résilience approfondie. Par exemple, dans le scénario précédent, l'équipe chargée de l'application pourrait choisir d'implémenter les trois réponses avec un délai entre chacune d'elles. Si l'indicateur d'échec du processus du serveur est toujours en état d'alarme au bout de 30 secondes, l'équipe peut supposer que le système d'exploitation n'a pas réussi à redémarrer le serveur d'applications. Par conséquent, ils peuvent créer un groupe de dimensionnement automatique pour créer un nouveau serveur virtuel et restaurer le processus du serveur d'applications. Si l'indicateur est toujours en état d'alarme après 300 secondes, une alerte peut être envoyée au personnel opérationnel pour qu'il se connecte au serveur d'origine et tente de rétablir le processus.

La réponse choisie par l'équipe d'application et l'entreprise doit refléter la volonté de l'entreprise de compenser les frais opérationnels par un investissement initial en temps d'ingénierie. Vous devez choisir une réponse (un modèle d'architecture tel que la stabilité statique, un modèle logiciel tel qu'un disjoncteur ou une procédure opérationnelle) en prenant soigneusement en compte les contraintes et la maintenance prévue de chaque option de réponse. Certaines réponses standard peuvent exister pour guider les équipes chargées des applications. Vous pouvez donc utiliser les bibliothèques et les modèles gérés par votre fonction d'architecture centralisée comme contribution à cette prise en compte.

Modélisation de résilience

La modélisation de la résilience documente la manière dont une application réagira aux différentes perturbations prévues. En anticipant les perturbations, votre équipe peut mettre en œuvre des processus d'observabilité, des contrôles automatisés et des processus de reprise afin d'atténuer ou de prévenir les défaillances malgré les perturbations. AWS a créé des directives pour le développement d'un modèle de résilience en utilisant le cadre d'analyse de la résilience. Ce cadre peut vous aider à anticiper les perturbations et leur impact sur votre application. En anticipant les perturbations, vous pouvez identifier les mesures d'atténuation nécessaires pour créer une application résiliente et fiable. Nous vous recommandons d'utiliser le cadre d'analyse de résilience pour mettre à jour votre modèle de résilience à chaque itération du cycle de vie de votre application.

L'utilisation de ce framework à chaque itération permet de réduire les incidents en anticipant les interruptions pendant la phase de conception et en testant l'application avant et après le déploiement en production. Le développement d'un modèle de résilience à l'aide de ce cadre vous permet de vous assurer que vous atteignez vos objectifs de résilience.

Échouer en toute

Si vous ne parvenez pas à éviter les perturbations, échouez en toute sécurité. Envisagez de créer votre application avec un mode de fonctionnement à sécurité intégrée par défaut, dans lequel aucune perte commerciale significative ne peut être subie. Un exemple d'état de sécurité intégrée pour une base de données serait d'utiliser par défaut les opérations en lecture seule, dans lesquelles les utilisateurs ne sont pas autorisés à créer ou à muter des données. En fonction de la sensibilité des données, vous souhaiterez peut-être même que l'application passe par défaut à l'état d'arrêt sans même effectuer de requêtes en lecture seule. Déterminez quel devrait être l'état de sécurité intégré de votre application et optez par défaut pour ce mode de fonctionnement dans des conditions extrêmes.

Étape 3 : Évaluer et tester

Au cours de la phase d'évaluation et de test du cycle de vie, l'application, ou les modifications apportées à une application existante, ont été conçues mais n'ont pas encore été mises en production. Au cours de cette étape, vous mettez en œuvre des activités visant à tester les pratiques mises en œuvre lors des étapes précédentes et à évaluer les résultats. L'application est peut-être toujours en cours de développement, ou le développement principal est peut-être terminé et l'application est peut-être en cours de test avant sa mise en production. Au cours de cette étape, vous vous concentrez sur le développement et l'exécution de tests qui confirment ou réfutent les attentes selon lesquelles l'application atteindra les objectifs de résilience définis. De plus, vous développez et testez les procédures opérationnelles du système. Les procédures de déploiement que vous avez développées à l'étape 2 : Conception et mise en œuvre sont mises en pratique et les résultats sont évalués. Bien que ces activités de test et d'évaluation commencent au cours de cette partie du cycle de vie, elles ne s'arrêtent pas là. Les tests et les évaluations se poursuivent au fur et à mesure que vous passez à l'étape 4 : exploitation.

La phase d'évaluation et de test est divisée en deux phases : les activités préalables au déploiement et les activités post-déploiement. Les activités de pré-déploiement consistent en des tâches qui doivent être effectuées avant de déployer l'application dans n'importe quel environnement, y compris le déploiement de nouvelles versions du logiciel ainsi que le déploiement initial dans un environnement de test. Les activités de post-déploiement ont lieu après le déploiement du logiciel dans un environnement de test ou de production. Les sections suivantes traitent de ces phases plus en détail.

Activités préalables au déploiement

Conception de l'environnement

L'environnement dans lequel vous testez et évaluez votre application influe sur la précision avec laquelle vous pouvez la tester et sur votre degré de confiance dans le fait que ces résultats reflètent fidèlement ce qui se passera en production. Vous pouvez peut-être effectuer des tests d'intégration localement sur les machines des développeurs en utilisant des services tels qu'Amazon DynamoDB (voir Configuration de DynamoDB en local dans la documentation DynamoDB). Cependant, à un moment donné, vous devez effectuer des tests dans un environnement qui reproduit votre environnement de production afin d'obtenir des résultats aussi fiables que possible. Cet environnement étant coûteux, nous vous recommandons d'adopter une approche progressive, ou en

pipeline, dans le cadre de laquelle les environnements de type production apparaîtront plus tard dans le pipeline.

Tests d'intégration

Les tests d'intégration sont le processus qui consiste à vérifier qu'un composant bien défini d'une application exécute correctement ses fonctions lorsqu'il fonctionne avec des dépendances externes. Ces dépendances externes peuvent être d'autres composants développés sur mesure, AWS des services que vous utilisez pour votre application, des dépendances tierces et des dépendances sur site. Ce guide se concentre sur les tests d'intégration qui démontrent la résilience de votre application. Cela suppose qu'il existe déjà des tests unitaires et d'intégration qui démontrent la précision fonctionnelle de votre logiciel.

Nous vous recommandons de concevoir des tests d'intégration qui testent spécifiquement les modèles de résilience que vous avez mis en œuvre, tels que les modèles de disjoncteurs ou le délestage (voir Étape 2 : Conception et mise en œuvre). Les tests d'intégration axés sur la résilience impliquent souvent d'appliquer une charge spécifique à l'application ou d'introduire intentionnellement des perturbations dans l'environnement en utilisant des fonctionnalités telles que AWS Fault Injection Service ().AWS FIS Idéalement, vous devez exécuter tous les tests d'intégration dans le cadre de votre pipeline CI/CD et vous assurer d'exécuter des tests chaque fois que le code est validé. Cela vous permet de détecter et de réagir rapidement à toute modification du code ou des configurations qui entraîne des violations de vos objectifs de résilience. Les applications distribuées à grande échelle sont complexes, et même des modifications mineures peuvent avoir un impact significatif sur la résilience de parties apparemment indépendantes de votre application. Essayez d'exécuter vos tests à chaque validation. AWS fournit un excellent ensemble d'outils pour faire fonctionner votre pipeline CI/CD et d'autres DevOps outils. Pour plus d'informations, consultez la section Introduction à DevOps on AWS sur le AWS site Web.

Pipelines de déploiement automatisés

Le déploiement et les tests dans vos environnements de pré-production sont des tâches répétitives et complexes qu'il est préférable de laisser à l'automatisation. L'automatisation de ce processus permet de libérer des ressources humaines et de réduire les risques d'erreur. Le mécanisme d'automatisation de ce processus est souvent appelé pipeline. Lorsque vous créez votre pipeline, nous vous recommandons de configurer une série d'environnements de test qui se rapprochent de plus en plus de votre configuration de production. Vous utilisez cette série d'environnements pour tester votre application à plusieurs reprises. Le premier environnement fournit un ensemble de fonctionnalités plus limité que l'environnement de production, mais son coût est nettement

inférieur. Les environnements suivants devraient ajouter des services et évoluer pour mieux refléter l'environnement de production.

Commencez par tester dans le premier environnement. Une fois que vos déploiements ont réussi tous vos tests dans le premier environnement de test, laissez l'application s'exécuter sous une certaine charge pendant un certain temps pour voir si des problèmes surviennent au fil du temps. Vérifiez que vous avez correctement configuré l'observabilité (voir Précision des alarmes plus loin dans ce guide) afin de pouvoir détecter tout problème éventuel. Lorsque cette période d'observation s'est terminée avec succès, déployez votre application dans votre environnement de test suivant et répétez le processus en ajoutant des tests ou une charge supplémentaires selon les besoins de l'environnement. Après avoir suffisamment testé votre application de cette manière, vous pouvez utiliser les méthodes de déploiement que vous avez précédemment configurées pour déployer l'application en production (voir Définir des stratégies CI/CD plus haut dans ce guide). L'article Automating safe and handoff deployments in the Amazon Builders' Library est une excellente ressource qui décrit comment Amazon automatise le déploiement du code. Le nombre d'environnements qui précèdent votre déploiement en production varie en fonction de la complexité de votre application et des types de dépendances qu'elle comporte.

Test de charge

À première vue, les tests de charge ressemblent à des tests d'intégration. Vous testez une fonction discrète de votre application et ses dépendances externes pour vérifier qu'elle fonctionne comme prévu. Les tests de charge vont ensuite au-delà des tests d'intégration pour se concentrer sur le fonctionnement de l'application sous des charges bien définies. Les tests de charge nécessitent la vérification des fonctionnalités correctes. Ils doivent donc être effectués après un test d'intégration réussi. Il est important de comprendre dans quelle mesure l'application répond aux charges attendues et comment elle se comporte lorsque la charge dépasse les attentes. Cela vous permet de vérifier que vous avez mis en œuvre les mécanismes nécessaires pour garantir la résilience de votre application en cas de charge extrême. Pour un guide complet sur les tests de charge AWS, consultez la section Test de charge distribué sur AWS la bibliothèque de AWS solutions.

Activités postérieures au déploiement

La résilience est un processus continu et l'évaluation de la résilience de votre application doit se poursuivre après le déploiement de l'application. Les résultats de vos activités post-déploiement, telles que les évaluations continues de la résilience, peuvent nécessiter que vous réévaluiez et mettiez à jour certaines des activités de résilience que vous avez effectuées plus tôt dans le cycle de vie de résilience.

Réalisation d'évaluations de résilience

L'évaluation de la résilience ne s'arrête pas une fois que vous avez déployé votre application en production. Même si vous disposez de pipelines de déploiement bien définis et automatisés, les modifications peuvent parfois se produire directement dans un environnement de production. En outre, il se peut que vous n'ayez pas encore pris en compte certains facteurs lors de votre vérification de résilience avant le déploiement. AWS Resilience Hubfournit un emplacement central où vous pouvez évaluer si votre architecture déployée répond à vos besoins définis en matière de RPO et de RTO. Vous pouvez utiliser ce service pour effectuer des évaluations à la demande de la résilience de votre application, automatiser les évaluations et même les intégrer dans vos outils CI/CD, comme indiqué dans le billet de AWS blog Évaluation continue de la résilience des applications avec AWS Resilience Hub et. AWS CodePipeline L'automatisation de ces évaluations est une bonne pratique car elle permet de garantir que vous évaluez en permanence votre posture de résilience en production.

tests de reprise après sinistre

Au cours de l'étape 2 : Conception et mise en œuvre, vous avez développé des stratégies de reprise après sinistre (DR) intégrées à votre système. Au cours de l'étape 4, vous devez tester vos procédures de reprise après sinistre pour vous assurer que votre équipe est parfaitement préparée à un incident et que vos procédures fonctionnent comme prévu. Vous devez tester régulièrement toutes vos procédures de reprise après sinistre, y compris le basculement et le retour en arrière, et examiner les résultats de chaque exercice pour déterminer si et comment les procédures de votre système doivent être mises à jour pour obtenir les meilleurs résultats possibles. Lorsque vous développez initialement votre test DR, planifiez le test bien à l'avance et assurez-vous que toute l'équipe comprend à quoi s'attendre, comment les résultats seront mesurés et quel mécanisme de feedback sera utilisé pour mettre à jour les procédures en fonction des résultats. Une fois que vous aurez acquis les compétences nécessaires pour exécuter des tests de reprise après sinistre planifiés, pensez à exécuter des tests de reprise après sinistre inopinés. Les véritables catastrophes ne se produisent pas selon un calendrier, vous devez donc être prêt à mettre en œuvre votre plan à tout moment. Cependant, inopiné ne signifie pas imprévu. Les principales parties prenantes doivent encore planifier l'événement pour s'assurer qu'une surveillance appropriée est en place et que les clients et les applications critiques ne sont pas affectés négativement.

Détection des écarts

Des modifications imprévues de configuration dans les applications de production peuvent se produire même lorsque l'automatisation et des procédures bien définies sont en place. Pour

détecter les modifications apportées à la configuration de votre application, vous devez disposer de mécanismes permettant de détecter les dérives, c'est-à-dire les écarts par rapport à une configuration de référence. Pour savoir comment détecter la dérive dans vos AWS CloudFormation piles, consultez la section Détection des modifications de configuration non gérées apportées aux piles et aux ressources dans la documentation. AWS CloudFormation Pour détecter la dérive dans l' AWS environnement de votre application, consultez la section Détecter et résoudre la dérive AWS Control Tower dans la AWS Control Tower documentation.

Tests synthétiques

Les tests synthétiques sont le processus de création d'un logiciel configurable qui s'exécute en production, sur une base planifiée, afin de tester les API de votre application de manière à simuler l'expérience de l'utilisateur final. Ces tests sont parfois appelés canaris, en référence à l'utilisation initiale du terme dans les mines de charbon. Les tests synthétiques peuvent souvent fournir des alertes précoces lorsqu'une application subit une interruption, même si celle-ci est partielle ou intermittente, comme c'est souvent le cas pour les défaillances grises.

Ingénierie du chaos

L'ingénierie du chaos est un processus systématique qui consiste à soumettre délibérément une application à des événements perturbateurs de manière à atténuer les risques, à surveiller de près sa réponse et à mettre en œuvre les améliorations nécessaires. Son objectif est de valider ou de remettre en question les hypothèses concernant la capacité de l'application à gérer de telles perturbations. Au lieu de laisser ces événements au hasard, l'ingénierie du chaos permet aux ingénieurs d'orchestrer des expériences dans un environnement contrôlé, généralement pendant les périodes de faible trafic, avec un support technique facilement disponible pour une atténuation efficace.

L'ingénierie du chaos commence par la compréhension des conditions de fonctionnement normales, appelées régime permanent, de l'application considérée. À partir de là, vous formulez une hypothèse qui détaille le comportement efficace de l'application en cas de perturbation. Vous exécutez l'expérience, qui implique l'injection délibérée de perturbations, y compris, mais sans s'y limiter, la latence du réseau, les pannes de serveur, les erreurs de disque dur et l'altération des dépendances externes. Vous analysez ensuite les résultats de l'expérience et améliorez la résilience de l'application en fonction de vos apprentissages. L'expérience constitue un outil précieux pour améliorer divers aspects de l'application, notamment ses performances, et permet de découvrir des problèmes latents qui auraient pu rester cachés autrement. En outre, l'ingénierie du chaos permet de révéler les lacunes des outils d'observabilité et d'alarme, et vous aide à les affiner.

Cela contribue également à réduire le temps de récupération et à améliorer les compétences opérationnelles. L'ingénierie du chaos accélère l'adoption des meilleures pratiques et cultive un état d'esprit d'amélioration continue. En fin de compte, cela permet aux équipes de développer et de perfectionner leurs compétences opérationnelles grâce à des exercices réguliers et à des répétitions.

AWS vous recommande de commencer vos efforts d'ingénierie du chaos dans un environnement hors production. Vous pouvez utiliser AWS Fault Injection Service (AWS FIS) pour exécuter des expériences d'ingénierie du chaos avec des défauts à usage général ainsi que des défauts spécifiques à. AWS Ce service entièrement géré inclut des alarmes d'arrêt et des contrôles complets des autorisations afin que vous puissiez facilement adopter l'ingénierie du chaos en toute sécurité et en toute confiance.

Étape 4 : opérer

Après avoir terminé l'étape 3 : évaluation et test, vous êtes prêt à déployer l'application en production. Au stade Operate, vous déployez votre application en production et gérez l'expérience de vos clients. La conception et la mise en œuvre de votre application déterminent bon nombre de ses résultats en matière de résilience, mais cette étape se concentre sur les pratiques opérationnelles utilisées par votre système pour maintenir et améliorer la résilience. La mise en place d'une culture d'excellence opérationnelle contribue à créer des normes et à uniformiser ces pratiques.

Observabilité

Pour comprendre l'expérience client, il est essentiel de recourir à la surveillance et à l'alarme. Vous devez instrumenter votre application pour comprendre son état, et vous avez besoin de points de vue variés, ce qui signifie que vous devez mesurer à la fois du côté serveur et du côté client, généralement à l'aide de canaris. Vos métriques doivent inclure des données sur les interactions de votre application avec ses dépendances et des dimensions conformes à vos limites d'isolation des pannes. Vous devez également produire des journaux fournissant des détails supplémentaires sur chaque unité de travail effectuée par votre application. Vous pouvez envisager de combiner les métriques et les journaux en utilisant une solution telle que le format de métrique CloudWatch intégré Amazon. Vous constaterez probablement que vous souhaitez toujours plus d'observabilité, alors considérez les compromis en termes de coûts, d'efforts et de complexité nécessaires pour mettre en œuvre le niveau d'instrumentation souhaité.

Les liens suivants fournissent les meilleures pratiques pour instrumenter votre application et créer des alarmes :

- Surveillance des services de production chez Amazon (présentation AWS re:Invent 2020)
- Amazon Builders' Library : l'excellence opérationnelle chez Amazon (présentation re:Invent 2021)AWS
- Bonnes pratiques en matière d'observabilité chez Amazon (présentation AWS re:Invent 2022)
- Instrumentation des systèmes distribués pour une visibilité opérationnelle (article Amazon Builders' Library)
- Création de tableaux de bord pour une visibilité opérationnelle (article Amazon Builders' Library)

Gestion d'événements

Vous devriez mettre en place un processus de gestion des événements pour gérer les défaillances lorsque vos alarmes (ou pire encore, vos clients) vous indiquent que quelque chose ne va pas. Ce processus doit inclure l'engagement d'un opérateur de garde, l'escalade des problèmes et l'établissement de guides pour des approches cohérentes de dépannage qui aident à éliminer les erreurs humaines. Cependant, les déficiences ne se produisent généralement pas de manière isolée ; une seule application peut avoir un impact sur plusieurs autres applications qui en dépendent. Vous pouvez résoudre rapidement les problèmes en comprenant toutes les applications concernées et en réunissant les opérateurs de plusieurs équipes lors d'une seule conférence téléphonique. Toutefois, en fonction de la taille et de la structure de votre organisation, ce processus peut nécessiter une équipe opérationnelle centralisée.

Outre la mise en place d'un processus de gestion des événements, vous devez régulièrement revoir vos indicateurs par le biais de tableaux de bord. Des évaluations régulières vous aident à comprendre l'expérience client et les tendances à long terme en matière de performances de votre application. Cela vous permet d'identifier les problèmes et les goulots d'étranglement avant qu'ils n'aient un impact significatif sur la production. L'examen des indicateurs de manière cohérente et standardisée présente des avantages importants, mais nécessite une adhésion du haut vers le bas et un investissement en temps.

Les liens suivants fournissent les meilleures pratiques en matière de création de tableaux de bord et de révisions des indicateurs opérationnels :

- Création de tableaux de bord pour une visibilité opérationnelle (article Amazon Builders' Library)
- L'approche d'Amazon pour réussir en cas d'échec (présentation AWS re:Invent 2019)

Résilience continue

Au cours de l'étape 2 : conception et mise en œuvre et de l'étape 3 : évaluation et test, vous avez lancé des activités de révision et de test avant de déployer votre application en production. Pendant la phase d'exploitation, vous devez continuer à répéter ces activités en production. Vous devez régulièrement revoir la posture de résilience de votre application par le biais des révisions du AWS Well-Architected Framework, des évaluations de l'état de préparation opérationnelle (ORR) et du cadre d'analyse de la résilience. Cela permet de s'assurer que votre application ne s'écarte pas des bases et des normes établies et de vous tenir au courant des nouvelles directives ou des mises

à jour. Ces activités de résilience continues vous aident à découvrir des perturbations imprévues auparavant et à proposer de nouvelles mesures d'atténuation.

Vous pouvez également envisager de lancer des journées de jeu et des expériences d'ingénierie du chaos en production une fois que vous les aurez exécutées avec succès dans des environnements de pré-production. Les journées de jeu simulent des événements connus que vous avez mis en place des mécanismes de résilience pour atténuer. Par exemple, une journée de jeu peut simuler une défaillance du service AWS régional et mettre en œuvre un basculement multirégional. Bien que la mise en œuvre de ces activités puisse nécessiter des efforts considérables, les deux pratiques vous aident à vous assurer que votre système est résilient aux modes de défaillance pour lesquels vous l'avez conçu.

En exploitant vos applications, en rencontrant des événements opérationnels, en examinant les métriques et en testant votre application, vous aurez de nombreuses occasions de réagir et d'apprendre.

Étape 5 : Réagir et apprendre

La façon dont votre application réagit aux événements perturbateurs influence sa fiabilité. Il est également essentiel de tirer les leçons de l'expérience et de la manière dont votre application a réagi aux perturbations dans le passé pour améliorer sa fiabilité.

La phase de réponse et d'apprentissage se concentre sur les pratiques que vous pouvez mettre en œuvre pour mieux répondre aux événements perturbateurs dans vos applications. Il inclut également des pratiques qui vous aideront à tirer le meilleur parti des expériences de vos équipes opérationnelles et de vos ingénieurs.

Création de rapports d'analyse des incidents

Lorsqu'un incident survient, la première mesure à prendre est de prévenir le plus rapidement possible de nouveaux dommages aux clients et à l'entreprise. Une fois l'application rétablie, l'étape suivante consiste à comprendre ce qui s'est passé et à identifier les mesures à prendre pour éviter que cela ne se reproduise. Cette analyse post-incident est généralement capturée sous la forme d'un rapport qui documente l'ensemble des événements qui ont entraîné une détérioration de l'application, ainsi que les effets de l'interruption sur l'application, les clients et l'entreprise. Ces rapports deviennent de précieux outils d'apprentissage et devraient être largement diffusés au sein de l'entreprise.

> **Note**
>
> Il est essentiel d'effectuer une analyse des incidents sans attribuer de responsabilité. Supposons que tous les opérateurs aient pris les mesures les meilleures et les plus appropriées compte tenu des informations dont ils disposaient. N'utilisez pas les noms des opérateurs ou des ingénieurs dans un rapport. Invoquer une erreur humaine comme cause de déficience peut inciter les membres de l'équipe à être surveillés afin de se protéger, ce qui se traduirait par la saisie d'informations incorrectes ou incomplètes.

Un bon rapport d'analyse des incidents, tel que celui documenté dans le processus Amazon Correction of Error (COE), suit un format standardisé et tente de saisir, de manière aussi détaillée que possible, les conditions qui ont entraîné une détérioration de l'application. Le rapport détaille une série d'événements horodatés et capture des données quantitatives (souvent des métriques et des captures d'écran provenant de tableaux de bord de surveillance) qui décrivent l'état mesurable de

l'application au cours de la chronologie. Le rapport doit rendre compte des processus de réflexion des opérateurs et des ingénieurs qui ont pris des mesures, ainsi que des informations qui les ont conduits à leurs conclusions. Le rapport doit également détailler les performances des différents indicateurs, par exemple, quelles alarmes ont été déclenchées, si ces alarmes reflétaient correctement l'état de l'application, le délai entre les événements et les alarmes qui en ont résulté, et le temps nécessaire pour résoudre l'incident. La chronologie décrit également les runbooks ou les automatisations qui ont été initiés et comment ils ont aidé l'application à retrouver un état utile. Ces éléments du calendrier aident votre équipe à comprendre l'efficacité des réponses automatisées et des réponses des opérateurs, notamment la rapidité avec laquelle ils ont résolu le problème et leur efficacité à atténuer les perturbations.

Ce portrait détaillé d'un événement historique est un puissant outil pédagogique. Les équipes doivent stocker ces rapports dans un référentiel central accessible à l'ensemble de l'entreprise afin que d'autres puissent examiner les événements et en tirer des leçons. Cela peut améliorer l'intuition de vos équipes quant à ce qui peut mal tourner en production.

Un référentiel de rapports d'incidents détaillés devient également une source de matériel de formation pour les opérateurs. Les équipes peuvent utiliser un rapport d'incident pour organiser une journée de jeu sur table ou en direct, au cours de laquelle les équipes reçoivent des informations reprenant la chronologie enregistrée dans le rapport. Les opérateurs peuvent parcourir le scénario à l'aide d'informations partielles provenant de la chronologie et décrire les actions qu'ils entreprendraient. Le modérateur du jour du match peut ensuite fournir des conseils sur la façon dont l'application a réagi en fonction des actions de l'opérateur. Cela développe les compétences des opérateurs en matière de dépannage, afin qu'ils puissent anticiper et résoudre les problèmes plus facilement.

Une équipe centralisée chargée de la fiabilité des applications doit conserver ces rapports dans une bibliothèque centralisée accessible à l'ensemble de l'entreprise. Cette équipe devrait également être chargée de maintenir le modèle de rapport et de former les équipes sur la manière de rédiger le rapport d'analyse des incidents. L'équipe de fiabilité doit examiner régulièrement les rapports afin de détecter les tendances au sein de l'entreprise qui peuvent être prises en compte par le biais de bibliothèques logicielles, de modèles d'architecture ou de modifications des processus d'équipe.

Réalisation d'examens opérationnels

Comme indiqué dans la section Étape 4 : Exploitation, les évaluations opérationnelles sont l'occasion de passer en revue les dernières mises à jour des fonctionnalités, les incidents et les indicateurs opérationnels. La revue opérationnelle est également l'occasion de partager les enseignements tirés des mises à jour des fonctionnalités et des incidents avec l'ensemble de la communauté

des ingénieurs de votre organisation. Au cours de l'examen opérationnel, les équipes examinent les déploiements de fonctionnalités annulés, les incidents survenus et la manière dont ils ont été gérés. Cela donne aux ingénieurs de l'ensemble de l'organisation l'occasion de tirer des leçons de l'expérience des autres et de poser des questions.

Proposez vos évaluations opérationnelles à la communauté des ingénieurs de votre entreprise afin qu'elle puisse en savoir plus sur les applications informatiques qui gèrent l'entreprise et les types de problèmes qu'elle peut rencontrer. Ils emporteront ces connaissances avec eux lors de la conception, de la mise en œuvre et du déploiement d'autres applications pour l'entreprise.

Examen des performances des alarmes

Les alarmes, comme indiqué lors de la phase d'exploitation, peuvent entraîner des alertes sur le tableau de bord, la création de tickets, l'envoi d'e-mails ou l'envoi d'une pagination des opérateurs. Une application comportera de nombreuses alarmes configurées pour surveiller divers aspects de son fonctionnement. Au fil du temps, la précision et l'efficacité de ces alarmes doivent être revues afin d'accroître la précision des alarmes, de réduire le nombre de faux positifs et de consolider les alertes dupliquées.

Précision de l'alarme

Les alarmes doivent être aussi spécifiques que possible afin de réduire le temps que vous devez consacrer à l'interprétation ou au diagnostic de la perturbation spécifique à l'origine de l'alarme. Lorsqu'une alarme est déclenchée en réponse à une défaillance de l'application, les opérateurs qui reçoivent l'alarme et y répondent doivent d'abord interpréter les informations transmises par l'alarme. Les informations peuvent être un simple code d'erreur correspondant à un plan d'action tel qu'une procédure de restauration, ou elles peuvent inclure des lignes provenant des journaux d'applications que vous devez consulter pour comprendre pourquoi l'alarme a été déclenchée. Au fur et à mesure que votre équipe apprend à utiliser une application plus efficacement, elle doit affiner ces alarmes pour les rendre aussi claires et concises que possible.

Comme il est impossible d'anticiper toutes les perturbations possibles d'une application, il y aura toujours des alarmes générales qui nécessiteront l'analyse et le diagnostic de l'opérateur. Votre équipe doit s'efforcer de réduire le nombre d'alarmes générales afin d'améliorer les temps de réponse et de réduire le temps moyen de réparation (MTTR). Idéalement, il devrait y avoir une one-to-one relation entre une alarme et une réponse automatisée ou exécutée par l'homme.

Faux positifs

Les alarmes qui ne nécessitent aucune action de la part des opérateurs mais qui produisent des alertes sous forme d'e-mails, de pages ou de tickets seront ignorées par les opérateurs au fil du temps. Régulièrement, ou dans le cadre d'une analyse des incidents, passez en revue les alarmes pour identifier celles qui sont souvent ignorées ou qui ne nécessitent aucune action de la part des opérateurs (faux positifs). Vous devez vous efforcer de supprimer l'alarme ou de l'améliorer afin qu'elle émette une alerte exploitable aux opérateurs.

Faux négatifs

Lors d'un incident, les alarmes configurées pour alerter pendant l'incident peuvent échouer, peut-être en raison d'un événement qui a un impact inattendu sur l'application. Dans le cadre de l'analyse d'un incident, vous devez passer en revue les alarmes qui auraient dû être déclenchées mais qui ne l'ont pas été. Vous devez vous efforcer d'améliorer ces alarmes afin qu'elles reflètent mieux les conditions susceptibles de découler d'un événement. Il se peut également que vous deviez créer des alarmes supplémentaires correspondant à la même interruption mais déclenchées par un symptôme différent de la perturbation.

Alertes dupliquées

Une interruption qui altère votre application est susceptible de provoquer de multiples symptômes et d'entraîner plusieurs alarmes. Régulièrement, ou dans le cadre d'une analyse d'incident, vous devez passer en revue les alarmes et alertes émises. Si les opérateurs ont reçu des alertes dupliquées, créez des alarmes agrégées pour les regrouper en un seul message d'alerte.

Réalisation d'examens des métriques

Votre équipe doit collecter des indicateurs opérationnels concernant votre application, tels que le nombre d'incidents par niveau de gravité par mois, le temps nécessaire pour détecter l'incident, le temps nécessaire pour identifier la cause, le temps nécessaire pour y remédier, ainsi que le nombre de tickets créés, d'alertes envoyées et de pages créées. Passez en revue ces indicateurs au moins une fois par mois pour comprendre la charge de travail qui pèse sur le personnel opérationnel, le signal-to-noise ratio auquel il est confronté (par exemple, alertes informatives par rapport aux alertes exploitables) et pour savoir si l'équipe améliore sa capacité à exploiter les applications sous son contrôle. Utilisez cette revue pour comprendre les tendances relatives aux aspects mesurables de l'équipe des opérations. Sollicitez des idées auprès de l'équipe sur la manière d'améliorer ces indicateurs.

Fournir des formations et des habilitations

Il est difficile de saisir une description détaillée d'une application et de son environnement à l'origine d'un incident ou d'un comportement inattendu. En outre, modéliser la résilience de votre application pour anticiper de tels scénarios n'est pas toujours simple. Votre organisation doit investir dans des supports de formation et de facilitation permettant à vos équipes opérationnelles et à vos développeurs de participer à des activités telles que la modélisation de la résilience, l'analyse des incidents, les journées de jeu et les expériences d'ingénierie du chaos. Cela améliorera la fidélité des rapports produits par vos équipes et les connaissances qu'elles capturent. Les équipes seront également mieux équipées pour anticiper les défaillances sans compter sur un groupe d'ingénieurs plus restreint et plus expérimenté qui doivent apporter leur point de vue par le biais de révisions planifiées.

Création d'une base de connaissances sur les incidents

Un rapport d'incident est un résultat standard d'une analyse d'incident. Vous devez utiliser le même rapport ou un rapport similaire pour documenter les scénarios dans lesquels vous avez détecté un comportement anormal de l'application, même si l'application n'a pas été altérée. Utilisez la même structure de rapport standardisée pour saisir les résultats des expériences chaotiques et des journées de jeu. Le rapport représente un instantané de l'application et de son environnement à l'origine d'un incident ou d'un comportement inattendu. Vous devez stocker ces rapports standardisés dans un référentiel central accessible à tous les ingénieurs de l'entreprise.

Les équipes opérationnelles et les développeurs peuvent ensuite effectuer des recherches dans cette base de connaissances pour comprendre ce qui a perturbé les applications dans le passé, quels types de scénarios auraient pu provoquer des perturbations et ce qui a permis d'éviter toute détérioration des applications. Cette base de connaissances devient un accélérateur pour améliorer les compétences de vos équipes opérationnelles et de vos développeurs, et leur permet de partager leurs connaissances et leurs expériences. En outre, vous pouvez utiliser les rapports comme matériel de formation ou comme scénarios pour les journées de jeu ou les expériences de chaos afin d'améliorer l'intuition de l'équipe opérationnelle et sa capacité à résoudre les problèmes liés aux perturbations.

> **Note**
>
> Un format de rapport normalisé donne également aux lecteurs un sentiment de familiarité et les aide à trouver plus rapidement les informations qu'ils recherchent.

Mettre en œuvre la résilience en profondeur

Comme indiqué précédemment, une organisation avancée mettra en œuvre plusieurs réponses à une alarme. Il n'y a aucune garantie qu'une réponse sera efficace. En superposant les réponses, une application sera mieux équipée pour échouer correctement. Nous vous recommandons de mettre en œuvre au moins deux réponses pour chaque indicateur afin de garantir qu'une réponse individuelle ne devienne pas un point de défaillance unique susceptible de mener à un scénario de reprise après sinistre. Ces couches doivent être créées par ordre séquentiel, de sorte qu'une réponse successive ne soit effectuée que si la réponse précédente s'est révélée inefficace. Vous ne devez pas exécuter de réponses à plusieurs niveaux à une seule alarme. Utilisez plutôt une alarme qui indique si une réponse a échoué et, dans l'affirmative, déclenche la réponse en couches suivante.

Conclusion et ressources

Ce guide présente un cycle de vie qui vous aide à améliorer en permanence la résilience de vos applications en mettant en œuvre les meilleures pratiques en cinq étapes : définition des objectifs, conception et mise en œuvre, évaluation et test, exploitation, réponse et apprentissage.

Pour plus d'informations sur les services et les concepts abordés dans ce guide, consultez les ressources suivantes.

AWS services :

- AWS Backup
- AWS Elastic Disaster Recovery
- AWS Fault Injection Service (AWS FIS)
- AWS Resilience Hub
- Contrôleur Amazon Application Recovery (ARC)
- AWS X-Ray

Articles de blog et articles :

- Disponibilité et au-delà : comprendre et améliorer la résilience des systèmes distribués sur AWS
- AWS Limites d'isolation des défauts
- AWS Principes fondamentaux de plusieurs régions
- L'ingénierie du chaos dans le cloud
- Évaluation continue de la résilience des applications avec AWS Resilience Hub et AWS CodePipeline
- Reprise après sinistre des applications sur site pour AWS
- Pilier de fiabilité — AWS Well-Architected Framework
- Cadre d'analyse de résilience

Collaborateurs

Les contributeurs à ce guide incluent :

- Bruno Emer, architecte principal des solutions, AWS
- Clark Richey, architecte de solutions principal, AWS
- Elaine Harvey, directrice générale des services de fiabilité, AWS
- Jason Barto, architecte de solutions principal, AWS
- John Formento, architecte de solutions principal, AWS
- Lisi Lewis, directrice principale du marketing des produits, AWS
- Michael Haken, architecte de solutions principal, AWS
- Neeraj Kumar, architecte de solutions principal, AWS
- Wangechi Doble, architecte de solutions principal, 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
Publication initiale	—	6 octobre 2023

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 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 (RDSAmazon) pour Oracle dans le. AWS Cloud
- Racheter (rachat) : optez pour un autre produit, généralement en passant d'une licence traditionnelle à un modèle SaaS. Exemple : migrez votre système de gestion de la relation client (CRM) vers Salesforce.com.
- Réhéberger (lift and shift) : transférez une application vers le cloud sans apporter de modifications pour tirer parti des fonctionnalités du cloud. Exemple : migrez votre base de données Oracle locale vers Oracle sur une EC2 instance du AWS Cloud.
- Relocaliser (lift and shift au niveau de l'hyperviseur) : transférez l'infrastructure vers le cloud sans acheter de nouveau matériel, réécrire des applications ou modifier vos opérations existantes. Vous migrez des serveurs d'une plateforme sur site vers un service cloud pour la même plateforme. Exemple : migrer un Microsoft Hyper-V application à 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.

ACID

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 la synchronisation des bases de données source et cible est maintenue, mais seule la base de données source gère les transactions provenant de la connexion d'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

SQLFonction 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, consistance, isolation, 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 basé sur les 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 AWS la documentation AWS Identity and Access Management (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 CAForganise les directives 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, 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, consultez le AWS CAFsite Web et le AWS CAFlivre blanc.

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 WQFest 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 API appels 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'indexation 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.

CCoE

Voir le Centre d'excellence du cloud.

CDC

Voir capture des données de modification.

modifier la capture de données (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 l'utiliser à diverses CDC 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 CCoEarticles 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 uneCCoE, établir un modèle d'exploitation)
- 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 or Bitbucket 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, AWS Panorama propose des appareils qui ajoutent des CV aux réseaux de caméras locaux, et Amazon SageMaker 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 de configuration (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 issues de la phase CMDB de découverte et d'analyse du portefeuille lors 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 YAML modèle. 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 source, de génération, de test, intermédiaire et de production du processus de publication du logiciel. CI/CD is commonly described as a pipeline. CI/CDpeut 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. DVSMétend 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 la façon dont vous pouvez utiliser le design piloté par domaine avec le motif Strangler Fig, voir Modernisation de l'ancienne version de Microsoft. ASP NET(ASMX) des services Web de manière incrémentielle à l'aide de conteneurs et d'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.

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.

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 dans 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 AWS Identity and Access Management (IAM) principaux. Ces comptes ou principaux peuvent se connecter à votre service de point de terminaison de manière privée en créant des points de VPC terminaison d'interface. Pour plus d'informations, consultez la section Créer un service de point de terminaison dans la documentation Amazon Virtual Private Cloud (AmazonVPC).

planification des ressources d'entreprise (ERP)

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

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 pipeline CI/CD, 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 grands enjeux en matière de AWS CAF 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. EDAest 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. Ceci est généralement exprimé sous la forme d'un score numérique qui peut être calculé à l'aide de

diverses techniques, telles que les explications additives de Shapley (SHAP) et les dégradés 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.

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.

G

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.

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. Ils sont mis en œuvre à l'aide de politiques de contrôle des services et de limites IAM d'autorisations. 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 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.

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 IAM principaux qui définit leurs autorisations au sein de l' AWS Cloud environnement.

application inactive

Application dont l'utilisation moyenne CPU 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.

IIoT

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.

entrant (entrée) VPC

Dans une architecture AWS multi-comptes, une architecture 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

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 Élaborer une stratégie de transformation numérique industrielle pour l'Internet des objets (IIoT).

inspection VPC

Dans une architecture AWS multi-comptes, système centralisé VPC qui gère les inspections du trafic réseau entre VPCs (identiques ou différents Régions AWS), 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. ITILconstitue la base deITSM.

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 ITSM outils, consultez 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.

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, consultez la section Appliquer les autorisations du moindre privilège dans la IAM documentation.

lift and shift

Voir 7 Rs.

système de poids faible

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

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 au fur et à mesure de son fonctionnement. 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.

MES

Voir le système d'exécution de la fabrication.

Transport de télémétrie 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.

Migration Acceleration Program (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. MAPinclut une méthodologie de migration pour exécuter les migrations existantes de manière méthodique et 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 les opérations, les analystes commerciaux et les propriétaires, les ingénieurs de migration, les développeurs et les DevOps professionnels travaillant dans le cadre de sprints. Entre 20 et 50 % du portefeuille d'applications d'entreprise est constitué de modèles répétés qui peuvent être optimisés par une approche d'usine. Pour plus d'informations, veuillez consulter la rubrique discussion of migration factories et le guide Cloud Migration Factory dans cet ensemble de contenus.

métadonnées de migration

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

modèle de migration

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

Évaluation du portefeuille de migration (MPA)

Outil en ligne qui fournit des informations pour valider l'analyse de rentabilisation en faveur de la migration vers le. AWS Cloud MPAfournit une évaluation détaillée du portefeuille (dimensionnement correct des serveurs, tarification, TCO comparaisons, analyse des coûts de migration) ainsi que la planification de la migration (analyse et collecte des données des applications, regroupement des applications, hiérarchisation des migrations et planification des vagues). L'MPAoutil (nécessite une connexion) est disponible gratuitement pour tous les AWS consultants et consultants APN partenaires.

Évaluation de l'état de préparation à la migration (MRA)

Processus qui consiste à obtenir des informations sur l'état de préparation d'une entreprise 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. MRAest la première phase de la stratégie de AWS migration.

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

Voir 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-États-Unis

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. OPC-UA fournit une norme d'interopérabilité avec des schémas de cryptage, d'authentification et d'autorisation des données.

accord au niveau opérationnel () OLA

Un accord qui précise ce que les groupes informatiques fonctionnels s'engagent à fournir les uns aux autres, afin de soutenir un accord 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. OCMaide les organisations à se préparer et à passer à de nouveaux systèmes et stratégies en accélérant l'adoption des changements, en résolvant les problèmes de transition et en suscitant 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, consultez le OCMguide.

contrôle d'accès à l'origine (OAC)

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

identité d'accès à l'origine (OAI)

Dans CloudFront, une option permettant de restreindre l'accès afin de sécuriser votre contenu Amazon S3. Lorsque vous utilisezOAI, CloudFront crée un principal auprès duquel Amazon S3

peut s'authentifier. Les principaux authentifiés ne peuvent accéder au contenu d'un compartiment S3 que par le biais d'une distribution spécifique CloudFront . Voir également OAC, qui fournit un contrôle d'accès plus granulaire et amélioré.

ORR

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

DE

Voir technologie opérationnelle.

sortant (sortie) VPC

Dans une architecture AWS multi-comptes, 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

Une politique de IAM gestion attachée IAM aux principaux pour définir les autorisations maximales que l'utilisateur ou le rôle peut avoir. Pour plus d'informations, consultez la section Limites des autorisations dans la IAM documentation.

informations personnellement 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. PIILes exemples incluent les noms, les adresses et les coordonnées.

PII

Voir les informations personnelles identifiables.

manuel stratégique

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

PLC

Voir contrôleur logique programmable.

PLM

Consultez la section Gestion du cycle de vie des produits.

politique

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

persistance polyglotte

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

évaluation du portefeuille

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

predicate

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

prédicat pushdown

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

contrôle préventif

Contrôle de sécurité conçu pour empêcher qu'un événement ne se produise. Ces contrôles constituent une première ligne de défense pour empêcher tout accès non autorisé ou toute

modification indésirable de votre réseau. Pour plus d'informations, veuillez consulter Preventative controls dans Implementing security controls on AWS.

principal

Entité capable d'effectuer AWS des actions et d'accéder à des ressources. Cette entité est généralement un utilisateur root pour un Compte AWS, un IAM rôle ou un utilisateur. Pour plus d'informations, consultez les termes et concepts de Principal in Roles dans la IAM documentation.

Confidentialité dès la conception

Une approche de l'ingénierie des systèmes qui prend en compte le respect de la vie privée tout au long du processus d'ingénierie.

zones hébergées privées

Conteneur contenant des informations sur la manière dont vous souhaitez qu'Amazon Route 53 réponde aux DNS requêtes relatives à un domaine et à 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.

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 environnement basé sur des microservices MES, 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 SQL relationnelle.

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

RACImatrice

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

rançongiciel

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

RASCImatrice

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 du 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 la section 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, responsable, consultée, informée (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 RASCImatrice, et si vous l'excluez, elle est appelée RACImatrice.

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.

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)

L'utilisation d'SQLexpressions simples et flexibles qui ont défini des règles d'accès. RCACconsiste en 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

SAML2,0

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 AWS API opérations sans que vous ayez à créer un compte utilisateur IAM pour tous les membres de votre organisation. Pour plus d'informations sur la fédération SAML basée sur la version 2.0, consultez la section À propos de la fédération SAML basée sur la version 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.

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 combinent des systèmes de gestion des informations de sécurité (SIM) et de gestion des événements de sécurité (SEM). Un SIEM système 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 et 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 VPC de sécurité, l'application de correctifs à une EC2 instance Amazon ou la rotation des informations d'identification.

chiffrement côté serveur

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

politique de contrôle des services (SCP)

Politique qui propose un contrôle centralisé des autorisations pour tous les comptes d'une organisation dans AWS Organizations. SCPsdéfinissez 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

Le URL point d'entrée d'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.

accord 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 un exemple de la façon d'appliquer ce modèle, voir Modernisation de l'ancienne version de MicrosoftASP.NET(ASMX) des services Web de manière incrémentielle à l'aide de conteneurs et d'Amazon API Gateway.

sous-réseau

Une série d'adresses IP dans votreVPC. 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.

T

balises

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

variable cible

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

liste de tâches

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

environnement de test

Voir environnement.

entraînement

Pour fournir des données à partir desquelles votre modèle de ML peut apprendre. Les données d'entraînement doivent contenir la bonne réponse. L'algorithme d'apprentissage identifie des modèles dans les données d'entraînement, qui mettent en correspondance les attributs des données d'entrée avec la cible (la réponse que vous souhaitez prédire). Il fournit un modèle de ML qui capture ces modèles. Vous pouvez alors utiliser le modèle de ML pour obtenir des prédictions sur de nouvelles données pour lesquelles vous ne connaissez pas la cible.

passerelle de transit

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

flux de travail basé sur jonction

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

accès sécurisé

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

réglage

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

équipe de deux pizzas

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

U

incertitude

Un concept qui fait référence à des informations imprécises, incomplètes ou inconnues susceptibles de compromettre la fiabilité des modèles de ML prédictifs. Il existe deux types d'incertitude : l'incertitude épistémique est causée par des données limitées et incomplètes, alors que l'incertitude aléatoire est causée par le bruit et le caractère aléatoire inhérents aux données.

Pour plus d'informations, veuillez consulter le guide Quantifying uncertainty in deep learning systems.

tâches indifférenciées

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

environnements supérieurs

Voir environnement.

V

mise à vide

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

contrôle de version

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

VPCpeering

Une connexion entre deux VPCs qui vous permet d'acheminer le trafic en utilisant des adresses IP privées. Pour plus d'informations, consultez What is VPC peering dans la VPC documentation Amazon.

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

SQLFonction 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.

WORM

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 de type « 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.

application zombie

Application dont l'utilisation moyenne de CPU 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.