AWSLivre blanc

Principes de base d'AWS pour plusieurs régions

Copyright © 2023 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Principes de base d'AWS pour plusieurs régions: AWSLivre blanc

Copyright © 2023 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

Principes de base d'AWS pour plusieurs régions

Date de publication : 20 décembre 2022 (Révisions du document)

Résumé

Ce paper avancé de 300 niveaux est destiné aux architectes du cloud et aux cadres supérieurs qui créent des charges de travail et AWS qui souhaitent utiliser une architecture multirégionale pour améliorer la résilience de leurs charges de travail. Ce paper suppose une connaissance de base de l'AWSinfrastructure et des services. Il décrit les cas d'utilisation multirégionaux courants, partage les concepts multirégionaux fondamentaux et les implications en matière de conception, de développement et de déploiement, et fournit des conseils prescriptifs pour vous aider à mieux déterminer si une architecture multirégionale convient à vos charges de travail.

Êtes-vous Well-Architected ?

Le AWSWell-Architected Framework vous aide à comprendre les avantages et les inconvénients des décisions que vous prenez lors de la création de systèmes dans le cloud. Les six piliers du Framework vous permettent d'apprendre les meilleures pratiques architecturales pour concevoir et exploiter des systèmes fiables, sécurisés, efficaces, rentables et durables. À l'aide du AWS Well-Architected Tool, disponible gratuitement dans le AWS Management Console, vous pouvez évaluer votre charge de travail par rapport à ces meilleures pratiques en répondant à une série de questions pour chaque pilier.

Pour obtenir des conseils d'experts supplémentaires et les meilleures pratiques relatives à votre architecture cloud (déploiements d'architecture de référence, diagrammes et livres blancs), consultez le Centre d'architecture. AWS

Introduction

Chacune Région AWSse compose de plusieurs zones de disponibilité indépendantes et physiquement séparées au sein d'une zone géographique. Une séparation logique stricte entre les services logiciels de chaque région est maintenue. Cette conception ciblée garantit qu'une défaillance de l'infrastructure ou des services dans une région n'entraînera pas une défaillance corrélée dans une autre région.

La plupart des AWS clients peuvent atteindre leurs objectifs de résilience pour une charge de travail dans une seule région en utilisant plusieurs zones de disponibilité (AZ) ou AWS services régionaux. Cependant, un sous-ensemble de clients opte pour des architectures multirégionales pour trois raisons.

- Ils ont des exigences élevées en matière de disponibilité et de continuité des opérations pour leurs charges de travail les plus élevées qui, selon eux, ne peuvent être satisfaites dans une seule région.
- Ils doivent satisfaire aux exigences de souveraineté des données (telles que le respect des lois, réglementations et conformité locales) qui nécessitent des charges de travail pour fonctionner dans une juridiction donnée.
- Ils doivent améliorer les performances et l'expérience client en fonction de la charge de travail en exécutant les charges de travail sur les sites les plus proches des utilisateurs finaux.

Ce paper met l'accent sur les exigences de haute disponibilité et de continuité des opérations, et vous aide à comprendre les considérations relatives à l'adoption d'une architecture multirégionale pour une charge de travail. Nous décrivons les concepts fondamentaux qui s'appliquent à la conception, au développement et au déploiement d'une charge de travail multirégionale, ainsi qu'un cadre normatif pour vous aider à déterminer si une architecture multirégionale est le bon choix pour une charge de travail particulière. Vous devez vous assurer qu'une architecture multirégionale est le bon choix pour votre charge de travail, car ces architectures sont difficiles et il est possible que, si elles ne sont pas effectuées correctement, la disponibilité globale de la charge de travail diminue.

Ingénierie et exploitation au service de la résilience dans une seule région

Avant de vous plonger dans les concepts multirégionaux, commencez par vérifier que votre charge de travail est déjà aussi résiliente que possible dans une seule région. Pour ce faire, évaluez votre charge de travail par rapport au pilier de fiabilité et au pilier d'excellence opérationnelle du AWS Well-Architected Framework, et apportez les modifications nécessaires pour adopter les meilleures pratiques recommandées. Les concepts suivants sont abordés dans le AWS Well-Architected Framework :

- Segmentation de la charge de travail en fonction des limites du domaine
- Contrats de service bien définis
- Gestion des dépendances et couplage
- Gestion des échecs, des nouvelles tentatives et des stratégies de réduction
- Opérations idempotentes et transactions avec état ou sans état
- Préparation opérationnelle et gestion du changement
- Comprendre l'état des charges de travail
- Réagir aux événements

Pour approfondir la résilience d'une seule région, passez en revue et appliquez les concepts abordés dans Modèles de résilience multi-AZ avancés pour la gestion des défaillances grises. Ce paper présente les meilleures pratiques relatives à l'utilisation de répliques dans chaque zone de disponibilité pour contenir les défaillances, et développe les concepts multi-AZ introduits dans AWS Well Architected. Une fois que vous avez pleinement appliqué les concepts recommandés et les meilleures pratiques pour atteindre la plus haute résilience dans une seule région, une charge de travail spécifique peut être évaluée par rapport aux principes fondamentaux des architectures multirégionales afin de déterminer si la résilience de la charge de travail peut être augmentée à l'aide d'une approche multirégionale.

Fondement multirégional 1 : Comprendre les exigences

Comme indiqué précédemment, la haute disponibilité et la continuité des opérations sont des raisons courantes pour lesquelles on opte pour des architectures multirégionales. Les indicateurs de disponibilité mesurent le pourcentage de temps pendant lequel une charge de travail est disponible pour être utilisée sur une période définie, tandis que les indicateurs de continuité des opérations mesurent le rétablissement lors d'événements à grande échelle et généralement de plus longue durée.

La mesure de la disponibilité est un processus quasi continu. Les mesures ou métriques spécifiques peuvent varier, mais elles se fondent généralement autour d'un objectif de disponibilité, le plus souvent appelé neuf (disponibilité à 99,99 %, par exemple). En ce qui concerne les objectifs de disponibilité, il n'y a pas de solution universelle. Les objectifs de disponibilité doivent être établis au niveau de la charge de travail plutôt que d'appliquer un seul objectif à toutes les charges de travail, en séparant les composants non critiques des composants critiques.

Pour assurer la continuité des opérations, les point-in-time mesures suivantes sont généralement utilisées :

- Objectif de temps de rétablissement (RTO) — Le RTO est le délai maximum acceptable entre l'interruption du service et le rétablissement du service. Cette valeur détermine une durée acceptable pendant laquelle le service est perturbé.
- Objectif de point de restauration (RPO) — Le RPO est le délai maximum acceptable depuis le dernier point de récupération des données. Cela permet de déterminer ce qui est considéré comme une perte de données acceptable entre le dernier point de restauration et une interruption de service.

À l'instar de la définition d'objectifs de disponibilité, le RTO et le RPO doivent également être définis au niveau de la charge de travail. Pour atteindre une continuité des opérations plus agressive ou des exigences de haute disponibilité, des investissements accrus sont nécessaires. Cela dit, toutes les applications ne peuvent pas exiger ou ne nécessitent pas le même niveau de résilience. La création d'un mécanisme de hiérarchisation peut aider à établir le cadre permettant aux entreprises et aux responsables informatiques d'identifier les applications les plus exigeantes en fonction de leur impact sur l'entreprise, et de les hiérarchiser en conséquence. Vous trouverez des exemples de hiérarchisation dans les tableaux suivants.

Tableau 1 — Exemple de hiérarchisation de la résilience pour les SLA

Contrat de niveau de service (SLA) de disponibilité	Niveau de résilience	Temps d'arrêt acceptable/an
99,99 %	Platine	52,60 minutes
99,90 %	Doré	8,77 heures
99,5 %	Argenté	1,83 jours

Tableau 2 — Exemple de hiérarchisation de la résilience pour le RTO et le RPO

Palier	RTO maximal	RPO maximal	Critères	Coût
Platine	15 minutes	cinq minutes	Charges de travail critiques	$$$
Doré	15 minutes — six heures	deux heures	Charges de travail importantes, mais non critiques	$$
Argenté	six heures — quelques jours	24 heures	Charges de travail non critiques	$

Lors de la conception de charges de travail axées sur la résilience, il est nécessaire de comprendre la relation entre haute disponibilité et continuité des opérations. Par exemple, si une charge de travail nécessite une disponibilité de 99,99 %, un temps d'arrêt maximal de 53 minutes par an est tolérable. La détection d'une panne peut prendre au moins cinq minutes et un opérateur peut prendre dix minutes supplémentaires pour intervenir, prendre des décisions sur les étapes de restauration et exécuter ces étapes. Il n'est pas rare qu'il faille 30 à 45 minutes pour se rétablir d'un seul problème. Dans ce cas, l'adoption d'une stratégie multirégionale visant à fournir une instance isolée qui supprime l'impact corrélé peut permettre la poursuite des opérations en basculant dans un délai limité, tout en triant indépendamment la déficience initiale. C'est là que la définition du RTO et du RPO appropriés est nécessaire.

Pour les charges de travail critiques qui ont des besoins de disponibilité extrêmes (par exemple, une disponibilité de 99,99 % ou plus) ou des exigences strictes en matière de continuité des opérations qui ne peuvent être satisfaites qu'en basculant vers une autre région, une approche multirégionale peut être appropriée. Toutefois, ces exigences ne s'appliquent généralement qu'à un petit sous-ensemble du portefeuille de charges de travail d'une entreprise dont le temps de restauration est limité, mesuré en minutes ou en heures. À moins qu'une application n'ait besoin d'un temps de restauration de quelques minutes ou de quelques heures, il peut être préférable d'attendre qu'une interruption régionale de l'application soit corrigée dans la région affectée, et cela correspond généralement aux charges de travail de niveau inférieur.

Avant de mettre en œuvre une architecture multirégionale, les décideurs commerciaux et les équipes techniques doivent s'entendre sur les implications financières, y compris les facteurs de coûts opérationnels et d'infrastructure. Une architecture multirégionale typique peut entraîner une augmentation des coûts deux fois supérieure à celle d'une approche à région unique. Bien qu'il existe plusieurs modèles multirégionaux de continuité des activités, tels que le fonctionnement en veille chaude, en veille chaude et en veilleuse, le modèle présentant le moins de risques d'atteindre les objectifs de reprise impliquera le fonctionnement en mode veille chaud, ce qui doublera le coût de votre charge de travail.

Principaux conseils

- Les objectifs de disponibilité et de continuité des opérations tels que le RTO et le RPO doivent être établis par charge de travail et alignés sur les parties prenantes commerciales et informatiques.
- La plupart des objectifs de disponibilité et de continuité des opérations peuvent être atteints au sein d'une seule région. Pour les objectifs qui ne peuvent pas être atteints avec une seule région, il convient d'envisager le recours à plusieurs régions, en ayant une vision claire des compromis entre les coûts, la complexité et les avantages.

Deuxième élément fondamental pour plusieurs régions : comprendre les données

La gestion des données n'est pas un problème trivial avec les architectures multirégionales. La distance géographique entre les régions impose une latence inévitable, qui se traduit par le temps nécessaire pour répliquer les données entre les régions. Des compromis entre la disponibilité, la cohérence des données et l'introduction d'ordres de grandeur de latence plus élevés dans une charge de travail utilisant une architecture multirégionale seront nécessaires. Que vous utilisiez la réplication asynchrone ou synchrone, vous devez modifier votre application pour gérer les changements de comportement imposés par la technologie de réplication. Il est très difficile de transformer une application existante conçue pour une seule région en une application multirégionale en raison des problèmes liés à la cohérence des données et à la latence. Il est essentiel de comprendre les exigences de cohérence des données et les modèles d'accès aux données pour des charges de travail particulières pour évaluer les compromis.

2a : Comprendre les exigences de cohérence des données

Le théorème CAP fournit une référence pour raisonner sur les compromis entre la cohérence des données, la disponibilité et les partitions réseau, dont deux seulement peuvent être satisfaits en même temps pour une charge de travail. La multirégion inclut par définition les partitions réseau entre les régions. Vous devez donc choisir entre disponibilité et cohérence.

Si vous sélectionnez la disponibilité des données entre les régions, vous ne subirez pas de latence significative lors des écritures transactionnelles, car la réplication asynchrone des données validées entre les régions est tributaire, ce qui réduit la cohérence entre les régions jusqu'à la fin de la réplication. Avec la réplication asynchrone, en cas de panne dans la région principale, il y a une forte probabilité que des écritures soient en attente de réplication depuis la région principale. Cela conduit à un scénario dans lequel les données les plus récentes ne sont pas disponibles tant que la réplication ne reprend pas, et un processus de rapprochement est nécessaire pour gérer les transactions en vol qui n'ont pas été répliquées depuis la région qui a connu la panne.

Pour les charges de travail où la réplication asynchrone est privilégiée, vous pouvez utiliser des services tels qu'Amazon Aurora et Amazon DynamoDB, qui fournissent une réplication asynchrone entre régions. Les tables globales Amazon Aurora Global Database et Amazon DynamoDB disposent toutes deux de métriques CloudWatchAmazon par défaut pour faciliter le suivi du délai de réplication.

L'ingénierie de la charge de travail pour tirer parti des architectures axées sur les événements constitue un avantage pour une stratégie multirégionale, car cela signifie que la charge de travail peut inclure la réplication asynchrone des données et permet la reconstruction de l'état en rejouant les événements. Étant donné que les services de streaming et de messagerie mettent en mémoire tampon les données de charge utile des messages dans une seule région, un processus de basculement ou de retour en arrière régional doit inclure un mécanisme permettant de rediriger les flux de données d'entrée des clients, ainsi que de réconcilier les charges utiles en vol et/ou non livrées stockées dans la région qui a connu la panne.

Si la cohérence est sélectionnée, vous subirez une latence importante car les données sont répliquées de manière synchrone lors des écritures transactionnelles. Lorsque vous écrivez dans plusieurs régions de manière synchrone, si l'écriture échoue dans toutes les régions, la disponibilité est potentiellement réduite car la transaction ne sera pas validée et devra être réessayée. Les tentatives d'écriture synchrone des données dans toutes les régions se font au détriment de la latence à chaque tentative. À un moment donné, une fois les tentatives épuisées, il faudra décider soit d'échouer complètement la transaction, réduisant ainsi la disponibilité, soit de valider la transaction uniquement dans les régions disponibles, ce qui entraînera des incohérences. Il existe des technologies de formation de quorum telles que Paxos, qui peuvent aider à répliquer et à valider des données de manière synchrone, mais qui nécessitent un investissement important pour les développeurs.

Lorsque les écritures impliquent une réplication synchrone entre plusieurs régions pour répondre à de fortes exigences de cohérence, la latence d'écriture augmente d'un ordre de grandeur. Une latence d'écriture plus élevée ne peut généralement pas être intégrée ultérieurement à une application sans modifications importantes. Idéalement, il doit être pris en compte lors de la conception initiale de l'application. Pour les charges de travail multirégionales où la réplication synchrone est une priorité, les solutions AWS partenaires peuvent vous aider.

2b : Comprendre les modèles d'accès aux données

Les modèles d'accès aux données de charge de travail appartiennent à l'un des types suivants : lecture intensive ou écriture intensive. La compréhension de cette caractéristique pour une charge de travail particulière orientera le choix d'une architecture multirégionale appropriée.

Pour les charges de travail intensives en lecture, telles que le contenu statique entièrement en lecture seule, une architecture multirégionale active/active peut être réalisée sans complexité significative. La diffusion de contenu statique en périphérie à l'aide d'un réseau de distribution de contenu (CDN) garantit la disponibilité en mettant en cache le contenu le plus proche de l'utilisateur final ; l'utilisation

d'ensembles de fonctionnalités tels que le basculement d'Origin au sein d'Amazon CloudFront peut y contribuer. Une autre option consiste à déployer le calcul sans état dans plusieurs régions et à utiliser le DNS pour acheminer les utilisateurs vers la région la plus proche afin de lire le contenu. La Route 53 avec une politique de routage par géolocalisation peut être utilisée pour y parvenir.

Pour les charges de travail intensives en lecture dont le pourcentage de lectures est supérieur à celui des écritures, une stratégie globale de lecture locale et d'écriture peut être utilisée. Cela implique que toutes les écritures sont dirigées vers une base de données d'une région spécifique avec une réplication asynchrone des données vers toutes les autres régions, et des lectures peuvent être effectuées dans n'importe quelle région pour y parvenir. Cette approche nécessite une charge de travail pour garantir la cohérence finale, car les lectures locales peuvent être périmées en raison de la latence accrue liée à la réplication interrégionale des écritures.

Aurora Global Database peut vous aider à fournir des répliques de lecture dans une région de secours qui peut uniquement gérer l'ensemble du trafic de lecture localement, et une seule banque de données principale dans une région spécifique pour gérer les écritures. Les données sont répliquées de manière asynchrone depuis les bases de données principales vers les bases de données de secours (Read Replicas) et les bases de données de secours peuvent être promues au rang de base principale si vous devez effectuer des opérations de basculement vers la région de secours. Si une charge de travail convient mieux aux modèles de données non relationnels, DynamoDB peut également être utilisé dans cette approche. Encore une fois, la charge de travail doit être cohérente, ce qui peut nécessiter une réécriture si elle n'a pas été conçue pour cela dès le départ.

Pour les charges de travail nécessitant beaucoup d'écriture, une région principale doit être sélectionnée et la capacité de basculement vers une région de secours doit être intégrée à la charge de travail. Par rapport à une approche active/active, une approche principale/de réserve est moins compliquée. En effet, pour une architecture active/active, la charge de travail devra être réécrite pour gérer le routage intelligent vers les régions, établir une affinité de session, garantir des transactions idempotentes et gérer les conflits potentiels.

La plupart des charges de travail nécessitant une résilience multirégionale ne nécessiteront pas une approche active/active. Une stratégie de partitionnement peut être utilisée pour accroître la résilience en limitant le rayon d'action d'une déficience au sein de la clientèle. Si vous pouvez partager efficacement une base de clients, différentes régions principales peuvent être sélectionnées pour chaque partition. Par exemple, si vous pouvez partager des clients de manière à ce que la moitié des clients soient alignés sur la région 1 et l'autre moitié sur la région deux, en traitant les

régions comme des cellules, une approche cellulaire multirégionale peut être créée, ce qui permet de réduire le rayon d'impact de votre charge de travail.

L'approche de partitionnement peut être combinée à une approche principale/de secours afin de fournir des fonctionnalités de basculement pour les partitions. Un processus de basculement testé devra être intégré à la charge de travail et un processus de rapprochement des données devra également être conçu pour garantir la cohérence transactionnelle des magasins de données après le basculement. Elles sont abordées plus en détail plus loin dans ce paper.

Principaux conseils

- Il est fort probable que les écritures en attente de réplication ne soient pas validées dans la région de secours en cas d'échec. Les données ne seront pas disponibles jusqu'à ce que la réplication reprenne (dans l'hypothèse d'une réplication asynchrone).
- Dans le cadre du basculement, un processus de réconciliation des données sera nécessaire pour garantir le maintien d'un état de cohérence transactionnelle pour les banques de données utilisant la réplication asynchrone.
- Lorsqu'une forte cohérence est requise, les charges de travail doivent être modifiées pour tolérer la latence requise de la banque de données qui se réplique de manière synchrone.

Principe fondamental 3 pour plusieurs régions : comprendre les dépendances de votre charge de travail

Une charge de travail spécifique peut avoir plusieurs dépendances dans une région, telles que les AWS services utilisés, les dépendances internes, les dépendances tierces, les dépendances réseau, les certificats, les clés, les secrets et les paramètres. Pour garantir le fonctionnement de la charge de travail en cas de panne, il ne doit y avoir aucune dépendance entre la région principale et la région de secours ; chacune doit pouvoir fonctionner indépendamment l'une de l'autre. Pour ce faire, toutes les dépendances de la charge de travail doivent être examinées de près pour s'assurer qu'elles sont disponibles dans chaque région. Cela est nécessaire car une panne dans la région principale ne devrait pas avoir d'impact dans la région de secours. En outre, il est impératif de connaître le fonctionnement de la charge de travail lorsqu'une dépendance est dégradée ou totalement indisponible, afin que des solutions puissent être conçues pour gérer cette situation de manière appropriée.

3a : AWS services

Lors de la conception d'une architecture multirégionale, il est nécessaire de comprendre les AWS services spécifiques qui seront utilisés. Le premier aspect consiste à comprendre les fonctionnalités dont dispose le service pour permettre l'utilisation de plusieurs régions, et si une solution doit être conçue pour atteindre les objectifs multirégionaux. Par exemple, avec Amazon Aurora et Amazon DynamoDB, il existe une fonctionnalité permettant de répliquer les données de manière asynchrone vers une région de secours. Toutes les dépendances de AWS service devront être disponibles dans toutes les régions à partir desquelles une charge de travail sera exécutée. Pour vous assurer que les services qui seront utilisés sont disponibles dans les régions souhaitées, consultez la liste de Région AWS tous les services.

3b : Dépendances internes et tierces

En ce qui concerne les dépendances internes d'une charge de travail, assurez-vous qu'elle est disponible auprès des régions à partir desquelles la charge de travail fonctionnera. Par exemple, si la charge de travail est composée de nombreux microservices, connaissez tous les microservices qui constituent une fonctionnalité commerciale. À partir de là, assurez-vous que tous ces microservices sont déployés dans chaque région à partir de laquelle la charge de travail fonctionnera.

Les appels interrégionaux entre microservices au sein d'une même charge de travail ne sont pas conseillés, et l'isolement régional doit être maintenu. En effet, la création de dépendances entre régions augmente le risque de défaillance corrélée, ce qui annule les avantages que vous essayez d'obtenir avec des implémentations régionales isolées de la charge de travail. Les dépendances sur site peuvent également faire partie de la charge de travail. Il est donc impératif de comprendre comment les caractéristiques de ces intégrations pourraient changer si la région principale devait changer. Par exemple, si la région de veille est située plus loin de l'environnement sur site, l'augmentation de la latence aura un impact négatif.

Comprendre les solutions SaaS (Software as a Service), les kits de développement logiciel (SDK) et les autres dépendances entre produits tiers, et être en mesure de mettre en œuvre des scénarios dans lesquels ces dépendances sont dégradées ou indisponibles permettra de mieux comprendre le fonctionnement et le comportement de la chaîne de systèmes en fonction des différents modes de défaillance. Ces dépendances peuvent se trouver dans le code d'une application, qu'il s'agisse de la façon dont les secrets sont gérés en externe à l'aide d'AWS Secrets Manager ou d'une solution de coffre-fort tierce (telle que Hashicorp), ou de systèmes d'authentification dépendant d'IAM Identity Center pour les connexions fédérées.

La redondance en matière de dépendances peut contribuer à accroître la résilience. Il est également possible qu'une solution SaaS ou une dépendance tierce utilise le même primaire Région AWS que la charge de travail. Dans ce cas, vous devez travailler avec le fournisseur pour déterminer si sa posture de résilience correspond aux exigences de la charge de travail.

En outre, soyez conscient du destin partagé entre la charge de travail et ses dépendances, telles que les applications tierces. Si les dépendances ne sont pas disponibles dans (ou depuis) une région secondaire après un basculement, la charge de travail risque de ne pas être complètement rétablie.

3c : Mécanisme de basculement

Le système de noms de domaine (DNS) est couramment utilisé comme mécanisme de basculement pour transférer le trafic de la région principale vers une région de secours. Passez en revue et examinez de manière critique toutes les dépendances prises par le mécanisme de basculement. Par exemple, si votre charge de travail utilise Amazon Route 53, comprendre que le plan de contrôle est hébergé dans US-East-1 signifie que vous devenez dépendant du plan de contrôle de cette région spécifique. Cela n'est pas recommandé dans le cadre d'un mécanisme de basculement si la région principale est également US-East-1. Si un autre mécanisme de basculement est utilisé, il est nécessaire de bien comprendre tout scénario dans lequel il ne fonctionnerait pas comme prévu. Une fois cette compréhension établie, planifiez les mesures d'urgence ou élaborez un nouveau

mécanisme si nécessaire. Consultez la section Création de mécanismes de reprise après sinistre à l'aide d'Amazon Route 53 pour découvrir les approches que vous pouvez utiliser pour réussir le basculement sur incident.

Comme indiqué dans la section sur les dépendances internes, tous les microservices faisant partie d'une capacité commerciale doivent être disponibles dans chaque région dans laquelle la charge de travail est déployée. Dans le cadre de la stratégie de basculement, les capacités de l'entreprise doivent basculer simultanément pour éliminer le risque d'appels interrégionaux. Par ailleurs, si les microservices basculent indépendamment, cela peut entraîner un comportement indésirable dans lequel les microservices peuvent effectuer des appels interrégionaux, ce qui entraîne une latence et peut entraîner l'indisponibilité de la charge de travail en cas d'expiration du délai d'attente du client.

3d : Dépendances de configuration

Les certificats, les clés, les secrets et les paramètres font partie de l'analyse de dépendance nécessaire lors de la conception pour plusieurs régions. Dans la mesure du possible, il est préférable de localiser ces composants dans chaque région afin qu'ils ne soient pas partagés entre les régions en ce qui concerne ces dépendances. Pour les certificats, l'expiration doit varier selon les régions et, si possible, dans chaque région, afin d'éviter qu'un certificat expirant (avec des alarmes configurées pour avertir à l'avance) ait un impact sur plusieurs régions.

Les clés et les secrets de chiffrement doivent également être spécifiques à la région. Ainsi, en cas d'erreur lors de la rotation d'une clé ou d'un secret, l'impact est limité à une région spécifique.

Enfin, tous les paramètres de charge de travail doivent être stockés localement pour que la charge de travail puisse être récupérée dans la région spécifique.

Principaux conseils

- Une architecture multirégionale bénéficie de la séparation physique et logique entre les régions. L'introduction de dépendances entre régions au niveau de la couche d'application annule cet avantage. Évitez de telles dépendances.
- Les contrôles de basculement devraient fonctionner sans aucune dépendance vis-à-vis de la région principale.
- La coordination du basculement au niveau des capacités de l'entreprise doit être effectuée pour éliminer la possibilité d'une latence et d'une dépendance accrues des appels interrégionaux.

Élément fondamental multirégional 4 : Préparation opérationnelle

L'exploitation d'une charge de travail multirégionale est une tâche complexe qui comporte des défis opérationnels spécifiques à plusieurs régions. Il s'agit notamment Compte AWS de la gestion, de la refonte des processus de déploiement, de la création d'une stratégie d'observabilité multirégionale, de la création et du test d'un runbook de basculement et de reprise, puis de la gestion des coûts. Un examen du niveau de préparation opérationnelle (ORR) peut aider les équipes à préparer une charge de travail pour la production, qu'elle soit exécutée dans une seule région ou dans plusieurs régions.

4a : Compte AWS gestion

Pour déployer une charge de travail entre plusieurs régionsRégions AWS, assurez-vous que tous les quotas de AWS service d'un compte sont égaux. Tout d'abord, découvrez tous les AWS services qui font partie de l'architecture, examinez l'utilisation prévue dans les régions de secours, puis comparez-les à l'utilisation actuelle. Dans certains cas, si la région de secours n'a jamais été utilisée auparavant, vous pouvez vous référer aux quotas de service par défaut pour comprendre le point de départ. Ensuite, pour tous les services qui seront utilisés, demandez une augmentation de quota à l'aide de la console Service Quotas (connexion requise) ou des API.

AWSLes rôles Identity and Access Management (IAM) doivent être configurés dans chaque région pour garantir que les opérateurs, les outils d'automatisation et les AWS services disposent des autorisations appropriées sur les ressources de la région de secours. L'isolation des rôles au niveau régional permet d'atteindre l'isolement régional que nous recherchons pour les architectures multirégionales. Assurez-vous que ces autorisations sont en place avant de passer en ligne avec une région en veille.

4b : Pratiques de déploiement

Grâce aux fonctionnalités multirégionales, le déploiement de la charge de travail dans plusieurs régions peut s'avérer complexe. AWS CloudFormationpermet de déployer l'infrastructure dans une ou plusieurs régions et peut être adaptée en fonction de vos besoins. AWS CodePipelinepermet de fournir un pipeline d'intégration/de livraison continue (CI/CD) presque continu, qui comporte des actions interrégionales qui permettent le déploiement dans des régions différentes de la région dans laquelle se trouve le pipeline. Ceci, combiné à des stratégies de déploiement robustes telles que le bleu/vert, permet un déploiement de temps d'arrêt minimal, voire nul.

Toutefois, le déploiement de fonctionnalités dynamiques peut être plus complexe lorsque l'état de l'application ou des données n'est pas externalisé vers un magasin persistant. Dans ces situations, adaptez soigneusement le processus de déploiement à vos besoins. Concevez le pipeline et le processus de déploiement pour déployer dans une région à la fois, plutôt que dans plusieurs régions simultanément. Cela réduit le risque de défaillances corrélées entre les régions. Pour en savoir plus sur les techniques utilisées par Amazon pour automatiser les déploiements de logiciels, consultez l'article Automating safe and handoff deployments de la Builder Library.

4c : Observabilité

Lors de la conception pour plusieurs régions, réfléchissez à la manière dont la santé de tous les composants de chaque région sera surveillée afin d'obtenir une vision globale de la santé régionale. Cela peut inclure des mesures de surveillance du délai de réplication, ce qui n'est pas pris en compte pour la charge de travail d'une seule région.

Lorsque vous créez une architecture multirégionale, pensez également à observer les performances de la charge de travail depuis les régions de secours. Cela inclut un bilan de santé et des canaris (tests synthétiques) effectués depuis la région d'attente, afin de fournir un aperçu extérieur de l'état de santé du primaire. En outre, vous pouvez utiliser Amazon CloudWatch Internet Monitor pour comprendre l'état du réseau externe et les performances de vos charges de travail du point de vue de l'utilisateur final. De même, la région principale doit disposer de la même observabilité pour surveiller la région de réserve. Ces canaris devraient surveiller les indicateurs de l'expérience client afin de se faire une idée globale de la charge de travail. Cela est nécessaire car s'il y avait un problème dans la région principale, l'observabilité dans la région principale pourrait être altérée et cela aurait une incidence sur la capacité d'évaluer l'état de la charge de travail.

Dans ce cas, l'observation en dehors de cette région peut fournir des informations. Ces mesures doivent être regroupées dans des tableaux de bord disponibles dans chaque région, et des alarmes doivent être créées dans chaque région. Amazon étant un service régional, il CloudWatch est indispensable de disposer de ces services dans les deux régions. Ces données de surveillance seront utilisées pour effectuer l'appel de basculement d'une région principale à une région de secours.

4d : Processus, procédures et tests

C'est le meilleur moment pour répondre à la question « Quand dois-je basculer ? » est bien avant que vous n'en ayez besoin. Les plans de continuité des activités incluant les personnes, les processus

et les technologies doivent tous être définis bien avant la survenue d'un problème et être testés régulièrement. Décidez d'un cadre décisionnel en matière de recouvrement. S'il existe un processus de restauration bien rodé et que le délai de restauration est bien connu, il est possible de choisir le moment où démarrer le processus de restauration qui répond à l'objectif du RTO par le biais d'un basculement sur incident. Cela peut se produire immédiatement après l'identification d'un problème lié à l'application dans la région principale, ou après un événement où les options de restauration de l'application dans la région ont été épuisées et devraient maintenant commencer un basculement pour répondre au RTO.

Bien que l'action de basculement elle-même doive être automatisée à 100 %, la décision d'activer le basculement doit être prise par un humain (généralement un petit nombre de personnes prédéterminées au sein de l'organisation). En outre, les critères permettant de décider d'un basculement doivent être clairement définis et compris globalement par l'organisation. Ces processus peuvent être définis et complétés à l'aide des runbooks de AWS System Manager, ce qui permet une end-to-end automatisation complète et garantit la cohérence du processus en cours d'exécution pendant les tests et le basculement.

Ces runbooks doivent être disponibles dans la région principale et dans la région de secours pour démarrer les processus de basculement ou de retour en arrière. Une fois cette automatisation mise en place, une cadence de test régulière doit être définie et suivie. Cela garantit que lorsqu'un événement se produit, la réponse est exécutée selon un processus bien défini et mis en pratique dans lequel l'organisation a confiance. Il est également important de garder à l'esprit les tolérances établies pour les processus de rapprochement des données. Confirmez que les exigences RPO/RTO établies sont respectées avec le processus proposé.

4e : Coût et complexité

Les implications financières d'une architecture multirégionale dépendent de l'augmentation de l'utilisation de l'infrastructure, des frais opérationnels et du temps consacré aux ressources. Comme indiqué précédemment, le coût d'infrastructure dans une région de secours est similaire au coût d'infrastructure dans une région principale lors du préapprovisionnement, soit deux fois le coût. Fournissez de la capacité de manière à ce qu'elle soit suffisante pour les opérations quotidiennes, tout en réservant une capacité tampon suffisante pour tolérer les pics de demande, et configurez les mêmes limites dans chaque région.

En outre, des modifications au niveau de l'application peuvent être nécessaires pour fonctionner correctement dans une architecture multirégionale si vous adoptez une architecture active-active, dont la conception et l'exploitation peuvent nécessiter beaucoup de temps et de ressources. Au

minimum, les entreprises devraient consacrer du temps à comprendre les dépendances techniques et commerciales de chaque région et à concevoir des processus de basculement et de retour sur incident.

Les équipes devraient également effectuer des exercices de basculement et de retour en arrière normaux pour se sentir à l'aise avec les runbooks qui seront utilisés lors d'un événement. Bien qu'ils soient extrêmement importants et cruciaux pour obtenir le résultat attendu d'un investissement multirégional, ces exercices représentent un coût d'opportunité et privent du temps et des ressources d'autres activités.

Principaux conseils

- AWSles quotas de service doivent être revus et être égaux dans toutes les régions où la charge de travail s'appliquera.
- Le processus de déploiement doit cibler une région à la fois, plutôt que plusieurs régions simultanément.
- Des mesures supplémentaires, telles que le délai de réplication, doivent être surveillées et sont spécifiques aux scénarios multirégionaux.
- Étendre la surveillance de la charge de travail au-delà de la région principale. Les indicateurs de l'expérience client doivent être surveillés par région et mesurés en dehors de chaque région dans laquelle une charge de travail est exécutée.
- Le failover et le failback doivent être testés régulièrement. Garantissez la mise en œuvre d'un runbook unique pour les processus de basculement et de reprise, utilisé à la fois lors des tests et lors d'un événement en direct. Les runbooks destinés aux tests et aux événements en direct ne peuvent pas être différents.

Conclusion

Ce livre blanc décrit les cas d'utilisation courants de l'architecture multirégionale, les principes fondamentaux de la mise en œuvre d'une architecture multirégionale et les implications de cette approche. Ces principes fondamentaux peuvent être appliqués à n'importe quelle charge de travail et utilisés comme cadre pour aider à décider si une architecture multirégionale est la bonne approche pour une entreprise en particulier.

Collaborateurs

Les personnes qui ont contribué à ce document incluent :

Contributeur technique :

- John Formento, Jr., architecte de solutions principal, équipe AWS multirégionale

Contributeur éditorial :

- Lisi Lewis, directrice principale du marketing des produits

Suggestions de lecture

Pour plus d'informations, reportez-vous à :

- Modèles de résilience multi-AZ avancés (AWSlivre blanc)
- Pilier de fiabilité - AWS Well-Architected Framework
- Disponibilité et au-delà : comprendre et améliorer la résilience des systèmes distribués sur AWS (AWSlivre blanc)
- AWSLimites d'isolation des défauts (AWSlivre blanc)

Révisions du document

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

Modification	Description	Date
Document publié	Première publication.	20 décembre 2022

Avis

Il incombe aux clients de procéder à une évaluation indépendante des informations contenues dans le présent document. Ce document : (a) est fourni à titre informatif uniquement, (b) représente les offres de produits et les pratiques AWS actuelles, qui sont sujettes à modification sans préavis, et (c) ne donne lieu à aucun engagement ni aucune assurance de la part d'AWS et de ses sociétés apparentées, fournisseurs ou concédants de licence. Les produits ou services AWS sont fournis « tels quels » sans garantie, représentation ou condition d'aucune sorte, tant expresse qu'implicite. Les responsabilités et obligations d'AWS vis-à-vis de ses clients sont régies par les contrats AWS. Le présent document ne fait partie d'aucun, et ne modifie aucun, contrat entre AWS et ses clients.

© 2022, Amazon Web Services, Inc. ou ses sociétés apparentées. Tous droits réservés.

Glossaire AWS

Pour connaître la terminologie la plus récente d’AWS, consultez le Glossaire AWS dans la Référence Glossaire AWS.

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.