Whitepaper do AWS

Fundamentos de várias regiões da AWS

Fundamentos de várias regiões da AWS: Whitepaper do AWS

Table of Contents

Fundamentos de várias regiões da AWS

Data de publicação: 20 de dezembro de 2022 (Revisões do documento)

Resumo

Este documento avançado de 300 níveis é destinado a arquitetos de nuvem e líderes seniores que criam cargas de trabalho com base em AWS pessoas interessadas em usar uma arquitetura multirregional para melhorar a resiliência de suas cargas de trabalho. Este paper pressupõe um conhecimento básico de AWS infraestrutura e serviços. Ele descreve casos de uso comuns de várias regiões, compartilha conceitos e implicações fundamentais de várias regiões sobre design, desenvolvimento e implantação e fornece orientação prescritiva para ajudá-lo a determinar melhor se uma arquitetura multirregional é adequada para suas cargas de trabalho.

Você é Well-Architected?

O AWSWell-Architected Framework ajuda você a entender os prós e os contras das decisões tomadas ao criar sistemas na nuvem. Os seis pilares do Framework permitem que você aprenda as melhores práticas arquitetônicas para projetar e operar sistemas confiáveis, seguros, eficientes, economicamente viáveis e sustentáveis. Usando o AWS Well-Architected Tool, disponível gratuitamente no AWS Management Console, você pode analisar seus workloads em relação a essas melhores práticas respondendo a um conjunto de perguntas para cada pilar.

Para obter orientações especializadas e melhores práticas adicionais para a arquitetura de sua nuvem (implantações de arquitetura de referência, diagramas e whitepapers), consulte o Centro de arquitetura da AWS.

Introdução

Cada uma Região da AWSconsiste em várias zonas de disponibilidade independentes e fisicamente separadas dentro de uma área geográfica. A separação lógica estrita entre os serviços de software em cada região é mantida. Esse design proposital garante que uma falha na infraestrutura ou nos serviços em uma região não resulte em uma falha correlacionada em outra região.

A maioria dos AWS clientes pode atingir seus objetivos de resiliência para uma carga de trabalho em uma única região usando várias zonas de disponibilidade (AZs) ou serviços regionais. AWS No entanto, um subconjunto de clientes busca arquiteturas multirregionais por três motivos.

- Eles têm requisitos de alta disponibilidade e continuidade de operações para suas cargas de trabalho de nível mais alto, que acreditam que não podem ser atendidos em uma única região.
- Eles precisam atender aos requisitos de soberania de dados (como a adesão às leis, regulamentações e conformidade locais), que exigem que as cargas de trabalho operem dentro de uma determinada jurisdição.
- Eles precisam melhorar o desempenho e a experiência do cliente para a carga de trabalho executando as cargas de trabalho em locais mais próximos aos usuários finais.

Este paper se concentra nos requisitos de alta disponibilidade e continuidade das operações e ajuda você a abordar as considerações para a adoção de uma arquitetura multirregional para uma carga de trabalho. Descrevemos conceitos fundamentais que se aplicam ao design, desenvolvimento e implantação de uma carga de trabalho multirregional, juntamente com uma estrutura prescritiva para ajudá-lo a determinar se uma arquitetura multirregional é a escolha certa para uma carga de trabalho específica. Você precisa garantir que uma arquitetura multirregional seja a escolha certa para sua carga de trabalho, porque essas arquiteturas são desafiadoras e é possível que, se não forem feitas corretamente, a disponibilidade geral da carga de trabalho diminua.

Engenharia e operação para resiliência em uma única região

Antes de mergulhar nos conceitos de várias regiões, comece confirmando que sua carga de trabalho já é a mais resiliente possível em uma única região. Para conseguir isso, avalie sua carga de trabalho em relação ao Pilar de Confiabilidade e ao Pilar de Excelência Operacional do AWS Well-Architected Framework e faça as alterações necessárias para adotar as melhores práticas recomendadas. Os seguintes conceitos são abordados no AWS Well-Architected Framework:

- Segmentação da carga de trabalho com base nos limites do domínio
- Contratos de serviço bem definidos
- Gerenciamento de dependências e acoplamento
- Lidando com falhas, novas tentativas e estratégias de recuo
- Operações idempotentes e transações com estado versus transações sem estado
- Prontidão operacional e gerenciamento de mudanças
- Entendendo a integridade da carga de trabalho
- Respondendo a eventos

Para levar mais longe a resiliência de uma única região, revise e aplique os conceitos discutidos em Padrões avançados de resiliência Multi-AZ para lidar com falhas cinzentas. Este paper fornece as melhores práticas sobre o uso de réplicas em cada zona de disponibilidade para conter falhas e expande os conceitos Multi-AZ introduzidos no AWS Well Architected. Depois de aplicar totalmente os conceitos recomendados e as melhores práticas para alcançar a maior resiliência em uma única região, uma carga de trabalho específica pode ser avaliada em relação aos fundamentos de arquiteturas multirregionais para determinar se a resiliência da carga de trabalho pode ser aumentada usando uma abordagem multirregional.

Princípio fundamental da multirregião 1: Compreender os requisitos

Conforme mencionado anteriormente, a alta disponibilidade e a continuidade das operações são motivos comuns para buscar arquiteturas multirregionais. As métricas de disponibilidade medem a porcentagem de tempo em que uma carga de trabalho está disponível para uso em um período definido, enquanto as métricas de continuidade das operações medem a recuperação de eventos de grande escala e, normalmente, de maior duração.

Medir a disponibilidade é um processo quase contínuo. Medidas ou métricas específicas podem variar, mas normalmente se aglutinam em torno de uma disponibilidade alvo, geralmente chamada de nove (como disponibilidade de 99,99%). Com metas de disponibilidade, um tamanho único não serve para todos. As metas de disponibilidade precisam ser estabelecidas em um nível de carga de trabalho em vez de aplicar uma única meta em todas as cargas de trabalho, separando os componentes não críticos dos críticos.

Para a continuidade das operações, as seguintes point-in-time medidas são normalmente usadas:

- Objetivo de tempo de recuperação (RTO) — RTO é o atraso máximo aceitável entre a interrupção do serviço e a restauração do serviço. Esse valor determina uma duração aceitável pela qual o serviço está comprometido.
- Objetivo do Ponto de Recuperação (RPO) — O RPO é o tempo máximo aceitável desde o último ponto de recuperação de dados. Isso determina o que é considerado uma perda de dados aceitável entre o ponto de recuperação mais recente e a interrupção do serviço.

Assim como definir metas de disponibilidade, o RTO e o RPO também devem ser definidos no nível da carga de trabalho. Para alcançar uma continuidade mais agressiva das operações ou requisitos de alta disponibilidade, é necessário aumentar o investimento. Dito isso, nem todo aplicativo pode exigir ou exigir o mesmo nível de resiliência. A criação de um mecanismo de hierarquização pode ajudar a estabelecer a estrutura para alinhar os proprietários de negócios e de TI na identificação dos aplicativos mais exigentes com base no impacto nos negócios e hierarquizá-los adequadamente. Exemplos de hierarquização podem ser encontrados nas tabelas a seguir.

Tabela 1 — Exemplo de escalonamento de resiliência para SLA

Acordo de nível de serviço (SLA) de disponibilidade	Nível de resiliência	Tempo de inatividade aceitável/ano
99,99%	Platina	52.60 minutos
99,90%	Ouro	8.7 horas
99,5%	Prata	1,83 dias

Tabela 2 — Exemplo de hierarquização de resiliência para RTO e RPO

Nível	RTO máximo	RPO máximo	Critérios	Custo
Platina	15 minutos	cinco minutos	Cargas de trabalho de missão crítica	$$$
Ouro	15 minutos — seis horas	duas horas	Cargas de trabalho importantes, mas não essenciais	$$
Prata	seis horas — alguns dias	24 horas	Cargas de trabalho não críticas	$

Ao projetar cargas de trabalho para resiliência, é necessário entender a relação entre alta disponibilidade e continuidade das operações. Por exemplo, se uma carga de trabalho exigir 99,99% de disponibilidade, não mais do que 53 minutos de inatividade por ano são toleráveis. Pode levar pelo menos cinco minutos para detectar uma falha e outros dez minutos para que um operador se envolva, tome decisões sobre as etapas de recuperação e execute essas etapas. Não é incomum que um único problema leve de 30 a 45 minutos para ser recuperado. Nesse caso, ter uma estratégia multirregional para fornecer uma instância isolada que elimine o impacto correlacionado pode permitir a continuidade das operações por meio de failover dentro de um tempo limitado e, ao mesmo tempo, fazer a triagem da deficiência inicial de forma independente. É aqui que é necessário definir o RTO e o RPO apropriados.

Para cargas de trabalho de missão crítica que têm necessidades extremas de disponibilidade (por exemplo, 99,99% de disponibilidade ou mais) ou requisitos rigorosos de continuidade de operações que só podem ser atendidos por meio de failover para outra região, uma abordagem multirregional pode ser apropriada. No entanto, esses requisitos geralmente são aplicáveis apenas a um pequeno subconjunto do portfólio de carga de trabalho de uma empresa que tem um tempo de recuperação limitado medido em minutos ou horas. A menos que um aplicativo precise de um tempo de recuperação de minutos ou algumas horas, esperar que uma interrupção regional do aplicativo seja corrigida na região afetada pode ser uma abordagem melhor e normalmente está alinhada com cargas de trabalho de nível inferior.

Antes de implementar uma arquitetura multirregional, os tomadores de decisão de negócios e as equipes técnicas devem estar alinhados com as implicações de custo, incluindo os fatores de custo operacional e de infraestrutura. Uma arquitetura típica de várias regiões pode gerar um aumento de custo de duas vezes em relação a uma abordagem de região única. Embora existam vários padrões multirregionais para a continuidade dos negócios, como operar com espera quente, espera quente e luz piloto, o padrão com o menor risco de atingir os objetivos de recuperação envolverá a execução de espera dinâmica e dobrará o custo de sua carga de trabalho.

Orientação-chave

- As metas de disponibilidade e continuidade das operações, como RTO e RPO, devem ser estabelecidas por carga de trabalho e alinhadas com os negócios e as partes interessadas de TI.
- A maioria das metas de disponibilidade e continuidade das operações pode ser alcançada em uma única região. Para metas que não podem ser alcançadas com uma única região, deve-se considerar a multirregião, com uma visão clara das compensações entre custo, complexidade e benefícios.

Fundamental multirregional 2: compreender os dados

O gerenciamento de dados não é um problema trivial com arquiteturas multirregionais. A distância geográfica entre regiões impõe uma latência inevitável, que se manifesta como o tempo necessário para replicar dados entre regiões. Serão necessárias compensações entre disponibilidade, consistência de dados e introdução de ordens maiores de magnitude de latência em uma carga de trabalho que usa uma arquitetura multirregional. Seja usando a replicação assíncrona ou síncrona, você precisará modificar seu aplicativo para lidar com as mudanças comportamentais impostas pela tecnologia de replicação. É muito difícil transformar um aplicativo existente projetado para uma única região em várias regiões devido aos desafios relacionados à consistência e latência dos dados. Compreender os requisitos de consistência de dados e os padrões de acesso aos dados para cargas de trabalho específicas é fundamental para ponderar as compensações.

2a: Entendendo os requisitos de consistência de dados

O teorema CAP fornece uma referência para raciocinar sobre as compensações entre consistência de dados, disponibilidade e partições de rede, das quais apenas duas podem ser satisfeitas ao mesmo tempo para uma carga de trabalho. A multirregião, por definição, inclui partições de rede entre regiões, então você precisa escolher entre disponibilidade e consistência.

Se você selecionar a disponibilidade dos dados em todas as regiões, não incorrerá em latência significativa durante as gravações transacionais, pois há uma dependência da replicação assíncrona dos dados comprometidos entre as regiões, resultando na redução da consistência entre as regiões até que a replicação seja concluída. Com a replicação assíncrona, quando há uma falha na região primária, há uma alta probabilidade de gravações pendentes de replicação da região primária. Isso leva a um cenário em que os dados mais recentes ficam indisponíveis até que a replicação seja retomada, e um processo de reconciliação é necessário para lidar com transações em andamento que não foram replicadas da região que sofreu a interrupção.

Para cargas de trabalho em que a replicação assíncrona é preferida, você pode usar serviços como o Amazon Aurora e o Amazon DynamoDB, que fornecem replicação assíncrona entre regiões. As tabelas globais do Amazon Aurora Global Database e do Amazon DynamoDB têm métricas padrão da CloudWatchAmazon para ajudar a monitorar o atraso na replicação.

Projetar a carga de trabalho para aproveitar as arquiteturas orientadas por eventos é um benefício para uma estratégia multirregional, pois significa que a carga de trabalho pode adotar a replicação assíncrona de dados e permite a reconstrução do estado por meio da repetição de eventos. Como

os serviços de streaming e mensagens armazenam em buffer os dados da carga útil das mensagens em uma única região, um processo regional de failover/failback deve incluir um mecanismo para redirecionar os fluxos de dados de entrada do cliente, bem como reconciliar cargas em andamento e/ou não entregues armazenadas na região que sofreu a interrupção.

Se a consistência for selecionada, você incorrerá em latência significativa à medida que os dados forem replicados de forma síncrona durante as gravações transacionais. Ao gravar em várias regiões de forma síncrona, se a gravação não for bem-sucedida em todas as regiões, a disponibilidade é potencialmente reduzida porque a transação não será confirmada e precisará ser repetida. As novas tentativas de gravar os dados em todas as regiões de forma síncrona são feitas à custa da latência em cada tentativa. Em algum momento, quando as novas tentativas forem esgotadas, será necessário tomar a decisão de falhar completamente na transação, reduzindo assim a disponibilidade, ou confirmar a transação somente nas regiões disponíveis, causando inconsistência. Existem tecnologias de formação de quórum, como a Paxos, que podem ajudar a replicar e confirmar dados de forma síncrona, mas que precisam de um investimento significativo do desenvolvedor.

Quando as gravações envolvem replicação síncrona em várias regiões para atender aos fortes requisitos de consistência, a latência de gravação aumenta em uma ordem de magnitude. Uma latência de gravação mais alta não é algo que normalmente possa ser adaptado a um aplicativo sem alterações significativas. Idealmente, isso deve ser levado em consideração quando o aplicativo está sendo projetado pela primeira vez. Para cargas de trabalho multirregionais em que a replicação síncrona é uma prioridade, as soluções de AWSparceiros podem ajudar.

2b: Entendendo os padrões de acesso aos dados

Os padrões de acesso aos dados da carga de trabalho se enquadram em um dos seguintes tipos: leitura intensiva ou gravação intensiva. A compreensão dessa característica para uma carga de trabalho específica orientará a seleção de uma arquitetura multirregional apropriada.

Para cargas de trabalho com uso intenso de leitura, como conteúdo estático que é totalmente somente para leitura, é possível obter uma arquitetura multirregional ativa/ativa sem complexidade significativa. A veiculação de conteúdo estático na borda usando uma Rede de Distribuição de Conteúdo (CDN) garante a disponibilidade ao armazenar em cache o conteúdo mais próximo do usuário final. Usar conjuntos de recursos como o failover Origin na Amazon CloudFront pode ajudar a conseguir isso. Outra opção é implantar a computação sem estado em várias regiões e usar o DNS para direcionar os usuários para a região mais próxima para ler o conteúdo. O Route 53 com política de roteamento de geolocalização pode ser usado para conseguir isso.

Para cargas de trabalho de leitura intensiva que têm uma porcentagem maior de leituras do que gravações, uma estratégia global de leitura local e gravação global pode ser usada. Isso implica que todas as gravações vão para um banco de dados em uma região específica com replicação assíncrona de dados para todas as outras regiões, e as leituras podem ser feitas em qualquer região para conseguir isso. Essa abordagem exige que uma carga de trabalho adote uma consistência eventual, pois as leituras locais podem ficar obsoletas devido ao aumento da latência na replicação de gravações entre regiões.

O Aurora Global Database pode ajudar no provisionamento de réplicas de leitura em uma região de espera que pode lidar exclusivamente com todo o tráfego de leitura localmente e com um único armazenamento de dados primário em uma região específica para lidar com gravações. Os dados são replicados de forma assíncrona dos bancos de dados primário para o stand-by (réplicas de leitura) e os bancos de dados stand-by podem ser promovidos para primários se você precisar fazer o failover das operações para a região standby. Se uma carga de trabalho for mais adequada para modelos de dados não relacionais, o DynamoDB também pode ser usado nessa abordagem. Novamente, a carga de trabalho precisa adotar uma consistência eventual, o que pode exigir que ela seja reescrita se não tiver sido projetada para isso desde o início.

Para cargas de trabalho com muita gravação, uma região primária deve ser selecionada e a capacidade de fazer failover para uma região em espera deve ser incorporada à carga de trabalho. Em comparação com uma abordagem ativa/ativa, uma abordagem primária/em espera é menos complicada. Isso ocorre porque, para uma arquitetura ativa/ativa, a carga de trabalho precisará ser reescrita para lidar com o roteamento inteligente para regiões, estabelecer afinidade de sessão, garantir transações idempotentes e lidar com possíveis conflitos.

A maioria das cargas de trabalho que buscam resiliência em várias regiões não exigirá uma abordagem ativa/ativa. Uma estratégia de fragmentação pode ser usada para fornecer maior resiliência, limitando o raio de explosão de uma deficiência em toda a base de clientes. Se você puder fragmentar efetivamente uma base de clientes, diferentes regiões primárias poderão ser selecionadas para cada fragmento. Por exemplo, se você puder fragmentar clientes de forma que metade dos clientes esteja alinhada à Região Um e a outra metade à Região Dois, tratando as regiões como células, uma abordagem celular multirregional pode ser criada, o que resulta na redução do raio de impacto de sua carga de trabalho.

A abordagem de fragmentação pode ser combinada com uma abordagem primária/em espera para fornecer recursos de failover para os fragmentos. Um processo de failover testado precisará ser incorporado à carga de trabalho e um processo de reconciliação de dados também precisará ser

projetado para garantir a consistência transacional dos armazenamentos de dados após o failover. Eles serão abordados com mais detalhes posteriormente neste paper.

Orientação-chave

- Há uma grande probabilidade de que as gravações pendentes para replicação não sejam confirmadas na região de espera quando houver uma falha. Os dados ficarão indisponíveis até que a replicação seja retomada (supondo a replicação assíncrona).
- Como parte do failover, será necessário um processo de reconciliação de dados para garantir que um estado transacionalmente consistente seja mantido nos armazenamentos de dados usando a replicação assíncrona.
- Quando for necessária uma consistência forte, as cargas de trabalho precisarão ser modificadas para tolerar a latência exigida do armazenamento de dados que se replica de forma síncrona.

Princípio fundamental da multirregião 3: Entender suas dependências de carga de trabalho

Uma carga de trabalho específica pode ter várias dependências em uma região, como AWS serviços usados, dependências internas, dependências de terceiros, dependências de rede, certificados, chaves, segredos e parâmetros. Para garantir a operação da carga de trabalho durante um cenário de falha, não deve haver dependências entre a região principal e a região de espera; cada uma deve ser capaz de operar independentemente uma da outra. Para conseguir isso, todas as dependências na carga de trabalho devem ser examinadas para garantir que estejam disponíveis em cada região. Isso é necessário porque uma falha na região primária não deve ter um impacto na região de espera. Além disso, o conhecimento de como a carga de trabalho opera quando uma dependência está em um estado degradado ou completamente indisponível é fundamental, para que as soluções possam ser projetadas para lidar com isso de forma adequada.

3a: serviços AWS

Ao projetar uma arquitetura multirregional, é necessário entender AWS os serviços específicos que serão usados. O primeiro aspecto é entender quais recursos o serviço tem para habilitar a multirregião e se uma solução deve ser projetada para atingir as metas multirregionais. Por exemplo, com o Amazon Aurora e o Amazon DynamoDB, há um recurso para replicar dados de forma assíncrona para uma região em espera. Todas as dependências de AWS serviço precisarão estar disponíveis em todas as regiões nas quais uma carga de trabalho será executada. Para garantir que os serviços que serão usados estejam disponíveis nas regiões desejadas, consulte a Lista de Região da AWS todos os serviços.

3b: Dependências internas e de terceiros

Para quaisquer dependências internas que uma carga de trabalho tenha, certifique-se de que ela esteja disponível nas regiões nas quais a carga de trabalho operará. Por exemplo, se a carga de trabalho for composta por muitos microsserviços, conheça todos os microsserviços que compõem uma capacidade de negócios. A partir daí, garanta que todos esses microsserviços sejam implantados em cada região na qual a carga de trabalho operará.

Chamadas entre regiões entre microsserviços dentro de uma carga de trabalho não são recomendadas, e o isolamento regional deve ser mantido. Isso ocorre porque a criação de dependências entre regiões aumenta o risco de falhas correlacionadas, o que anula os benefícios

que você está tentando obter com implementações regionais isoladas da carga de trabalho. As dependências locais também podem fazer parte da carga de trabalho, portanto, é fundamental entender como as características dessas integrações poderiam mudar se a região principal mudasse. Por exemplo, se a região de espera estiver localizada mais longe do ambiente local, o aumento da latência terá um impacto negativo.

Compreender as soluções de software como serviço (SaaS), os kits de desenvolvimento de software (SDKs) e outras dependências de produtos de terceiros e a capacidade de testar cenários em que essas dependências estão degradadas ou indisponíveis fornecerá mais informações sobre como a cadeia de sistemas opera e se comporta em diferentes modos de falha. Essas dependências podem estar em um código de aplicativo, desde como os segredos são gerenciados externamente usando o AWS Secrets Manager ou uma solução de cofre de terceiros (como a Hashicorp) até sistemas de autenticação que dependem do IAM Identity Center para logins federados.

Ter redundância quando se trata de dependências pode ajudar a aumentar a resiliência. Também existe a possibilidade de que uma solução SaaS ou dependência de terceiros esteja usando o mesmo primário Região da AWS da carga de trabalho. Se for esse o caso, você deve trabalhar com o fornecedor para determinar se a postura de resiliência corresponde aos requisitos da carga de trabalho.

Além disso, esteja ciente do destino compartilhado entre a carga de trabalho e suas dependências, como aplicativos de terceiros. Se as dependências não estiverem disponíveis em (ou de) uma região secundária após um failover, a carga de trabalho pode não se recuperar totalmente.

3c: Mecanismo de failover

O Sistema de Nomes de Domínio (DNS) é comumente usado como um mecanismo de failover para transferir o tráfego da região primária para uma região de espera. Analise e examine criticamente todas as dependências que o mecanismo de failover assume. Por exemplo, se sua carga de trabalho estiver usando o Amazon Route 53, entender que o plano de controle está hospedado em US-East-1 significa que você está se tornando dependente do plano de controle nessa região específica. Isso não é recomendado como parte de um mecanismo de failover se a região principal também for US-East-1. Se outro mecanismo de failover estiver sendo usado, é necessário um profundo entendimento de qualquer cenário em que ele não funcione conforme o esperado. Uma vez estabelecido esse entendimento, planeje a contingência ou desenvolva um novo mecanismo, se necessário. Analise a criação de mecanismos de recuperação de desastres usando o Amazon Route 53 para saber mais sobre as abordagens que você pode usar para realizar um failover bem-sucedido.

Conforme discutido na seção de dependências internas, todos os microsserviços que fazem parte de um recurso de negócios precisam estar disponíveis em cada região na qual a carga de trabalho é implantada. Como parte da estratégia de failover, a capacidade comercial precisa fazer o failover em conjunto para eliminar a chance de chamadas entre regiões. Como alternativa, se os microsserviços fizerem failover de forma independente, isso introduz o potencial de comportamento indesejável em que os microsserviços potencialmente fazem chamadas entre regiões, o que introduz latência e pode fazer com que a carga de trabalho fique indisponível no caso de tempo limite do cliente.

3d: dependências de configuração

Certificados, chaves, segredos e parâmetros fazem parte da análise de dependência necessária ao projetar para várias regiões. Sempre que possível, é melhor localizar esses componentes em cada região para que eles não tenham um destino compartilhado entre as regiões para essas dependências. Para certificados, a expiração deve variar entre eles e, se possível, em cada região, para evitar um cenário em que um certificado expirado (com alarmes configurados para notificação prévia) afete várias regiões.

As chaves e segredos de criptografia também devem ser específicos da região. Dessa forma, se houver um erro na rotação de uma chave ou segredo, o impacto será limitado a uma região específica.

Por fim, todos os parâmetros da carga de trabalho devem ser armazenados localmente para que a carga de trabalho seja recuperada na região específica.

Orientação-chave

- Uma arquitetura multirregional se beneficia da separação física e lógica entre regiões. A introdução de dependências entre regiões na camada de aplicação elimina esse benefício. Evite essas dependências.
- Os controles de failover devem funcionar sem dependências na região principal.
- A coordenação do failover na capacidade comercial precisa ser feita para eliminar a possibilidade de maior latência e dependência de chamadas entre regiões.

Fundamental multirregional 4: prontidão operacional

Operar uma carga de trabalho multirregional é uma tarefa complexa que vem com desafios operacionais específicos de várias regiões. Isso inclui Conta da AWS gerenciamento, processos de implantação reformulados, criação de uma estratégia de observabilidade em várias regiões, criação e teste de um runbook de failover e failback e, em seguida, gerenciar o custo. Uma análise de prontidão operacional (ORR) pode ajudar as equipes a preparar uma carga de trabalho para produção, seja em uma única região ou em várias regiões.

4a: gestão Conta da AWS

Para implantar uma carga de trabalho em todas as regiõesRegiões da AWS, garanta que todas as cotas de AWS serviço em uma conta estejam em paridade entre as regiões. Primeiro, conheça todos os AWS serviços que fazem parte da arquitetura, analise o uso planejado nas regiões em espera e compare-os com o uso atual. Em alguns casos, se a região em espera não tiver sido usada antes, você poderá consultar as cotas de serviço padrão para entender o ponto de partida. Em seguida, em todos os serviços que serão usados, solicite um aumento de cota usando o console de Cotas de Serviço (é necessário fazer login) ou as APIs.

AWSAs funções do Identity and Access Management (IAM) precisam ser configuradas em cada região para garantir que operadores, ferramentas de automação e AWS serviços tenham as permissões apropriadas para os recursos dentro da região em espera. O isolamento regional de funções alcança o isolamento regional que buscamos para arquiteturas multirregionais. Certifique-se de que essas permissões estejam em vigor antes de entrar em operação com uma região em espera.

4b: Práticas de implantação

Com recursos multirregionais, a implantação da carga de trabalho em várias regiões pode ser complexa. AWS CloudFormationajuda na implantação da infraestrutura em uma ou várias regiões e pode ser personalizada de acordo com suas necessidades. AWS CodePipelineajuda a fornecer um pipeline de integração/entrega contínua (CI/CD) quase contínuo, que tem ações entre regiões que permitem a implantação em regiões diferentes da região em que o pipeline está. Isso, combinado com estratégias robustas de implantação, como azul/verde, permite uma implantação de tempo de inatividade mínimo a zero.

No entanto, a implantação de recursos de monitoramento de estado pode ser mais complexa quando o estado do aplicativo ou dos dados não é externalizado para um armazenamento persistente. Nessas situações, adapte cuidadosamente o processo de implantação para atender às suas necessidades. Projete o pipeline e o processo de implantação para implantar em uma região por vez, em vez de várias regiões simultaneamente. Isso reduz a chance de falhas correlacionadas entre as regiões. Para saber mais sobre as técnicas que a Amazon usa para automatizar implantações de software, leia o artigo da Builder Library Automatizando implantações seguras e sem intervenção.

4c: Observabilidade

Ao projetar para várias regiões, considere como a saúde de todos os componentes em cada região será monitorada para obter uma visão holística da saúde regional. Isso pode incluir métricas de monitoramento para atraso na replicação, o que não é considerado para a carga de trabalho de uma única região.

Ao criar uma arquitetura multirregional, considere também observar o desempenho da carga de trabalho nas regiões em espera. Isso inclui fazer exames de saúde e canários (testes sintéticos) funcionando na região de espera, fornecendo uma visão externa da saúde da primária. Além disso, você pode usar o Amazon CloudWatch Internet Monitor para entender o estado da rede externa e o desempenho de suas cargas de trabalho do ponto de vista do usuário final. Da mesma forma, a região primária deve ter a mesma observabilidade para monitorar a região em espera. Esses canários devem monitorar as métricas de experiência do cliente para obter uma integridade geral da carga de trabalho. Isso é necessário porque, se houvesse um problema na região primária, a observabilidade na primária poderia ser prejudicada e afetaria a capacidade de avaliar a saúde da carga de trabalho.

Nesse caso, observar fora dessa região pode fornecer uma visão. Essas métricas devem ser agrupadas em painéis disponíveis em cada região e os alarmes criados em cada região. Como a Amazon CloudWatch é um serviço regional, é necessário tê-los em ambas as regiões. Esses dados de monitoramento serão usados para fazer a chamada para o failover de uma região primária para uma de espera.

4d: Processos, procedimentos e testes

O melhor momento para responder à pergunta “Quando devo fazer o failover?” é muito antes de você precisar. Os planos de continuidade de negócios, incluindo pessoas, processos e tecnologia, devem ser definidos com bastante antecedência e testados regularmente. Decida sobre uma estrutura de decisão de recuperação. Se houver um processo de recuperação bem praticado e o

tempo de recuperação for bem compreendido, é possível escolher o momento certo para iniciar o processo de recuperação que atenda à meta de RTO por meio de um failover. Esse momento pode ocorrer imediatamente após a identificação de um problema com o aplicativo na região principal ou pode ser um evento em que as opções de recuperação do aplicativo na região tenham sido esgotadas e agora seja necessário iniciar um failover para atender ao RTO.

Embora a ação de failover em si deva ser 100% automatizada, a decisão de ativar o failover deve ser tomada por uma pessoa (geralmente um pequeno número de indivíduos predeterminados na organização). Além disso, os critérios para decidir sobre um failover precisam ser claramente definidos e compreendidos globalmente pela organização. Esses processos podem ser definidos e concluídos usando os runbooks AWS do System Manager, que permitem a end-to-end automação completa e garantem a consistência da execução do processo durante o teste e o failover.

Esses runbooks devem estar disponíveis na região primária e de espera para iniciar os processos de failover ou failback. Depois que essa automação estiver implementada, uma cadência regular de testes deve ser definida e seguida. Isso garante que, quando houver um evento real, a resposta seja executada em um processo bem definido e praticado no qual a organização tenha confiança. Também é importante ter em mente as tolerâncias estabelecidas para os processos de reconciliação de dados. Confirme se os requisitos de RPO/RTO estabelecidos foram atendidos com o processo proposto.

4e: Custo e complexidade

As implicações de custo de uma arquitetura multirregional são impulsionadas pelo maior uso da infraestrutura, pela sobrecarga operacional e pelo tempo de recursos. Conforme mencionado anteriormente, o custo da infraestrutura em uma região em espera é semelhante ao custo da infraestrutura em uma região primária durante o pré-provisionamento, resultando em duas vezes o custo. Provisione a capacidade para que seja suficiente para as operações diárias, mas ainda reserve capacidade de buffer suficiente para tolerar picos na demanda — e configure os mesmos limites em cada região.

Além disso, podem ser necessárias alterações no nível do aplicativo para serem executadas com êxito em uma arquitetura multirregional se você estiver adotando uma arquitetura ativa-ativa, que pode exigir muito tempo e recursos para projetar e operar. No mínimo, as organizações precisariam gastar tempo entendendo as dependências técnicas e comerciais em cada região e projetando processos de failover e failback.

As equipes também devem realizar exercícios normais de failover e failback para se sentirem confortáveis com os runbooks que serão usados durante um evento. Embora sejam extremamente

importantes e cruciais para obter o resultado esperado de um investimento em várias regiões, esses exercícios representam um custo de oportunidade e retiram tempo e recursos de outras atividades.

Orientação-chave

- AWSas cotas de serviços precisam ser revisadas e em paridade em todas as regiões nas quais a carga de trabalho operará.
- O processo de implantação deve ter como alvo uma região por vez, em vez de várias regiões simultaneamente.
- Métricas adicionais, como atraso na replicação, precisam ser monitoradas e são específicas para cenários multirregionais.
- Estenda o monitoramento da carga de trabalho além da região principal. As métricas de experiência do cliente devem ser monitoradas por região e medidas fora de cada região em que a carga de trabalho está sendo executada.
- O failover e o failback precisam ser testados regularmente. Garanta a implementação de um único runbook para processos de failover e failback que seja usado durante o teste e em um evento ao vivo. Os runbooks para testes e eventos ao vivo não podem ser diferentes.

Conclusão

Este whitepaper discutiu casos de uso comuns para várias regiões, os fundamentos sobre como implementar uma arquitetura multirregional e as implicações dessa abordagem. Esses fundamentos podem ser aplicados a qualquer carga de trabalho e usados como uma estrutura para auxiliar na tomada de decisões sobre se uma arquitetura multirregional é ou não a abordagem correta para uma empresa específica.

Colaboradores

Os colaboradores deste documento incluem:

Colaborador técnico:

- John Formento, Jr., arquiteto principal de soluções, equipe AWS multirregional

Colaborador editorial:

- Lisi Lewis, gerente sênior de marketing de produtos

Outras fontes de leitura

Para obter informações adicionais, consulte:

- Padrões avançados de resiliência Multi-AZ (white paper) AWS
- Pilar de confiabilidade - AWS Well-Architected Framework
- Disponibilidade e muito mais: entendendo e melhorando a resiliência de sistemas distribuídos em AWS (AWSwhitepaper)
- AWSLimites de isolamento de falhas (AWSwhite paper)

Revisões do documento

Para ser notificado sobre atualizações desse whitepaper, inscreva-se no feed RSS.

Alteração	Descrição	Data
Documento publicado	Primeira publicação.	20 de dezembro de 2022

Avisos

Os clientes são responsáveis por fazer a própria avaliação independente das informações contidas neste documento. Este documento: (a) é apenas para fins informativos, (b) representa as ofertas e práticas de produtos atuais da AWS, que estão sujeitas a alterações sem aviso prévio e (c) não criam nenhum compromisso ou garantia da AWS e de suas afiliadas, fornecedores ou licenciadores. Os produtos ou serviços da AWS são fornecidos “no estado em que se encontram”, sem garantias, representações ou condições de qualquer tipo, expressas ou implícitas. As responsabilidades e as obrigações da AWS com os seus clientes são controladas por contratos da AWS, e este documento não é parte, nem modifica, qualquer contrato entre a AWS e seus clientes.

AWS Glossário

Para obter a terminologia mais recente da AWS, consulte o glossário da AWS na Referência do Glossário da AWS.

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.