AWS 백서

AWS 멀티 리전 기초

AWS 멀티 리전 기초: AWS 백서

Table of Contents

AWS 멀티 리전 기초

발행일: 2022년 12월 20일 () 문서 수정

요약

이 300레벨 고급 백서는 다중 지역 아키텍처를 사용하여 워크로드의 탄력성을 개선하려는 워크로드를 구축하는 클라우드 아키텍트와 고위 리더를 대상으로 합니다. AWS 이 백서에서는 AWS 인프라 및 서비스에 대한 기본 지식을 전제로 합니다. 일반적인 다중 지역 사용 사례를 설명하고, 설계, 개발 및 배포와 관련된 기본적인 다중 지역 개념과 의미를 공유하고, 다중 지역 아키텍처가 워크로드에 적합한지 여부를 더 잘 판단하는 데 도움이 되는 규범적 지침을 제공합니다.

귀사는 Well-Architected입니까?

AWS Well-Architected 프레임워크는 클라우드에서 시스템을 구축할 때 내리는 결정의 장단점을 이해하는 데 도움이 됩니다. 이 프레임워크를 사용하여 클라우드에서 안정적이고 안전하며 효율적이고 비용 효율적인 시스템을 설계하고 운영하기 위한 아키텍처 모범 사례를 살펴볼 수 있습니다. AWS Management Console에서 무료로 제공되는 AWS Well-Architected Tool을 사용하면 각 요소에 대한 일련의 질문에 답하여 모범 사례와 비교하여 워크로드를 검토할 수 있습니다.

참조 아키텍처 배포, 다이어그램, 백서 등 클라우드 아키텍처에 대한 더 많은 전문가 지침과 모범 사례를 보려면 AWS 아키텍처 센터를 참조하세요.

소개

각 가용 영역은 지리적 영역 내에 독립적이고 물리적으로 분리된 여러 개의 가용 영역으로 AWS 리전구성되어 있습니다. 각 지역의 소프트웨어 서비스 간에는 엄격한 논리적 분리가 유지됩니다. 이러한 용도에 맞는 설계를 통해 한 지역의 인프라 또는 서비스 장애가 다른 지역의 상관 관계 장애로 이어지지 않도록 할 수 있습니다.

대부분의 AWS 고객은 다중 가용 영역 (AZ) 또는 지역 서비스를 사용하여 단일 지역의 워크로드에 대한 복원력 목표를 달성할 수 있습니다. AWS 그러나 일부 고객은 다음과 같은 세 가지 이유로 다중 지역 아키텍처를 선호합니다.

- 이들은 최상위 계층 워크로드에 대한 높은 가용성과 운영 연속성 요구 사항을 가지고 있으며, 이러한 요구 사항은 단일 지역으로는 충족할 수 없다고 생각합니다.

- 워크로드가 특정 관할권 내에서 운영되도록 요구하는 데이터 주권 요구 사항 (예: 현지 법률, 규정 및 규정 준수) 을 충족해야 합니다.
- 최종 사용자와 가장 가까운 위치에서 워크로드를 실행하여 워크로드의 성능과 고객 경험을 개선해야 합니다.

이 백서는 운영 요구 사항의 고가용성 및 연속성 요구 사항에 초점을 맞추고 워크로드에 다중 지역 아키텍처를 채택하기 위한 고려 사항을 탐색하는 데 도움이 됩니다. 다중 지역 워크로드의 설계, 개발 및 배포에 적용되는 기본 개념과 함께 다중 지역 아키텍처가 특정 워크로드에 적합한 선택인지 판단하는 데 도움이 되는 규범적 프레임워크를 설명합니다. 다중 지역 아키텍처는 까다롭고 올바르게 구성하지 않으면 워크로드의 전체 가용성이 저하될 수 있으므로 다중 지역 아키텍처가 워크로드에 적합한 선택인지 확인해야 합니다.

단일 지역의 탄력성을 위한 엔지니어링 및 운영

다중 리전 개념을 자세히 살펴보기 전에 먼저 단일 리전에서 워크로드의 복원력이 이미 최대한 높은지 확인하십시오. 이를 위해서는 AWS Well-Architected Framework의 안정성 요소 및 운영 우수성 기둥을 기준으로 워크로드를 평가하고 권장 모범 사례를 채택하는 데 필요한 사항을 변경하십시오. AWSWell-Architected 프레임워크에서는 다음과 같은 개념을 다룹니다.

- 도메인 경계에 따른 워크로드 세분화
- 잘 정의된 서비스 계약
- 종속성 관리 및 결합
- 장애 처리, 재시도 및 백오프 전략
- 무력한 운영 및 스테이트풀 트랜잭션과 스테이트리스 트랜잭션
- 운영 준비 및 변경 관리
- 워크로드 상태 이해
- 이벤트 대응

단일 지역 복원력을 한 단계 더 발전시키려면 고급 다중 AZ 복구 패턴에서 설명하는 그레이 장애 처리를 위한 개념을 검토하고 적용하십시오. 이 백서에서는 장애를 억제하기 위해 각 가용 영역의 복제본을 사용하는 모범 사례를 제공하고 Well Architected에 도입된 다중 AZ 개념을 확장합니다. AWS 단일 지역에서 최고의 복원력을 달성하기 위한 권장 개념과 모범 사례를 완전히 적용한 후에는 다중 지역 아키텍처의 기본 사항을 기준으로 특정 워크로드를 평가하여 다중 지역 접근 방식을 사용하여 워크로드의 복원력을 높일 수 있는지 판단할 수 있습니다.

멀티 리전 기초 1: 요구 사항 이해

앞서 언급했듯이 다중 지역 아키텍처를 추구하는 일반적인 이유는 고가용성과 운영 연속성입니다. 가용성 지표는 정의된 기간 동안 워크로드를 사용할 수 있는 시간의 비율을 측정하는 반면, 운영 연속성 지표는 대규모 및 일반적으로 더 긴 기간이 소요되는 이벤트의 복구를 측정합니다.

가용성 측정은 거의 연속적인 프로세스입니다. 특정 측정값이나 지표는 다양할 수 있지만, 일반적으로 목표 가용성 (예: 99.99% 가용성) 을 중심으로 통합됩니다. 가용성 목표를 세울 때는 한 가지 방법이 모든 상황에 맞지는 않습니다. 가용성 목표는 모든 워크로드에 단일 목표를 적용하는 것보다는, 중요하지 않은 구성 요소를 중요한 구성 요소와 분리하여 워크로드 수준에서 설정해야 합니다.

운영 연속성을 위해 일반적으로 다음과 같은 측정이 사용됩니다. point-in-time

- 복구 시간 목표 (RTO) — RTO는 서비스 중단과 서비스 복원 사이의 허용 가능한 최대 지연입니다. 이 값에 따라 서비스가 손상될 수 있는 허용 기간이 결정됩니다.
- 복구 시점 목표 (RPO) - RPO는 마지막 데이터 복구 시점 이후 허용되는 최대 시간입니다. 이는 최근 복구 시점과 서비스 중단 사이에 허용되는 데이터 손실로 간주되는 범위를 결정합니다.

가용성 목표 설정과 마찬가지로 RTO와 RPO도 워크로드 수준에서 정의해야 합니다. 보다 엄격한 운영 연속성 또는 고가용성 요구 사항을 달성하려면 투자를 늘려야 합니다. 하지만 모든 애플리케이션이 동일한 수준의 복원력을 요구하거나 필요로 할 수는 없습니다. 계층화 메커니즘을 만들면 비즈니스 및 IT 소유자가 비즈니스에 미치는 영향을 기반으로 가장 까다로운 애플리케이션을 식별하고 그에 따라 계층화하는 프레임워크를 구축하는 데 도움이 될 수 있습니다. 계층화의 예는 다음 표에서 확인할 수 있습니다.

표 1 — SLA의 레질리언스 계층화 예시

가용성 서비스 수준 계약 (SLA)	레질리언스 티어	연간 허용 가능한 다운타임
99.99%	플래티넘	52.60분
99.90%	골드	8.77시간
99.5%	실버	1.83일

표 2 — RTO 및 RPO에 대한 레질리언스 계층화 예시

티어	최대 RTO	최대 RPO	기준	비용
플래티넘	15분	5분	업무상 중요한 워크로드	$$$
골드	15분 — 6시간	두 시간	중요하지만 업무상 중요한 워크로드는 아님	$$
실버	6시간 — 며칠	24시간	중요하지 않은 워크로드	$

탄력성을 위한 워크로드를 설계할 때는 고가용성과 운영 연속성 간의 관계를 이해하는 것이 필요합니다. 예를 들어 99.99% 의 가용성이 필요한 워크로드의 경우 연간 다운타임은 53분을 넘지 않아야 합니다. 장애를 감지하는 데 최소 5분이 걸리고 운영자가 개입하여 복구 단계를 결정하고 이러한 단계를 수행하는 데 10분이 더 걸릴 수 있습니다. 단일 문제를 복구하는 데 30~45분이 걸리는 것은 드문 일이 아닙니다. 이 경우 상호 관련된 영향을 제거하는 격리된 인스턴스를 제공하는 다중 지역 전략을 수립하면 제한된 시간 내에 페일오버하는 동시에 초기 장애를 개별적으로 분류하여 운영을 지속할 수 있습니다. 이때 적절한 RTO와 RPO를 정의해야 합니다.

가용성이 매우 높거나 (예: 99.99% 가용성 이상) 또는 다른 지역으로 페일오버해야만 충족할 수 있는 엄격한 운영 연속성 요구 사항이 있는 미션 크리티컬 워크로드의 경우 다중 지역 접근 방식이 적절할 수 있습니다. 그러나 이러한 요구 사항은 일반적으로 복구 시간이 분 또는 시간 단위로 제한되는 엔터프라이즈 워크로드 포트폴리오 중 일부에 대해서만 적용됩니다. 애플리케이션에 몇 분 또는 몇 시간의 복구 시간이 필요한 경우가 아니라면 애플리케이션의 지역적 중단이 영향을 받는 지역 내에서 해결될 때까지 기다리는 것이 더 나은 접근 방식일 수 있으며 일반적으로 하위 계층 워크로드에 맞춰 조정됩니다.

다중 지역 아키텍처를 구현하기 전에 비즈니스 의사 결정권자와 기술 팀은 운영 및 인프라 비용 동인을 포함하여 비용에 미치는 영향에 대해 의견을 조율해야 합니다. 일반적인 다중 지역 아키텍처는 단일 지역 접근 방식에 비해 비용이 2배 증가할 수 있습니다. 핫 스탠바이, 웜 스탠바이, 파일럿 라이트를 사용하여 실행하는 등 비즈니스 연속성을 위한 다중 지역 패턴이 여러 개 있지만, 복구 목표를 달성할 위험이 가장 낮은 패턴에는 핫 스탠바이 실행이 포함되므로 워크로드 비용이 두 배로 늘어납니다.

주요 지침

- RTO 및 RPO 같은 운영 목표의 가용성 및 연속성을 워크로드별로 설정하고 비즈니스 및 IT 이해 관계자와 연계해야 합니다.
- 단일 지역 내에서 대부분의 가용성 및 운영 지속성 목표를 달성할 수 있습니다. 단일 지역으로는 달성할 수 없는 목표의 경우 비용, 복잡성, 이점 간의 균형을 명확히 파악하여 다중 지역을 고려해야 합니다.

멀티 리전 기초 2: 데이터 이해

멀티 리전 아키텍처에서는 데이터 관리가 쉽지 않은 문제입니다. 지역 간 지리적 거리로 인해 불가피한 지연 시간이 발생하며, 이는 지역 간에 데이터를 복제하는 데 걸리는 시간으로 나타납니다. 다중 지역 아키텍처를 사용하는 워크로드의 가용성, 데이터 일관성, 그리고 더 높은 지연 시간 사이의 균형을 맞춰야 합니다. 비동기 복제를 사용하든 동기 복제를 사용하든 복제 기술로 인한 동작 변화를 처리하려면 애플리케이션을 수정해야 합니다. 단일 지역용으로 설계된 기존 애플리케이션을 다중 지역으로 만드는 것은 데이터 일관성과 지연 시간 문제로 인해 매우 어렵습니다. 특정 워크로드에 대한 데이터 일관성 요구 사항과 데이터 액세스 패턴을 이해하는 것은 장단점을 평가하는 데 매우 중요합니다.

2a: 데이터 일관성 요구 사항 이해

CAP 정리는 데이터 일관성, 가용성 및 네트워크 파티션 간의 장단점을 추론하기 위한 참조를 제공하며, 워크로드에 대해 동시에 두 개만 충족할 수 있습니다. 다중 지역은 정의상 지역 간 네트워크 파티션을 포함하므로 가용성과 일관성 중에서 선택해야 합니다.

지역 간 데이터 가용성을 선택하면 커밋된 데이터를 지역 간에 비동기식으로 복제해야 하므로 복제가 완료될 때까지 지역 간 일관성이 저하되므로 트랜잭션 쓰기 중에 지연 시간이 크게 발생하지 않습니다. 비동기식 복제를 사용하면 기본 지역에 장애가 발생할 경우 기본 지역에서 쓰기 작업이 복제 보류 중일 가능성이 높습니다. 이로 인해 복제가 재개될 때까지 최신 데이터를 사용할 수 없는 시나리오가 발생하며, 중단이 발생한 지역에서 복제되지 않은 진행 중인 트랜잭션을 처리하려면 조정 프로세스가 필요합니다.

비동기식 복제가 선호되는 워크로드의 경우 비동기식 교차 리전 복제를 제공하는 Amazon Aurora 및 Amazon DynamoDB와 같은 서비스를 사용할 수 있습니다. Amazon Aurora 글로벌 데이터베이스와 Amazon DynamoDB 글로벌 테이블 모두 복제 지연을 모니터링하는 데 도움이 되는 기본 CloudWatchAmazon 지표가 있습니다.

이벤트 기반 아키텍처를 활용하도록 워크로드를 설계하면 워크로드가 데이터의 비동기 복제를 수용할 수 있고 이벤트를 재생하여 상태를 재구성할 수 있기 때문에 다중 리전 전략의 이점이 됩니다. 스트리밍 및 메시징 서비스는 메시지 페이로드 데이터를 단일 지역에 버퍼링하므로 지역별 장애 조치/장애 복구 프로세스에는 클라이언트 입력 데이터 흐름을 리디렉션하고 중단이 발생한 지역에 저장된 전송 중 및/또는 전달되지 않은 페이로드를 조정하는 메커니즘이 포함되어야 합니다.

일관성을 선택하면 트랜잭션 쓰기 중에 데이터가 동기적으로 복제되므로 상당한 지연 시간이 발생합니다. 여러 지역에 동기적으로 쓸 때 모든 지역에서 쓰기가 성공하지 못하면 트랜잭션이 커밋되지 않아 다시 시도해야 하므로 가용성이 저하될 수 있습니다. 모든 지역에 데이터를 동기적으로 쓰려고 시도하는 재시도는 시도할 때마다 지연 시간을 감수해야 합니다. 재시도 횟수가 모두 소진되면 트랜잭션을 완

전히 실패시켜 가용성을 떨어뜨리거나 사용 가능한 지역에만 트랜잭션을 커밋하여 불일치를 초래할지 결정해야 합니다. 데이터를 동기식으로 복제하고 커밋하는 데 도움이 되는 Paxos와 같은 쿼럼 구성 기술이 있지만 이러한 기술에는 상당한 개발자 투자가 필요합니다.

강력한 일관성 요구 사항을 충족하기 위해 쓰기에 여러 지역에 걸친 동기 복제가 필요한 경우 쓰기 지연 시간이 10배 증가합니다. 일반적으로 쓰기 지연 시간이 길다고 해서 큰 변경 없이 애플리케이션에 맞게 조정할 수 있는 것은 아닙니다. 이상적으로는 애플리케이션을 처음 설계할 때 이 점을 고려해야 합니다. 동기 복제가 우선시되는 다중 지역 워크로드의 경우 AWS파트너 솔루션이 도움이 될 수 있습니다.

2b: 데이터 액세스 패턴 이해

워크로드 데이터 액세스 패턴은 읽기 집약형 또는 쓰기 집약형 유형 중 하나로 분류됩니다. 특정 워크로드의 이러한 특성을 이해하면 적절한 다중 지역 아키텍처를 선택하는 데 도움이 됩니다.

완전히 읽기 전용인 정적 콘텐츠와 같이 읽기 집약적인 워크로드의 경우 큰 복잡성 없이 액티브/액티브 멀티 리전 아키텍처를 구현할 수 있습니다. 콘텐츠 배포 네트워크 (CDN) 를 사용하여 엣지에서 정적 콘텐츠를 제공하면 최종 사용자와 가장 가까운 위치에 콘텐츠를 캐싱하여 가용성이 보장됩니다. Amazon 내에서 Origin 장애 조치와 같은 기능 세트를 사용하면 이를 달성하는 데 도움이 될 CloudFront 수 있습니다. 또 다른 옵션은 여러 지역에 스테이트리스 컴퓨팅을 배포하고 DNS를 사용하여 사용자를 가장 가까운 지역으로 라우팅하여 콘텐츠를 읽을 수 있도록 하는 것입니다. 지리적 위치 라우팅 정책이 있는 Route 53을 사용하여 이를 달성할 수 있습니다.

읽기 비율이 쓰기보다 큰 읽기 집약적 워크로드의 경우 로컬 읽기, 쓰기 글로벌 전략을 사용할 수 있습니다. 이를 위해서는 모든 쓰기가 특정 지역의 데이터베이스로 전송되고 다른 모든 지역에 데이터를 비동기로 복제해야 하며, 이를 위해서는 어느 지역에서든 읽기를 수행할 수 있습니다. 리전 간 쓰기 복제의 지연 시간이 길어져 로컬 읽기가 무용지물이 될 수 있으므로 이 접근 방식을 사용하려면 최종 일관성을 유지할 수 있는 워크로드가 필요합니다.

Aurora Global Database는 로컬에서 모든 읽기 트래픽만 처리할 수 있는 스탠바이 리전과 쓰기를 처리할 특정 리전의 단일 기본 데이터스토어에 읽기 전용 복제본을 프로비저닝하는 데 도움이 될 수 있습니다. 데이터는 기본 데이터베이스에서 대기 데이터베이스 (읽기 전용 복제본) 로 비동기적으로 복제되며, 대기 지역으로 작업을 페일오버해야 하는 경우 대기 데이터베이스를 기본 데이터베이스로 승격할 수 있습니다. 워크로드가 비관계형 데이터 모델에 더 적합하다면 DynamoDB를 이 접근 방식에서도 사용할 수 있습니다. 다시 말하지만, 워크로드는 최종 일관성을 수용해야 합니다. 처음부터 이를 위해 설계되지 않았다면 워크로드를 다시 작성해야 할 수도 있습니다.

쓰기 집약적인 워크로드의 경우 기본 지역을 선택하고 대기 지역으로 페일오버하는 기능을 워크로드에 엔지니어링해야 합니다. 액티브/액티브 접근 방식에 비해 기본/대기 접근 방식은 덜 복잡합니다. 액

티브/액티브 아키텍처의 경우 지역으로의 지능형 라우팅을 처리하고, 세션 친화성을 설정하고, 불능성 트랜잭션을 보장하고, 잠재적 충돌을 처리하려면 워크로드를 다시 작성해야 하기 때문입니다.

복원력을 위해 다중 지역을 찾는 대부분의 워크로드에는 능동적/능동적 접근 방식이 필요하지 않습니다. 샤딩 전략을 사용하면 클라이언트 기반 전반에서 손상의 폭발 범위를 제한하여 복원력을 높일 수 있습니다. 고객 기반을 효과적으로 샤드할 수 있다면 샤드별로 다른 기본 지역을 선택할 수 있습니다. 예를 들어 클라이언트의 절반은 지역 1에 정렬되고 절반은 지역 2에 정렬되도록 클라이언트를 분할하여 지역을 셀로 취급하면 다중 지역 셀 접근 방식을 만들어 워크로드에 미치는 영향을 줄일 수 있습니다.

샤딩 접근 방식을 기본/스탠바이 접근 방식과 결합하여 샤드에 페일오버 기능을 제공할 수 있습니다. 테스트를 거친 페일오버 프로세스를 워크로드에 맞게 엔지니어링해야 하며, 페일오버 후 데이터 저장소의 트랜잭션 일관성을 보장하려면 데이터 조정 프로세스도 엔지니어링해야 합니다. 이에 대해서는 이 백서의 뒷부분에서 더 자세히 다룹니다.

주요 지침

- 장애 발생 시 복제를 위해 보류 중인 쓰기가 스탠바이 리전에 커밋되지 않을 가능성이 높습니다. 복제가 재개될 때까지 데이터를 사용할 수 없습니다 (비동기 복제 가정).
- 페일오버의 일환으로 비동기 복제를 사용하는 데이터스토어의 트랜잭션 일관성을 유지하기 위한 데이터 조정 프로세스가 필요합니다.
- 강력한 일관성이 필요한 경우 동기식으로 복제하는 데이터스토어에 필요한 지연 시간을 허용하도록 워크로드를 수정해야 합니다.

멀티 리전 기초 3: 워크로드 종속성 이해

특정 워크로드에는 사용된 AWS 서비스, 내부 종속성, 타사 종속성, 네트워크 종속성, 인증서, 키, 암호 및 매개 변수와 같은 지역 내 여러 종속성이 있을 수 있습니다. 장애 시나리오에서 워크로드가 제대로 작동하려면 기본 리전과 스탠바이 리전 간에 종속성이 없어야 합니다. 각 리전은 서로 독립적으로 작동할 수 있어야 합니다. 이를 위해서는 워크로드의 모든 종속성을 면밀히 조사하여 각 지역 내에서 사용할 수 있는지 확인해야 합니다. 이는 기본 리전에 장애가 발생해도 스탠바이 리전에 영향을 미치지 않아야 하기 때문에 필요합니다. 또한 종속성이 저하된 상태이거나 완전히 사용할 수 없을 때 워크로드가 어떻게 작동하는지에 대한 지식도 필수적이므로 솔루션을 설계하여 이를 적절하게 처리할 수 있도록 해야 합니다.

3a: 서비스 AWS

다중 지역 아키텍처를 설계할 때는 사용될 특정 AWS 서비스를 이해하는 것이 필요합니다. 첫 번째 측면은 멀티 리전을 지원하는 데 필요한 서비스의 기능과 멀티 리전 목표를 달성하기 위해 솔루션을 설계해야 하는지를 이해하는 것입니다. 예를 들어 Amazon Aurora와 Amazon DynamoDB의 경우 데이터를 스탠바이 리전에 비동기적으로 복제하는 기능이 있습니다. 모든 AWS 서비스 종속성은 워크로드가 실행될 모든 지역에서 사용할 수 있어야 합니다. 사용할 서비스를 원하는 지역에서 사용할 수 있는지 확인하려면 AWS 리전al Services List를 검토하세요.

3b: 내부 및 타사 종속성

워크로드에 있는 내부 종속성의 경우 워크로드가 운영되는 지역 이외의 지역에서도 사용할 수 있는지 확인하세요. 예를 들어 워크로드가 여러 마이크로서비스로 구성된 경우 비즈니스 기능을 구성하는 모든 마이크로서비스에 대해 잘 알고 있어야 합니다. 거기에서 해당 마이크로서비스를 모두 워크로드가 운영되는 각 지역에 배포해야 합니다.

워크로드 내 마이크로서비스 간 지역 간 통화는 권장되지 않으며 지역별 격리를 유지해야 합니다. 지역 간 종속성을 만들면 상호 관련된 장애가 발생할 위험이 높아져 워크로드의 격리된 지역 구현으로 달성하려는 이점이 무효화되기 때문입니다. 온프레미스 종속성도 워크로드의 일부일 수 있으므로 기본 지역이 변경될 경우 이러한 통합의 특성이 어떻게 변경될 수 있는지 이해하는 것이 필수적입니다. 예를 들어 스탠바이 리전이 온프레미스 환경에서 더 멀리 떨어져 있는 경우 늘어난 지연 시간은 부정적인 영향을 미칩니다.

SaaS (Software as a Service) 솔루션, 소프트웨어 개발 키트 (SDK) 및 기타 타사 제품 종속성을 이해하고 이러한 종속성이 저하되거나 사용할 수 없는 시나리오를 실행할 수 있으면 다양한 장애 모드에서 시스템 체인이 어떻게 작동하고 작동하는지 더 잘 이해할 수 있습니다. 이러한 종속성은 AWS Secrets

Manager 또는 타사 볼트 솔루션 (예: Hashicorp) 을 사용하여 외부에서 비밀을 관리하는 방법부터 페더레이션 로그인을 위해 IAM Identity Center에 종속되는 인증 시스템에 이르기까지 애플리케이션 코드 내에 있을 수 있습니다.

종속성과 관련하여 중복성을 확보하면 복원력을 높이는 데 도움이 될 수 있습니다. SaaS 솔루션 또는 타사 종속성이 동일한 기본 AWS 리전 워크로드를 사용하고 있을 가능성도 있습니다. 이 경우 공급업체와 협력하여 해당 공급업체의 복원력 상태가 워크로드 요구 사항과 일치하는지 확인해야 합니다.

또한 워크로드와 해당 종속성 (예: 타사 애플리케이션) 간의 운명이 공유된다는 점도 염두에 두세요. 페일오버 후 보조 지역에서 종속성을 사용할 수 없는 경우 워크로드가 완전히 복구되지 않을 수 있습니다.

3c: 페일오버 메커니즘

도메인 이름 시스템 (DNS) 은 트래픽을 기본 지역에서 대기 지역으로 이동하는 장애 조치 메커니즘으로 일반적으로 사용됩니다. 페일오버 메커니즘이 취하는 모든 종속성을 비판적으로 검토하고 면밀히 조사하십시오. 예를 들어, 워크로드가 Amazon Route 53을 사용하는 경우 제어 플레인이 미국 동부-1에 호스팅된다는 점을 이해하면 해당 지역의 컨트롤 플레인에 대한 종속성을 갖게 됩니다. 기본 지역이 미국 동부 지역이기도 한 경우에는 장애 조치 메커니즘의 일부로 사용하지 않는 것이 좋습니다. 다른 페일오버 메커니즘을 사용하는 경우 예상대로 작동하지 않는 시나리오를 깊이 이해해야 합니다. 이러한 이해가 확립되면 비상 사태에 대비한 계획을 세우거나 필요한 경우 새로운 메커니즘을 개발하십시오. Amazon Route 53을 사용한 재해 복구 메커니즘 생성을 검토하여 장애 조치를 성공적으로 수행하는 데 사용할 수 있는 접근 방식에 대해 알아보십시오.

내부 종속성 섹션에서 설명한 바와 같이, 비즈니스 역량의 일부인 모든 마이크로서비스는 워크로드가 배포되는 각 지역에서 사용할 수 있어야 합니다. 장애 조치 전략의 일환으로 비즈니스 기능을 함께 페일오버하여 지역 간 통화가 발생할 가능성을 없애야 합니다. 또는 마이크로서비스가 독립적으로 페일오버되는 경우 마이크로서비스가 잠재적으로 지역 간 호출을 수행하는 바람직하지 않은 동작이 발생할 수 있으며, 이로 인해 지연 시간이 발생하고 클라이언트 시간 초과 시 워크로드를 사용할 수 없게 될 수 있습니다.

3d: 구성 종속성

인증서, 키, 암호 및 매개변수는 다중 지역용으로 설계할 때 필요한 종속성 분석의 일부입니다. 가능하면 이러한 구성 요소를 각 지역 내에서 현지화하여 이러한 종속성에 대해 지역 간에 운명이 공유되지 않도록 하는 것이 가장 좋습니다. 인증서의 경우 만료일은 각 지역마다 달라야 하며, 만료되는 인증서 (미리 알리도록 경보가 설정되어 있음) 가 여러 지역에 영향을 미치는 시나리오를 피하려면 가능하면 각 지역마다 만료일이 달라야 합니다.

암호화 키와 비밀도 지역별로 달라야 합니다. 이렇게 하면 키나 암호의 로테이션에 오류가 발생하는 경우 해당 영향이 특정 지역에만 제한됩니다.

마지막으로, 워크로드가 특정 지역에서 검색할 수 있도록 모든 워크로드 파라미터를 로컬에 저장해야 합니다.

주요 지침

- 다중 지역 아키텍처는 지역 간의 물리적 및 논리적 분리를 통해 이점을 얻습니다. 애플리케이션 계층에 지역 간 종속성을 도입하면 이러한 이점이 사라집니다. 이러한 종속성을 피하십시오.
- 장애 조치 제어는 기본 지역에 대한 종속성 없이 작동해야 합니다.
- 지역 간 통화의 지연 시간 및 종속성 증가 가능성을 없애려면 비즈니스 역량에 맞게 장애 조치를 조정해야 합니다.

멀티 리전 기본 4: 운영 준비

다중 지역 워크로드를 운영하는 것은 복잡한 작업이며 다중 지역에서만 발생하는 운영상의 문제가 수반됩니다. 여기에는 AWS 계정 관리, 재편된 배포 프로세스, 다중 지역 관찰 가능성 전략 수립, 페일오버 및 페일백 런북 생성 및 테스트, 비용 관리 등이 포함됩니다. 운영 준비 검토 (ORR) 는 팀이 단일 지역에서 실행하든 여러 지역에서 실행하든 관계없이 프로덕션에 사용할 워크로드를 준비하는 데 도움이 될 수 있습니다.

4a: 관리 AWS 계정

워크로드를 여러 지역에 배포하려면 계정 내 모든 AWS서비스 할당량이 지역 간에 AWS 리전 동일하게 적용되도록 해야 합니다. 먼저 아키텍처의 일부인 모든 AWS 서비스를 알아보고 대기 지역의 계획된 사용량을 살펴본 다음 현재 사용량과 비교하십시오. 경우에 따라 이전에 스탠바이 리전을 사용한 적이 없는 경우 기본 서비스 할당량을 참조하여 시작점을 이해할 수 있습니다. 그런 다음 사용할 모든 서비스에 대해 Service Quotas 콘솔 (로그인 필요) 또는 API를 사용하여 할당량 증가를 요청하세요.

AWS운영자, 자동화 도구 및 AWS 서비스가 대기 지역 내의 리소스에 대한 적절한 권한을 갖도록 하려면 각 지역에서 Identity and Access Management (IAM) 역할을 구성해야 합니다. 역할을 지역적으로 격리하면 다중 지역 아키텍처에서 우리가 추구하는 지역적 격리가 달성됩니다. 스탠바이 리전을 활성화하기 전에 이러한 권한이 제대로 적용되었는지 확인하세요.

4b: 배포 사례

다중 지역 기능을 사용하면 여러 지역에 워크로드를 배포하는 것이 복잡할 수 있습니다. AWS CloudFormation단일 또는 여러 지역에 인프라를 배포하는 데 도움이 되며 필요에 따라 조정할 수 있습니다. AWS CodePipeline거의 지속적인 통합/지속적 전달 (CI/CD) 파이프라인을 제공하는 데 도움이 됩니다. 이 파이프라인에는 파이프라인이 속한 지역과 다른 지역에 배포할 수 있는 지역 간 작업이 포함됩니다. 이를 블루/그린과 같은 강력한 배포 전략과 결합하면 다운타임이 전혀 없는 배포가 가능합니다.

그러나 애플리케이션 또는 데이터의 상태가 영구 저장소로 외부화되지 않는 경우 상태 저장 기능을 배포하는 것이 더 복잡할 수 있습니다. 이러한 상황에서는 필요에 맞게 배포 프로세스를 신중하게 조정하십시오. 여러 지역에 동시에 배포하는 대신 한 번에 한 지역에 배포하도록 배포 파이프라인과 프로세스를 설계하세요. 이렇게 하면 지역 간에 장애가 상호 연관될 가능성이 줄어듭니다. Amazon에서 소프트웨어 배포를 자동화하는 데 사용하는 기술에 대해 알아보려면 빌더 라이브러리 기사 안전한 수동 배포 자동화를 읽어보세요.

4c: 옵저버빌리티

여러 지역을 대상으로 설계할 때는 각 지역의 모든 구성 요소 상태를 모니터링하여 지역 건전성을 전체적으로 파악하는 방법을 고려하세요. 여기에는 단일 지역 워크로드에 대한 고려 사항이 아닌 복제 지연에 대한 모니터링 지표가 포함될 수 있습니다.

멀티 리전 아키텍처를 구축할 때는 스탠바이 리전의 워크로드 성능도 관찰해 보세요. 여기에는 스탠바이 리전에서 상태 점검 및 카나리아 (합성 테스트) 를 실행하여 프라이머리 리전의 상태를 외부에서 볼 수 있게 하는 것이 포함됩니다. 또한 Amazon CloudWatch Internet Monitor를 사용하여 최종 사용자 관점에서 외부 네트워크 상태와 워크로드 성능을 이해할 수 있습니다. 마찬가지로, 기본 지역에도 대기 지역을 모니터링할 수 있는 동일한 관찰 기능이 있어야 합니다. 이러한 카나리아는 고객 경험 지표를 모니터링하여 워크로드의 전반적인 상태를 파악해야 합니다. 이는 기본 지역에 문제가 발생할 경우 기본 지역의 관찰 가능성이 손상되어 워크로드 상태를 평가하는 능력에 영향을 미칠 수 있기 때문에 필요합니다.

이 경우 해당 지역 밖에서 관찰하면 통찰력을 얻을 수 있습니다. 이러한 지표를 각 지역에서 사용할 수 있는 대시보드와 각 지역에서 생성된 경보에 통합해야 합니다. CloudWatchAmazon은 지역 서비스이므로 두 지역 모두에 서비스를 제공하는 것이 필수입니다. 이 모니터링 데이터는 기본 지역에서 대기 지역으로의 장애 조치를 호출하는 데 사용됩니다.

4d: 프로세스, 절차 및 테스트

'언제 페일오버를 해야 할까요? '라는 질문에 답하기 가장 좋은 시기입니다. 필요하기까지는 훨씬 더 오래 걸렸습니다. 사람, 프로세스, 기술을 포함하는 비즈니스 연속성 계획은 모두 문제가 발생하기 전에 미리 정의하고 정기적으로 테스트해야 합니다. 복구 결정 프레임워크를 결정하십시오. 잘 실행된 복구 프로세스가 있고 복구 시간을 잘 이해하고 있는 경우 페일오버를 통해 RTO 목표를 충족하는 복구 프로세스를 시작할 시점을 선택할 수 있습니다. 이 시점은 기본 지역의 애플리케이션에 문제가 발견된 직후일 수도 있고, 지역 내 애플리케이션 내의 복구 옵션이 모두 사용되어 RTO를 충족하기 위해 이제 페일오버를 시작해야 하는 상황이 더 진행되었을 수도 있습니다.

페일오버 작업 자체는 100% 자동화되어야 하지만 페일오버를 활성화하는 결정은 사람 (일반적으로 조직 내에서 미리 결정된 소수의 개인) 이 내려야 합니다. 또한 페일오버를 결정하는 기준을 명확하게 정의하고 조직과 함께 전 세계적으로 이해해야 합니다. 이러한 프로세스는 AWSSystem Manager 런북을 사용하여 정의하고 완료할 수 있으며, 이를 통해 완전한 end-to-end 자동화가 가능하고 테스트 및 페일오버 중에 실행되는 프로세스의 일관성을 보장할 수 있습니다.

페일오버 또는 페일백 프로세스를 시작하려면 기본 및 스탠바이 리전에서 이러한 런북을 사용할 수 있어야 합니다. 이 자동화가 적용되면 정기적인 테스트 케이던스를 정의하고 준수해야 합니다. 이렇게 하

면 실제 이벤트가 발생했을 때 조직이 신뢰할 수 있는 잘 정의되고 실행된 프로세스에 따라 대응이 실행될 수 있습니다. 또한 데이터 조정 프로세스에 대해 설정된 허용 오차를 염두에 두는 것도 중요합니다. 설정된 RPO/RTO 요구 사항이 제안된 프로세스와 일치하는지 확인하십시오.

4e: 비용 및 복잡성

멀티 리전 아키텍처가 비용에 미치는 영향은 인프라 사용량, 운영 오버헤드 및 리소스 시간의 증가에 의해 좌우됩니다. 앞서 언급했듯이 스탠바이 리전의 인프라 비용은 사전 프로비저닝 시 프라이머리 리전의 인프라 비용과 비슷하므로 비용이 2배 더 비쌉니다. 일일 운영에 충분하도록 용량을 프로비저닝하면서도 수요 급증을 견딜 수 있을 만큼 충분한 버퍼 용량을 확보하고 각 지역에 동일한 제한을 구성하십시오.

또한 액티브-액티브 아키텍처를 채택하는 경우 멀티 리전 아키텍처에서 성공적으로 실행하려면 애플리케이션 수준의 변경이 필요할 수 있습니다. 이 경우 설계 및 운영에 시간과 리소스가 많이 소요될 수 있습니다. 조직은 최소한 각 지역의 기술 및 비즈니스 종속성을 이해하고 페일오버 및 페일백 프로세스를 설계하는 데 시간을 할애해야 합니다.

또한 팀은 일반적인 페일오버 및 페일백 연습을 거쳐야 이벤트 중에 사용할 런북에 익숙해질 수 있습니다. 여러 지역에 투자하여 기대되는 결과를 얻는 데 매우 중요하고 중요하기는 하지만, 이러한 연습은 기회 비용을 초래하고 다른 활동에 소요되는 시간과 자원을 낭비합니다.

주요 지침

- AWS워크로드가 운영될 모든 지역에서 서비스 할당량을 검토하고 동등하게 검토해야 합니다.
- 배포 프로세스는 여러 지역을 동시에 대상으로 하는 것이 아니라 한 번에 하나의 지역을 대상으로 해야 합니다.
- 복제 지연과 같은 추가 지표를 모니터링해야 하며, 이는 다중 지역 시나리오에만 해당됩니다.
- 워크로드에 대한 모니터링을 기본 지역 이상으로 확장하십시오. 고객 경험 지표는 지역별로 모니터링하고 워크로드가 실행되는 각 지역 외부에서 측정해야 합니다.
- 페일오버와 페일백은 정기적으로 테스트해야 합니다. 테스트와 라이브 이벤트 모두에서 사용되는 페일오버 및 페일백 프로세스를 위한 단일 런북을 구현해야 합니다. 테스트용 런북과 라이브 이벤트용 런북은 다를 수 없습니다.

결론

이 백서에서는 멀티 리전의 일반적인 사용 사례, 멀티 리전 아키텍처 구현 방법에 대한 기본 사항, 이 접근 방식의 영향에 대해 설명했습니다. 이러한 기본 원칙은 모든 워크로드에 적용할 수 있으며 다중 지역 아키텍처가 특정 비즈니스에 적합한 접근 방식인지 여부를 결정하는 데 도움이 되는 프레임워크로 사용할 수 있습니다.

기여자

다음은 이 문서의 기여자입니다.

기술 기여자:

- 존 포멘토 주니어, 수석 솔루션 아키텍트, 멀티 리전 팀 AWS

에디토리얼 기고자:

- 리시 루이스, 제품 마케팅 담당 수석 관리자

참조 자료

추가 정보는 다음을 참조하세요.

- 고급 다중 AZ 복원 패턴 (AWS백서)
- 신뢰성 기둥 - AWS Well-Architected 프레임워크
- 가용성 및 그 이상: 분산 시스템의 복원력에 대한 이해 및 개선 (백서) AWS AWS
- AWS장애 격리 경계 (AWS백서)

문서 수정

이 설명서에 대한 업데이트 알림을 받으려면 RSS 피드를 구독하면 됩니다.

변경 사항	설명	날짜
문서가 게시되었습니다.	첫 번째 간행물.	2022년 12월 20일

고지 사항

고객은 본 문서의 정보를 독립적으로 평가할 책임이 있습니다. 본 문서는 (a) 정보 제공의 목적으로만 제공되고, (b) 사전 통지 없이 변경될 수 있는 현재 AWS 제품 및 관행을 나타내고, (c) AWS 및 그 계열사, 공급업체 또는 라이선스 제공자로부터 어떠한 약속이나 보증도 하지 않습니다. AWS 제품 또는 서비스는 명시적이든 묵시적이든 어떠한 종류의 보증, 진술 또는 조건 없이 '있는 그대로' 제공됩니다. 고객에 대한 AWS의 책임 및 채무는 AWS 계약에 준거합니다. 본 문서는 AWS와 고객 간의 어떠한 계약도 구성하지 않으며 이를 변경하지도 않습니다.

AWS 용어집

최신 AWS 용어는 AWS 용어집 참조의 AWS 용어집을 참조하세요.

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.