



Amazon DynamoDB 글로벌 테이블 사용

# AWS 규범적 지침



## AWS 규범적 지침: Amazon DynamoDB 글로벌 테이블 사용

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon의 상표 및 브랜드 디자인은 Amazon 외 제품 또는 서비스와 함께, Amazon 브랜드 이미지를 떨어뜨리거나 고객에게 혼동을 일으킬 수 있는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon과 제휴 관계이거나 관련이 있거나 후원 관계와 관계없이 해당 소유자의 자산입니다.

# Table of Contents

소개 .....	1
개요 .....	2
주요 사실 .....	2
사용 사례 .....	3
쓰기 모드 .....	5
임의의 리전에 쓰기(기본 리전 없음) .....	5
한 리전에 쓰기 모드(단일 기본) .....	7
사용자 리전에 쓰기 모드(혼합 기본) .....	9
라우팅 전략 .....	12
클라이언트 기반 요청 라우팅 .....	12
컴퓨팅 계층 요청 라우팅 .....	14
Route 53 요청 라우팅 .....	15
Global Accelerator 요청 라우팅 .....	17
대피 프로세스 .....	19
라이브 리전 대피 .....	19
오프라인 리전 대피 .....	19
처리량 용량 계획 .....	22
준비 체크리스트 .....	24
FAQ .....	26
글로벌 테이블의 요금은 어떻게 되나요? .....	26
글로벌 테이블은 어떤 리전을 지원하나요? .....	26
글로벌 테이블에서는 GSI를 어떻게 처리하나요? .....	26
글로벌 테이블의 복제를 중지하려면 어떻게 해야 하나요? .....	26
Amazon DynamoDB Streams는 글로벌 테이블과 어떻게 상호 작용하나요? .....	27
글로벌 테이블은 트랜잭션을 어떻게 처리하나요? .....	27
글로벌 테이블은 DynamoDB Accelerator(DAX) 캐시와 어떻게 상호 작용하나요? .....	27
테이블의 태그가 전파되나요? .....	27
테이블을 모든 리전에 백업해야 하나요 아니면 한 리전에만 백업해야 하나요? .....	27
AWS CloudFormation을 사용하여 글로벌 테이블을 배포하려면 어떻게 해야 하나요? .....	28
결론 및 리소스 .....	29
문서 기록 .....	30
용어집 .....	31
# .....	31
A .....	32

---

B .....	34
C .....	36
D .....	39
E .....	43
F .....	45
G .....	46
H .....	47
I .....	48
L .....	50
M .....	51
O .....	55
P .....	57
Q .....	59
R .....	60
S .....	62
T .....	66
U .....	67
V .....	67
W .....	68
Z .....	69
.....	lxx

# Amazon DynamoDB 글로벌 테이블 사용

Jason Hunter, Amazon Web Services(AWS)

2024년 3월 ([문서 기록](#))

글로벌 테이블은 Amazon DynamoDB의 국제적인 입지를 기반으로 구축되어 완전관리형 다중 리전 다중 활성 데이터베이스를 제공하며, 이 데이터베이스는 대규모로 확장되는 글로벌 애플리케이션의 신속한 로컬 읽기 및 쓰기 성능을 제공합니다. 글로벌 테이블은 사용자가 선택한 DynamoDB 테이블을 자동으로 복제합니다. AWS 리전글로벌 테이블은 기존 DynamoDB API를 사용하므로 애플리케이션을 변경할 필요가 없습니다. 글로벌 테이블 사용에 따른 선결제 비용이나 약정은 없으며 사용한 리소스에 대해서만 비용을 지불합니다.

이 가이드에서는 DynamoDB 글로벌 테이블의 효과적인 사용 방법에 대해 설명합니다. 글로벌 테이블에 대한 주요 정보를 제공하고, 기능의 주요 사용 사례를 설명하고, 고려해야 할 세 가지 쓰기 모델을 분류하여 소개하고, 구현할 수 있는 네 가지 주요 요청 라우팅 옵션을 살펴보고, 라이브 리전 또는 오프라인 리전에서 대피하는 방법을 설명하고, 처리량 용량 계획에 대해 생각하는 방법을 설명하며, 글로벌 테이블을 배포할 때 고려해야 할 사항에 대한 체크리스트를 제공합니다.

[이 가이드는 AWS 다중 지역 기본 사항 백서와 동영상을 통한 데이터 복원력 설계 패턴에서 다루는 것처럼 AWS 다중 지역 배포의 더 큰 맥락에 적합합니다. AWS](#)

## 목차

- [개요](#)
- [쓰기 모드](#)
- [라우팅 전략](#)
- [대피 프로세스](#)
- [처리량 용량 계획](#)
- [준비 체크리스트](#)
- [FAQ](#)
- [결론 및 리소스](#)

# 글로벌 테이블 개요

## 주요 사실

- 글로벌 테이블에는 버전 [2017.11.29 \(레거시\) \(v1이라고도 함\)](#) 와 [2019.11.21 \(현재\) \(v2라고도 함\)](#) 의 두 가지 버전이 있습니다. 이 가이드는 현재 버전에만 초점을 맞춥니다.
- DynamoDB (글로벌 테이블 제외) 는 지역 서비스이므로 가용성이 높고 전체 가용 영역의 장애를 비롯한 인프라 장애에 대한 내재적 복원력이 뛰어납니다. 단일 지역 DynamoDB 테이블은 99.99% 의 가용성을 제공하도록 설계되었습니다. 자세한 내용은 [DynamoDB 서비스 수준 계약 \(SLA\)](#) 을 참조하십시오.
- DynamoDB 글로벌 테이블은 둘 이상의 리전 간에 데이터를 복제합니다. 다중 지역 DynamoDB 테이블은 99.999% 가용성을 제공하도록 설계되었습니다. 적절한 계획을 세우면 글로벌 테이블은 지역적 장애에 대한 복원력을 갖춘 아키텍처를 만드는 데 도움이 될 수 있습니다.
- 글로벌 테이블은 액티브-액티브 복제 모델을 사용합니다. DynamoDB의 관점에서 볼 때 각 리전의 테이블은 읽기 및 쓰기 요청을 수락할 수 있는 동등한 지위를 갖습니다. 쓰기 요청을 받은 후 로컬 복제본 테이블은 쓰기 작업을 백그라운드에서 참여 중인 다른 원격 지역에 복제합니다.
- 항목은 개별적으로 복제됩니다. 단일 트랜잭션 내에서 업데이트된 항목은 함께 복제되지 않을 수 있습니다.
- 원본 지역의 각 테이블 파티션은 다른 모든 파티션과 병렬로 쓰기 작업을 복제합니다. 원격 리전 내의 쓰기 작업 순서가 소스 리전 내에서 발생한 쓰기 작업 순서와 일치하지 않을 수 있습니다. 테이블 파티션에 대한 자세한 내용은 블로그 게시물 [Scaling DynamoDB: How partitions, hot keys, and split for heat impact performance](#) 를 참조하세요.
- 글로벌 테이블에 새로 쓰여진 항목은 일반적으로 1초 안에 모든 복제본 테이블에 전파됩니다. 가까운 리전이 대체로 더 빨리 전파됩니다.
- CloudWatchAmazon은 각 지역 쌍에 대한 ReplicationLatency 메트릭을 제공합니다. 이 값은 도착하는 항목을 보고 도착 시간을 초기 쓰기 시간과 비교하고 평균을 계산하여 계산합니다. 타이밍은 소스 지역 CloudWatch 내에 저장됩니다. 평균 및 최대 타이밍을 보면 평균 및 최악의 복제 지연을 결정하는 데 유용할 수 있습니다. 이 지연 시간에는 SLA가 없습니다.
- 개별 항목이 서로 다른 두 지역에서 거의 동시에 (이 ReplicationLatency 창 내에서) 업데이트되고 첫 번째 쓰기 작업이 복제되기 전에 두 번째 쓰기 작업이 발생하는 경우 쓰기 충돌이 발생할 수 있습니다. 글로벌 테이블은 쓰기 작업의 타임스탬프를 기반으로 하는 Last Writer Wins 메커니즘을 사용하여 이러한 충돌을 해결합니다. 첫 번째 작업은 두 번째 작업으로 "손실"됩니다. 이러한 충돌은 CloudWatch 또는 에 기록되지 않습니다AWS CloudTrail.

- 각 항목에는 비공개 시스템 속성으로 보관되는 마지막 쓰기 타임스탬프가 있습니다. Last Writer Wins 접근 방식은 수신 항목의 타임스탬프가 기존 항목의 타임스탬프보다 커야 하는 조건부 쓰기 작업을 사용하여 구현됩니다.
- 글로벌 테이블은 모든 항목을 모든 참여 지역에 복제합니다. 다른 복제 범위를 지정하려는 경우 여러 글로벌 테이블을 생성하고 각 테이블에 서로 다른 참여 지역을 할당할 수 있습니다.
- 로컬 리전은 복제본 리전이 오프라인이거나 리전이 ReplicationLatency 커져도 쓰기 작업을 허용합니다. 로컬 테이블은 각 항목이 성공할 때까지 원격 테이블에 항목 복제를 계속 시도합니다.
- 드문 경우이긴 하지만 리전이 완전히 오프라인 상태가 되는 경우 나중에 다시 온라인 상태가 되면 보류 중인 모든 아웃바운드 및 인바운드 복제가 재시도됩니다. 테이블을 다시 동기화하기 위한 특별한 작업은 필요하지 않습니다. Last Writer Wins 메커니즘은 데이터가 최종적으로 일관성을 유지하도록 합니다.
- 언제든지 DynamoDB 테이블에 새 리전을 추가할 수 있습니다. DynamoDB는 초기 동기화 및 진행 중인 복제를 처리합니다. 또한 리전 (원래 리전 포함) 을 제거할 수 있으며, 이렇게 하면 해당 리전의 로컬 테이블이 삭제됩니다.
- DynamoDB에는 글로벌 엔드포인트가 없습니다. 모든 요청은 해당 리전의 로컬인 글로벌 테이블 인스턴스에 액세스하는 리전 엔드포인트로 전송됩니다.
- DynamoDB에 대한 호출은 리전 간에 전송되어서는 안 됩니다. 가장 좋은 방법은 한 지역에 위치한 애플리케이션이 해당 지역의 로컬 DynamoDB 엔드포인트에만 직접 액세스하도록 하는 것입니다. 한 지역 (DynamoDB 계층 또는 주변 스택) 내에서 문제가 감지되면 최종 사용자 트래픽을 다른 지역에서 호스팅되는 다른 애플리케이션 엔드포인트로 라우팅해야 합니다. 글로벌 테이블을 통해 모든 리전에 있는 애플리케이션이 동일한 데이터에 액세스할 수 있습니다.

## 사용 사례

글로벌 테이블은 다음과 같은 일반적인 이점을 제공합니다.

- 지연 시간이 짧은 읽기 작업 데이터 사본을 최종 사용자 가까이에 두어 읽기 작업 중 네트워크 지연 시간을 줄일 수 있습니다. 데이터는 ReplicationLatency 가치만큼이나 최신 상태로 유지됩니다.
- 지연 시간이 짧은 쓰기 작업 최종 사용자는 가까운 지역에 쓰기 작업을 수행하여 네트워크 지연 시간을 줄이고 쓰기 작업을 완료하는 데 걸리는 시간을 줄일 수 있습니다. 쓰기 트래픽은 충돌이 발생하지 않도록 주의해서 라우팅해야 합니다. 라우팅 기법은 [이후 섹션에서](#) 설명합니다.
- 복원력 및 재해 복구 향상. 특정 리전의 성능이 저하되거나 완전히 중단된 경우 해당 리전을 회피 (해당 리전으로 전송되는 요청 일부 또는 전체 이동) 하여 초 단위로 측정되는 복구 시점 목표 (RPO) 및 복구 시간 목표 (RTO) 를 달성할 수 있습니다. 또한 글로벌 테이블을 사용하면 [DynamoDB SLA의](#) 월간 가동률이 99.99% 에서 99.999% 로 증가합니다.

- 원활한 리전 마이그레이션. 새 지역을 추가한 다음 이전 지역을 삭제하여 데이터 계층의 가동 중단 없이 한 지역에서 다른 지역으로 배포를 마이그레이션할 수 있습니다.

예를 들어, 피델리티 인베스트먼트는 [re:Invent 2022에서 DynamoDB 글로벌 테이블을 주문 관리 시스템에 사용하는 방법에 대해 발표했습니다](#). 이 회사의 목표는 가용 영역 및 지역 장애에 대한 복원력을 유지하는 동시에 온프레미스 처리로는 달성할 수 없는 규모로 안정적으로 짧은 지연 시간을 처리하는 것이었습니다.

# 글로벌 테이블의 쓰기 모드

글로벌 테이블은 테이블 수준에서 항상 액티브-액티브입니다. 그러나 쓰기 요청을 라우팅하는 방법을 제어하여 이를 액티브-패시브로 처리하는 것이 좋을 경우가 있습니다. 예를 들어 잠재적인 쓰기 충돌을 방지하기 위해 쓰기 요청을 단일 리전으로 라우팅하기로 결정할 수 있습니다.

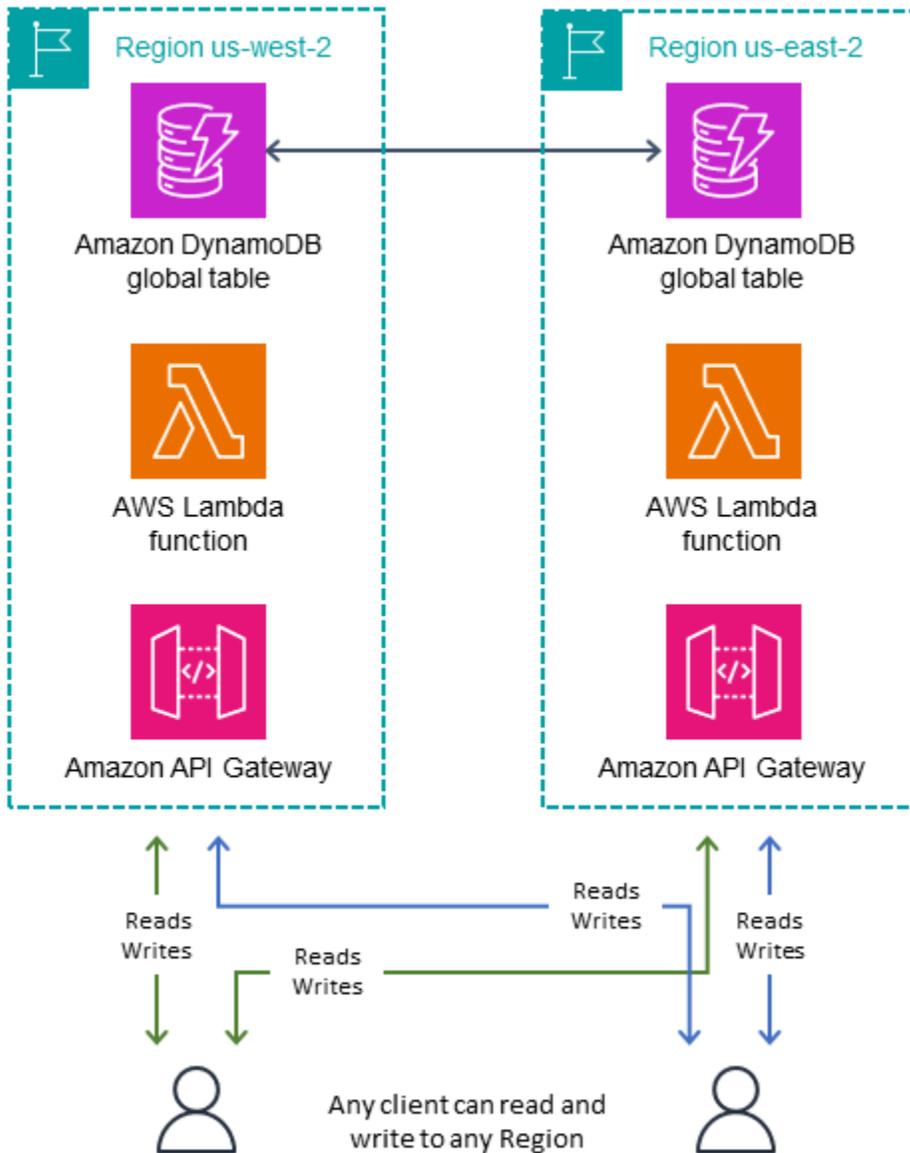
다음 세 섹션에서 설명하는 것처럼 세 가지 주요 관리 쓰기 패턴이 있습니다. 어떤 쓰기 패턴이 사용 사례에 적합한지 고려해야 합니다. 이 선택은 요청 라우팅, 리전 대피, 재해 복구 처리 방법에 영향을 미칩니다. 이후 섹션의 지침은 애플리케이션의 쓰기 모드에 따라 달라집니다.

## 주제

- [임의의 리전에 쓰기\(기본 리전 없음\)](#)
- [한 리전에 쓰기 모드\(단일 기본\)](#)
- [사용자 리전에 쓰기 모드\(혼합 기본\)](#)

## 임의의 리전에 쓰기(기본 리전 없음)

모든 리전에 쓰기 모드는 완전 활성 모드이며 쓰기 작업이 발생할 수 있는 위치에 제한을 두지 않습니다. 모든 지역은 언제든지 쓰기 요청을 수락할 수 있습니다. 가장 간단한 모드이지만 일부 유형의 응용 프로그램에서만 사용할 수 있습니다. 모든 쓰기 작업이 무력한 경우에 적합합니다. 중복성은 사용자가 연락처 데이터를 업데이트하는 경우와 같이 지역 간의 동시 또는 반복 쓰기 작업이 충돌하지 않도록 안전하게 반복 가능함을 의미합니다. 또한 모든 쓰기 작업이 확정적 기본 키 아래에 고유한 삽입이 이루어지는 추가 전용 데이터세트에서도 잘 작동하는데, 이는 등등성이 있는 특수한 경우입니다. 마지막으로, 이 모드는 쓰기 작업의 충돌 위험이 허용되는 경우에 적합합니다.



임의의 리전에 쓰기 모드는 가장 간단하게 구현할 수 있는 아키텍처입니다. 어느 리전이나 언제든지 쓰기 대상이 될 수 있으므로 라우팅이 더 쉽습니다. 최근의 모든 쓰기 작업을 보조 리전에 원하는 횟수만큼 재생할 수 있으므로 페일오버가 더 쉽습니다. 가능하면 이 쓰기 모드에 맞춰 설계해야 합니다.

예를 들어, 여러 비디오 스트리밍 서비스는 북마크, 리뷰, 시청 상태 플래그 등을 추적하기 위해 글로벌 테이블을 사용합니다. 이러한 배포에서는 모든 쓰기 작업이 동일한지 확인하는 한 모든 리전에 쓰기 모드를 사용할 수 있습니다. 새 최신 시간 코드 설정, 새 리뷰 할당, 새 시계 상태 설정 등 모든 업데이트가 사용자의 새 상태를 직접 할당하고 항목의 다음 올바른 값이 현재 값에 좌우되지 않는 경우가 이에 해당합니다. 우연히 사용자의 쓰기 요청이 다른 지역으로 라우팅되는 경우 마지막 쓰기 작업이 지속되고 글로벌 상태는 마지막 할당에 따라 해결됩니다. 이 모드에서의 읽기 작업은 결국 일관성이 유지되고 최신 ReplicationLatency 값만큼 지연됩니다.

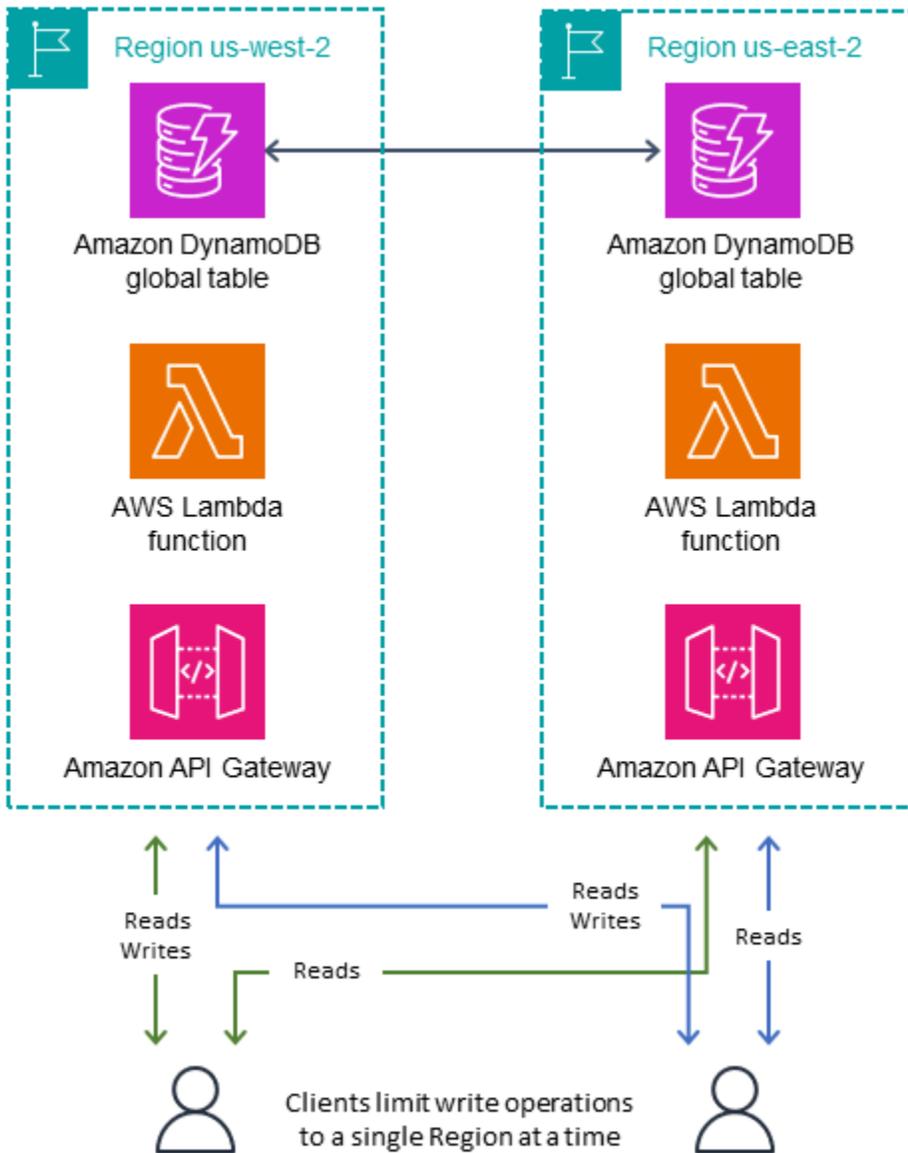
또 다른 예로 한 금융 서비스 회사는 시스템의 일부로 글로벌 테이블을 사용하여 각 고객의 직불 카드 구매 현황을 지속적으로 집계하여 해당 고객의 캐시백 보상을 계산합니다. 신규 거래는 전 세계에서 유입되어 여러 리전으로 이동합니다. 이 회사는 세심한 재설계를 통해 모든 리전 모드에 쓰기 기능을 사용할 수 있었습니다. 초기 설계 스케치에서는 고객당 하나의 RunningBalance 항목으로 유지되었습니다. 고객 조치로 밸런스가 ADD 표현식으로 업데이트되었는데, 이는 정확한 새 값이 현재 값에 따라 달라지기 때문입니다. 서로 다른 리전에서 거의 같은 시간에 동일한 밸런스에 대한 쓰기 작업이 두 번 발생하면 밸런스가 동기화되지 않았습니다. 재설계에서는 추가 전용 워크플로우가 있는 원장처럼 작동하는 이벤트 스트리밍을 사용합니다. 각 고객 작업은 해당 고객을 위해 유지 관리되는 항목 컬렉션에 새 항목을 추가합니다. (항목 컬렉션은 기본 키를 공유하지만 정렬 키가 다른 항목 집합입니다.) 각 쓰기 작업은 고객 ID를 파티션 키로 사용하고 트랜잭션 ID를 정렬 키로 사용하는 즉석 삽입입니다. 이 설계에서는 항목을 가져온 다음 클라이언트측 계산을 수행해야 Query 하기 때문에 밸런스 계산이 더 어려워지지만 모든 쓰기 작업이 중복되고 라우팅과 페일오버가 크게 단순화됩니다. (이에 대해서는 이 설명서 후반부의 섹션을 자세히 설명합니다.)

세 번째 예로는 온라인 광고 게재 서비스를 제공하는 회사가 있습니다. 이 회사는 데이터 손실 위험이 낮으면 모든 지역 모드에 대한 쓰기 설계의 단순화를 달성할 수 있다고 판단했습니다. 광고를 게재할 때 몇 밀리초 안에 충분한 메타데이터를 검색하여 어떤 광고를 게재할지 결정한 다음 광고 노출을 기록하여 동일한 광고를 곧 반복하지 않도록 할 수 있습니다. 글로벌 테이블을 사용하여 전 세계 최종 사용자에게 지연 시간이 짧은 읽기 작업과 지연 시간이 짧은 쓰기 작업을 모두 제공합니다. 사용자의 모든 광고 노출을 단일 항목 내에 기록하며, 이 항목은 증가 목록으로 표시됩니다. 항목 컬렉션에 추가하는 대신 하나의 항목을 사용하므로 삭제 작업에 대한 비용을 지불하지 않고도 각 쓰기 작업의 일부로 오래된 광고 노출을 제거할 수 있습니다. 이 쓰기 작업은 불안정하지 않습니다. 동일한 최종 사용자가 거의 동시에 여러 지역에서 게재되는 광고를 보게 되면 하나의 광고 노출에 대한 하나의 쓰기 작업이 다른 쓰기 작업을 덮어쓸 수 있습니다. 위험은 사용자가 때때로 반복되는 광고를 보게 될 수 있다는 것입니다. 그들은 이것이 받아들일 수 있다고 결정했습니다.

## 한 리전에 쓰기 모드(단일 기본)

한 리전에 쓰기 모드는 액티브-패시브이며 모든 테이블 쓰기 작업을 단일 활성 리전으로 라우팅합니다. (DynamoDB에는 단일 활성 지역이라는 개념이 없습니다. DynamoDB 외부의 계층이 이를 관리합니다.) 한 지역에 쓰기 모드에서는 쓰기 작업이 한 번에 한 지역에만 흐르도록 하여 쓰기 충돌을 방지합니다. 이 쓰기 모드는 조건식이나 트랜잭션을 사용하려는 경우에 유용합니다. 이러한 표현식은 최신 데이터를 기반으로 행동한다는 사실을 모르는 한 사용할 수 없으므로 모든 쓰기 요청을 최신 데이터가 있는 단일 리전으로 보내야 합니다.

결국 일관된 읽기 작업을 모든 복제본 리전으로 이동하여 지연 시간을 줄일 수 있습니다. 매우 일관된 읽기 작업은 단일 기본 리전으로 이동해야 합니다.



나중에 설명하겠지만 지역 장애에 대응하여 활성 지역을 변경해야 하는 경우가 있습니다. 일부 사용자는 follow-the-sun 배포 구현과 같은 정기적인 일정에 따라 현재 활성화된 지역을 변경합니다. 이렇게 하면 활동이 가장 많은 지역 (보통 낮이라 이름 붙여짐) 근처에 활성 지역이 배치되어 읽기 및 쓰기 작업 지연 시간이 가장 짧습니다. 또한 지역 변경 코드를 매일 호출하고 재해 복구 전에 제대로 테스트되었는지 확인할 수 있다는 부수적인 이점도 있습니다.

패시브 리전은 DynamoDB가 액티브 리전이 될 경우에만 구축되는 DynamoDB를 둘러싼 축소된 인프라를 유지할 수 있습니다. 이 가이드에서는 파일럿 라이트 및 워밍 스탠바이 설계에 대해서는 다루지 않습니다. 자세한 내용은 블로그 게시물인 [재해 복구 \(DR\) 아키텍처 AWS, 파트 III: 파일럿 라이트 및 워밍 스탠바이에서](#) 읽을 수 있습니다.

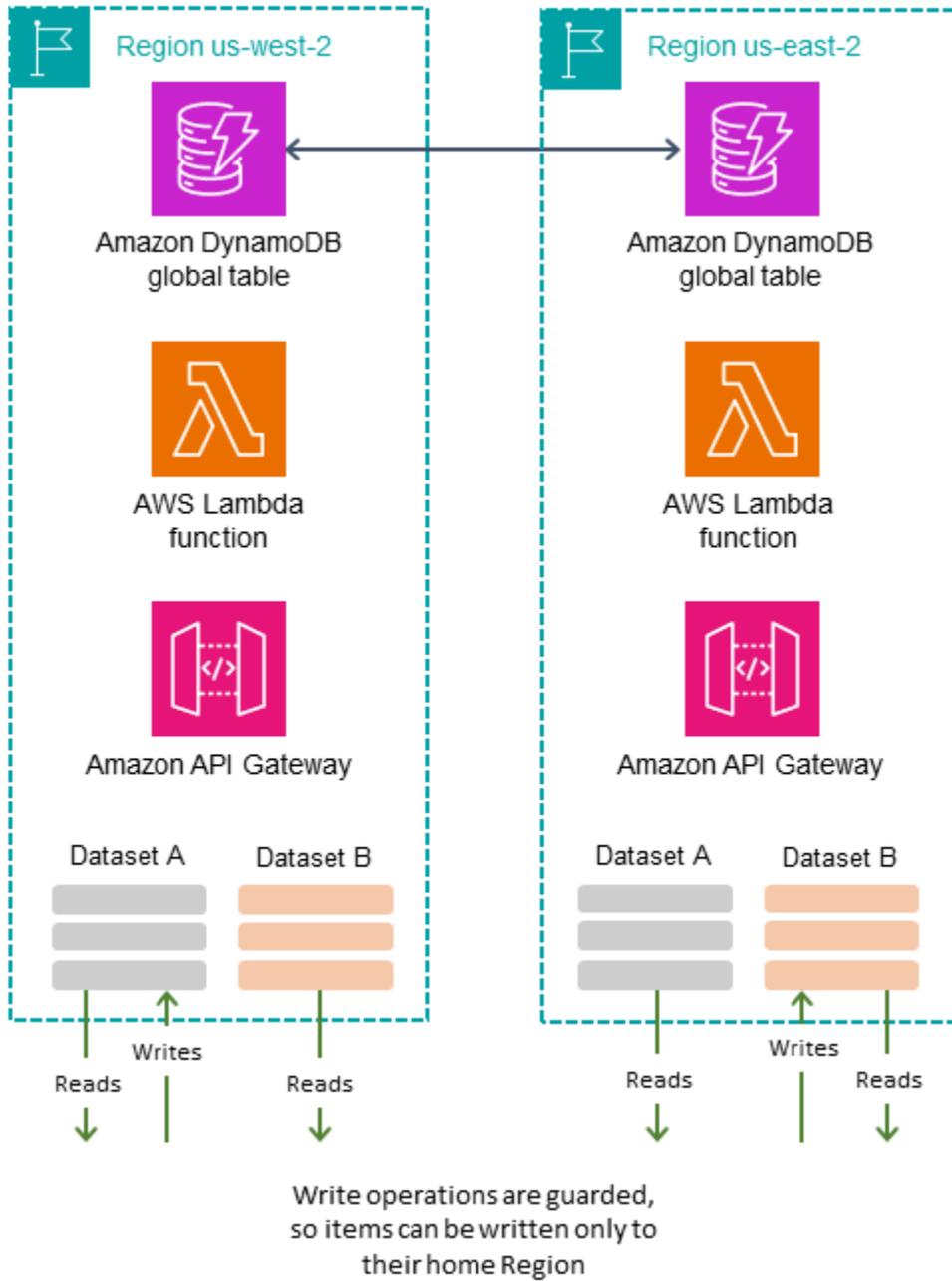
지연 시간이 짧고 전역으로 분산된 읽기 작업에 글로벌 테이블을 사용할 때는 한 리전에 쓰기 모드를 사용하는 것이 좋습니다. 전 세계 모든 지역에서 동일한 참조 데이터를 사용할 수 있어야 하는 대형 소셜 미디어 회사를 예로 들 수 있습니다. 데이터를 자주 업데이트하지는 않지만 업데이트할 때는 잠재적인 쓰기 충돌을 피하기 위해 한 지역에만 기록합니다. 읽기 작업은 모든 지역에서 항상 허용됩니다.

또 다른 예로, 앞서 설명한 금융 서비스 회사가 일일 캐시백 계산을 구현했다고 가정해 보겠습니다. 잔액을 계산할 때는 모든 지역에 쓰기 모드를 사용했지만 캐시백 지급액을 추적하려면 한 지역 모드로 쓰기 모드를 사용했습니다. 10달러를 지출할 때마다 1페니씩 보상하고 싶다면 전날의 모든 거래에 Query 대해 총 지출 금액을 계산하고 캐시백 결정을 새 테이블에 기록하고 쿼리된 항목 세트를 삭제하여 소비된 것으로 표시하고 남은 금액을 저장하는 단일 항목으로 교체해야 합니다. 이 작업에는 트랜잭션이 필요하므로 한 지역에 쓰기 모드에서 더 잘 작동합니다. 워크로드가 중복될 가능성이 없는 한 애플리케이션은 동일한 테이블에서도 쓰기 모드를 혼합할 수 있습니다.

## 사용자 리전에 쓰기 모드(혼합 기본)

리전에 쓰기 모드에서는 서로 다른 데이터 서브셋을 여러 홈 리전에 할당하고 홈 리전을 통해서만 항목에 쓰기 작업을 수행할 수 있습니다. 이 모드는 액티브-패시브이지만 아이템에 따라 액티브 리전을 할당합니다. 모든 리전은 중복되지 않는 자체 데이터세트의 기본 리전이므로 적절한 지역성을 보장하려면 쓰기 작업을 보호해야 합니다.

이 모드는 각 사용자와 관련된 데이터를 해당 사용자와 가까운 네트워크에 배치할 수 있기 때문에 쓰기 작업 지연 시간이 짧다는 점을 제외하면 한 지역에 쓰기 모드와 유사합니다. 또한 주변 인프라가 지역 간에 더 균등하게 분산되고 모든 리전이 인프라의 일부가 이미 활성화되어 있기 때문에 장애 조치 시나리오 중에 인프라를 구축하는 데 드는 작업이 줄어듭니다.



다음과 같은 여러 가지 방법으로 아이템의 홈 지역을 결정할 수 있습니다.

- **내재적:** 특수 속성이나 파티션 키에 내장된 값과 같은 데이터의 일부 측면을 통해 홈 리전이 명확해 집니다. 이 기법은 [Amazon DynamoDB 글로벌 테이블의 항목에 대한 홈 리전을 설정하기 위한 리전 고정 사용](#) 블로그 게시물에 설명되어 있습니다.
- **협상됨:** 각 데이터세트의 홈 리전은 할당을 유지 관리하는 별도의 글로벌 서비스와 같은 외부 방식으로 협상됩니다. 배정 기간이 한정될 수 있으며 그 이후에는 재협상 대상이 될 수 있습니다.

- 테이블 기반: 하나의 복제 글로벌 테이블을 만드는 대신 리전을 복제하는 것과 동일한 수의 글로벌 테이블을 생성합니다. 각 테이블의 이름은 해당 테이블의 홈 리전을 나타냅니다. 표준 운영에서는 모든 데이터를 홈 리전에 쓰고 다른 리전들은 읽기 전용 사본을 유지합니다. 페일오버 중에는 다른 리전이 해당 테이블에 대한 쓰기 작업을 일시적으로 채택합니다.

예를 들어 게임 회사에서 일하고 있다고 가정해 봅시다. 전 세계 모든 게이머에게 지연 시간이 짧은 읽기 및 쓰기 작업이 필요합니다. 각 게이머에게 가장 가까운 지역에 배정합니다. 해당 지역은 모든 읽기 및 쓰기 작업을 대신하여 강력한 read-after-write 일관성을 보장합니다. 하지만 게이머가 여행을 떠나거나 홈 지역에 정전이 발생한 경우 대체 리전에서 전체 데이터 사본을 사용할 수 있으며 게이머를 다른 홈 리전에 배정할 수 있습니다.

또 다른 예로, 여러분이 화상 회의 회사에서 일하고 있다고 가정해 봅시다. 각 컨퍼런스 콜의 메타데이터는 특정 지역에 할당됩니다. 발신자는 가장 가까운 지역을 사용하여 지연 시간을 최소화할 수 있습니다. 지역 장애가 발생한 경우 글로벌 테이블을 사용하면 시스템에서 호출 처리를 복제된 데이터 사본이 이미 있는 다른 지역으로 이동할 수 있으므로 빠른 복구가 가능합니다.

## 글로벌 테이블의 라우팅 전략

글로벌 테이블 배포에서 가장 복잡한 부분은 아마도 요청 라우팅 관리일 것입니다. 요청은 먼저 최종 사용자에서 출발하여 어떤 방식을 통해 선택 및 라우팅되는 리전으로 가야 합니다. 요청 시 해당 지역에서 일부 서비스 스택을 발견합니다. 여기에는 AWS Lambda 함수, 컨테이너 또는 Amazon Elastic Compute Cloud EC2 (Amazon) 노드가 지원하는 로드 밸런서로 구성될 수 있는 컴퓨팅 계층과 다른 데이터베이스 등 다른 서비스가 포함될 수 있습니다. 이 컴퓨팅 레이어는 DynamoDB와 통신합니다. 이 작업을 수행하려면 해당 지역의 로컬 엔드포인트를 사용해야 합니다. 글로벌 테이블의 데이터는 다른 모든 참여 리전에 복제되며, 각 리전의 DynamoDB 테이블에는 비슷한 서비스 스택이 있습니다.

글로벌 테이블은 다양한 리전의 각 스택에 동일한 데이터의 로컬 사본을 제공합니다. 단일 리전의 단일 스택을 상정한 설계를 고려하여 로컬 DynamoDB 테이블에 문제가 생기면 보조 리전의 DynamoDB 엔드포인트에 원격 호출을 하는 방법을 예상할 수 있습니다. 이는 모범 사례가 아닙니다. 리전 간 이동과 관련된 지연 시간은 로컬 액세스보다 100배 더 길 수 있습니다. back-and-forth 일련의 5개 요청은 로컬에서 수행될 경우 밀리초, 전 세계에서 수행할 때는 몇 초 정도 걸릴 수 있습니다. 처리를 위해 최종 사용자를 다른 리전으로 라우팅하는 것이 더 좋습니다. 복원력을 보장하려면 여러 지역에 걸친 복제가 필요합니다. 즉, 데이터 계층뿐만 아니라 컴퓨팅 계층도 복제해야 합니다.

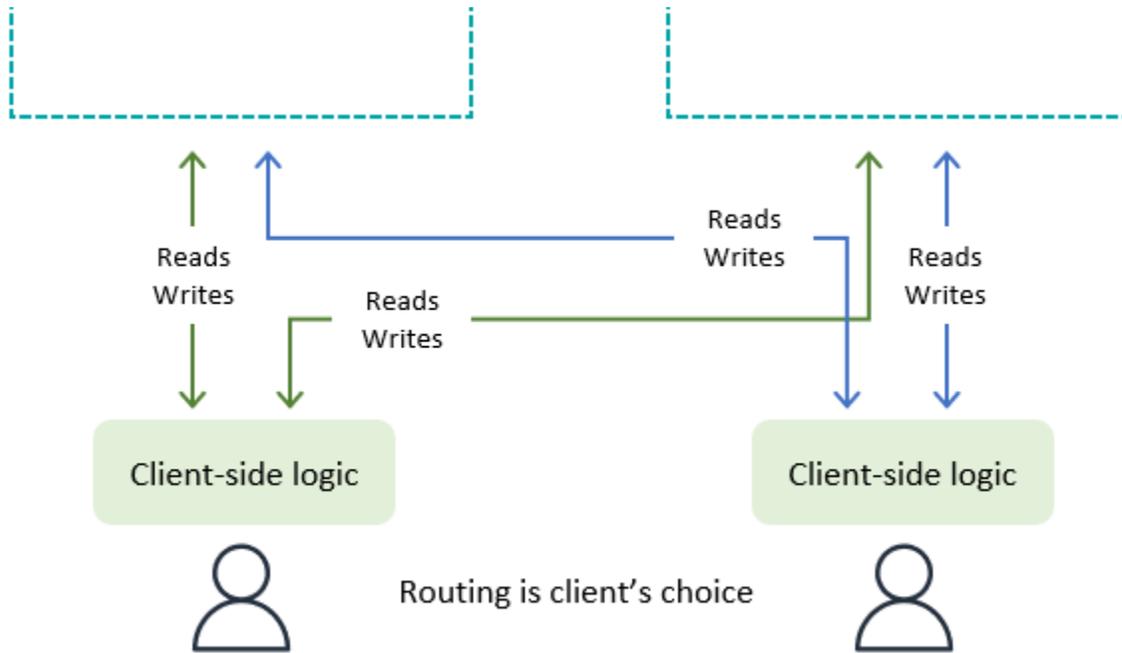
처리를 위해 최종 사용자 요청을 지역으로 라우팅하는 방법에는 여러 가지가 있습니다. 쓰기 모드와 페일오버 고려 사항에 따라 올바른 선택이 달라집니다. 이 섹션에서는 네 가지 옵션, 즉 클라이언트 기반, 컴퓨팅 레이어, Amazon Route 53 및 에 대해 설명합니다. AWS Global Accelerator

### 주제

- [클라이언트 기반 요청 라우팅](#)
- [컴퓨팅 계층 요청 라우팅](#)
- [Route 53 요청 라우팅](#)
- [Global Accelerator 요청 라우팅](#)

## 클라이언트 기반 요청 라우팅

클라이언트 기반 요청 라우팅을 사용하면 최종 사용자 클라이언트 (애플리케이션, 포함된 JavaScript 웹 페이지 또는 다른 클라이언트) 가 유효한 애플리케이션 엔드포인트 (예: 리터럴 DynamoDB 엔드포인트가 아닌 Amazon API Gateway 엔드포인트) 를 추적하고 자체 내장 로직을 사용하여 통신할 지역을 선택합니다. 무작위 선택, 가장 낮은 관찰 지연 시간, 가장 높은 관측 대역폭 측정 또는 로컬에서 수행된 상태 확인을 기반으로 선택할 수 있습니다.



클라이언트 기반 요청 라우팅은 성능 저하가 감지되면 실제 공용 인터넷 트래픽 상황과 같은 상황에 맞게 조정하여 지역을 전환할 수 있다는 장점이 있습니다. 클라이언트는 모든 잠재적 엔드포인트를 알고 있어야 하지만 새로운 리전 엔드포인트를 시작하는 경우는 드뭅니다.

모든 지역에 쓰기 모드를 사용하면 클라이언트가 선호하는 엔드포인트를 일방적으로 선택할 수 있습니다. 한 리전에 대한 액세스가 손상되면 클라이언트는 다른 엔드포인트로 라우팅할 수 있습니다.

한 지역에 쓰기 모드를 사용하는 경우 클라이언트는 쓰기 요청을 현재 활성 지역으로 라우팅하는 메커니즘이 필요합니다. 이는 현재 쓰기 요청을 수락하고 있는 지역을 경험적으로 테스트하는 것과 같은 기본 메커니즘일 수 있습니다 (쓰기 거부를 확인하고 대체 지역으로 대체). 또는 글로벌 코디네이터를 사용하여 현재 애플리케이션 상태를 쿼리하는 것과 같은 복잡한 메커니즘일 수도 있습니다 (아마도 [Amazon Application Recovery Controller \(ARC\) \(ARC\)](#) 라우팅 컨트롤을 기반으로 구축되었는데, 이 컨트롤은 [5개 지역, 쿼럼 기반 시스템을 통해 이와 같은 요구 사항에 맞게 글로벌 상태를 유지할 수 있습니다](#). 클라이언트는 최종 일관성을 위해 읽기 요청을 임의의 지역으로 보낼 수 있는지 아니면 강력한 일관성을 위해 활성 지역으로 라우팅해야 하는지를 결정할 수 있습니다.

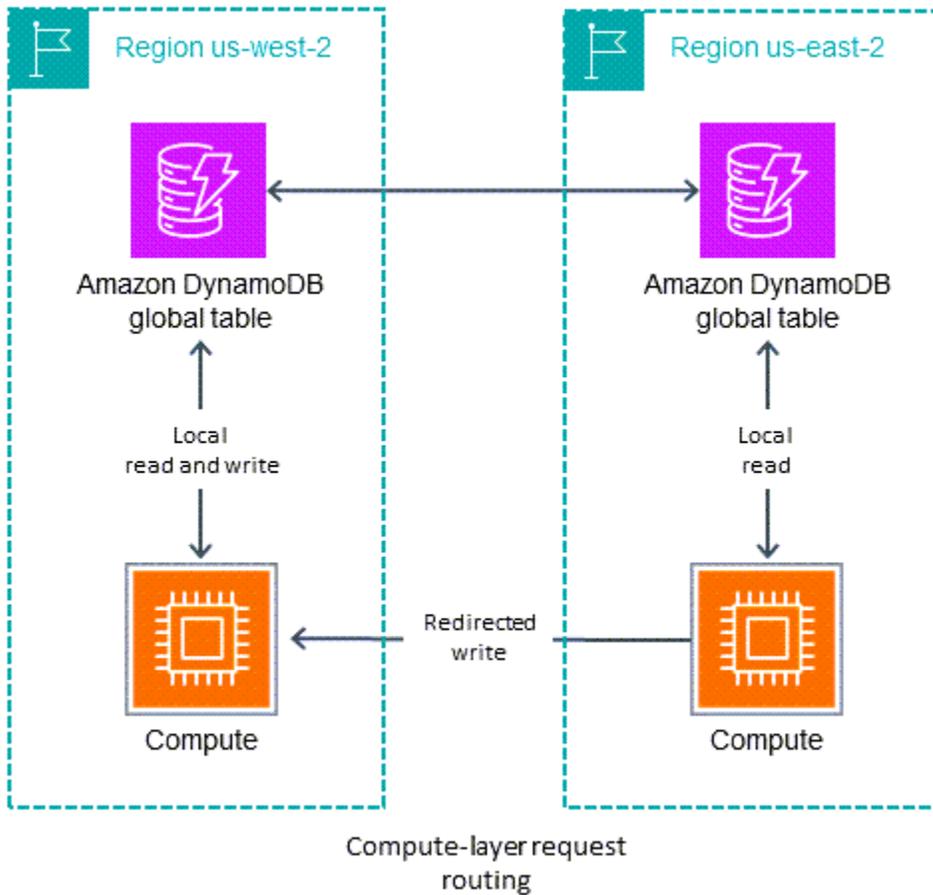
지역에 쓰기 모드를 사용할 경우 클라이언트는 사용 중인 데이터셋의 홈 지역을 결정해야 합니다. 예를 들어 클라이언트가 사용자 계정에 해당하고 각 사용자 계정이 지역에 속해 있는 경우 클라이언트는 글로벌 로그인 시스템에서 자격 증명과 함께 사용할 적절한 엔드포인트 할당을 요청할 수 있습니다.

예를 들어 웹을 통해 사용자가 비즈니스 재무를 관리할 수 있도록 지원하는 금융 서비스 회사는 지역 모드에 쓰기 기능이 있는 글로벌 테이블을 사용합니다. 각 사용자는 중앙 서비스에 로그인해야 합니다. 이 서비스는 자격 증명과 해당 자격 증명이 작동하는 지역의 엔드포인트를 반환합니다. 반환되는 지역

은 사용자 데이터세트가 현재 홈에 있는 위치를 기반으로 합니다. 보안 인증 정보는 짧은 기간 동안 유효합니다. 그 후 웹페이지는 새 로그인을 자동 협상하여 사용자의 활동을 새 지역으로 리디렉션할 수 있는 기회를 제공합니다.

## 컴퓨팅 계층 요청 라우팅

컴퓨팅 레이어 요청 라우팅을 사용하면 컴퓨팅 레이어에서 실행되는 코드가 요청을 로컬에서 처리할지 아니면 다른 지역에서 실행 중인 자체 복사본으로 전달할지를 결정합니다. 한 리전에 쓰기 모드를 사용하면 컴퓨팅 레이어가 활성 리전이 아님을 감지하여 로컬 읽기 작업을 허용하면서 모든 쓰기 작업을 다른 리전으로 전달할 수 있습니다. 이 컴퓨팅 계층 코드는 데이터 토폴로지 및 라우팅 규칙을 인식하고, 어떤 데이터에 대해 활성화되는 지역을 지정하는 최신 설정을 기반으로 데이터 토폴로지 및 라우팅 규칙을 안정적으로 적용해야 합니다. 해당 리전 내의 외부 소프트웨어 스택은 마이크로서비스가 읽기 및 쓰기 요청을 어떻게 라우팅하는지 몰라도 됩니다. 강력한 설계에서 수신 리전은 자신이 쓰기 작업의 현재 기본 리전인지 여부를 확인합니다. 기본 리전이 아니라면 글로벌 상태를 수정해야 한다는 오류 메시지가 생성됩니다. 또한 기본 리전이 변경 중일 경우 수신 리전은 쓰기 작업을 잠시 버퍼링할 수 있습니다. 어떤 경우든 리전의 컴퓨팅 스택은 해당 로컬 DynamoDB 엔드포인트에만 쓰지만 컴퓨팅 스택은 서로 통신할 수 있습니다.

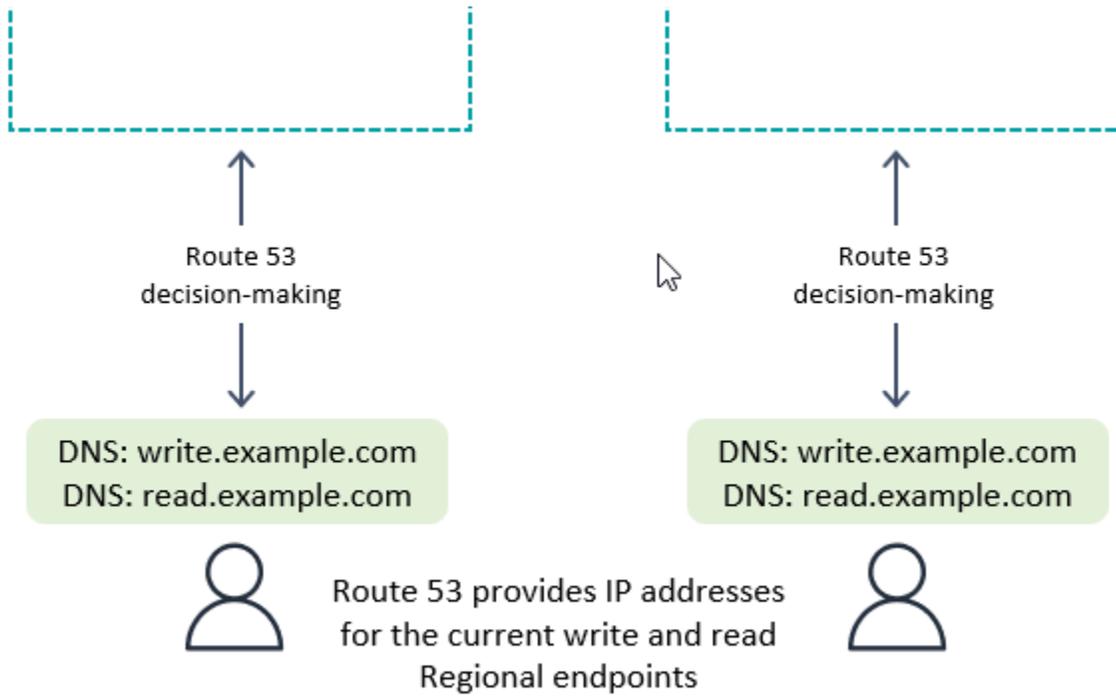


[뱅크드 그룹은 re:Invent 2022에서 소개한 바와 같이 이 라우팅 프로세스에 글로벌 오케스트레이션 및 상태 도구 \(GOaST\) 라는 시스템과 글로벌 다중 지역 라이브러리 \(GMRLib\) 라는 라이브러리를 사용합니다.](#) 이들은 단일 기본 모델을 사용합니다. follow-the-sun GOaST 이전 섹션에서 설명한 ARC 라우팅 제어와 마찬가지로 글로벌 상태를 유지합니다. 글로벌 테이블을 사용하여 어느 지역이 기본 지역인지, 다음 기본 스위치가 언제 스케줄링되는지 추적합니다. 모든 읽기 및 쓰기 작업이 진행되며 GMRLib, 이에 따라 GOaST 조정됩니다. GMRLib 읽기 작업을 낮은 지연 시간으로 로컬에서 수행할 수 있습니다. 쓰기 작업의 경우 로컬 지역이 현재 기본 지역인지 GMRLib 확인합니다. 현재 기본 리전인 경우 쓰기 작업이 직접 완료됩니다. 그렇지 않은 경우 쓰기 작업을 기본 지역의 으로 GMRLib 전달합니다. GMRLib 이 수신 라이브러리는 자신도 스스로를 기본 리전으로 간주한다는 것을 확인하고, 그렇지 않은 경우 글로벌 상태의 전파 지연을 나타내는 오류를 발생시킵니다. 이 접근 방식은 원격 DynamoDB 엔드포인트에 직접 쓰지 않으므로 검증에 도움이 됩니다.

## Route 53 요청 라우팅

Amazon Route 53은 도메인 이름 서비스 (DNS) 기술입니다. Route 53을 사용하면 클라이언트가 잘 알려진 DNS 도메인 이름을 조회하여 엔드포인트를 요청하고, Route 53은 가장 적절하다고 판단되는 지

역 엔드포인트에 해당하는 IP 주소를 반환합니다. Route 53에는 적절한 지역을 결정하는 데 사용하는 다양한 [라우팅 정책](#)이 있습니다. 또한 [장애 조치 라우팅](#)을 수행하여 [상태 확인에 실패한](#) 지역 외부로 트래픽을 라우팅할 수 있습니다.



모든 지역에 쓰기 모드를 사용하거나 백엔드의 컴퓨팅 레이어 요청 라우팅과 결합할 경우, Route 53은 가장 가까운 네트워크나 지리적 근접성에 있는 지역 선택 또는 기타 선택과 같은 복잡한 내부 규칙에 따라 자유롭게 지역을 반환할 수 있습니다.

한 지역에 쓰기 모드를 사용하면 (를 사용하여) 현재 활성화된 지역을 반환하도록 Route 53을 구성할 수 있습니다. ARC 클라이언트가 패시브 지역에 연결하려는 경우 (예: 읽기 작업용) 다른 DNS 이름을 조회할 수 있습니다.

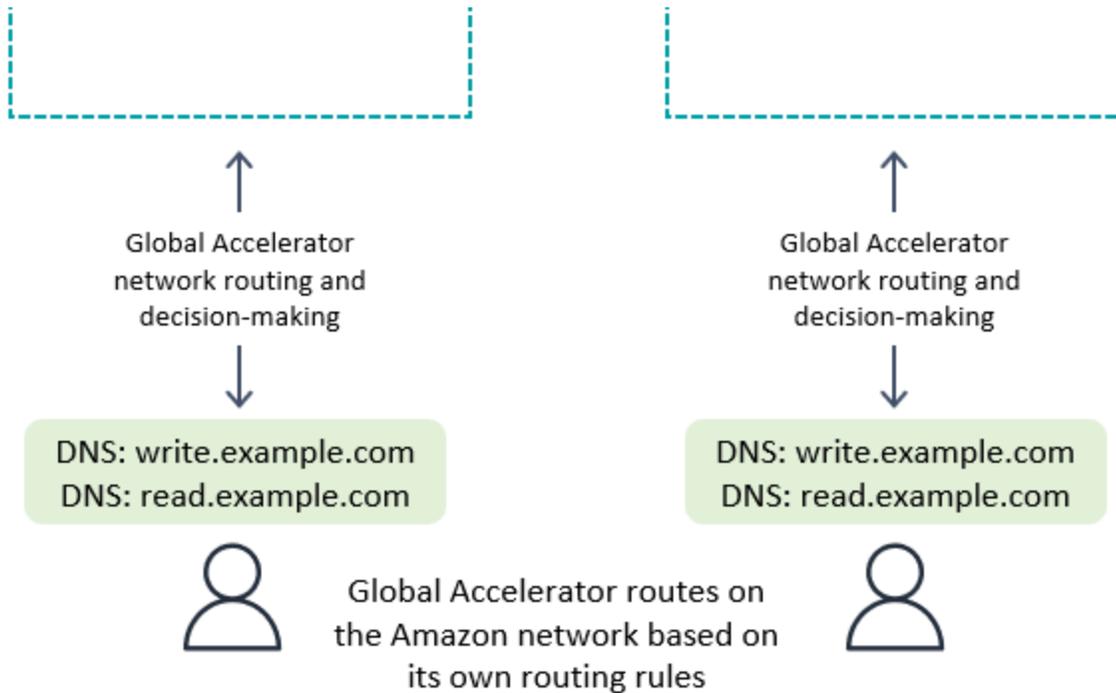
#### **Note**

클라이언트는 도메인 이름의 time to live (TTL) 설정에 표시된 시간 동안 Route 53의 응답에 IP 주소를 캐시합니다. 더 길면 모든 클라이언트가 새 엔드포인트를 인식할 수 있도록 복구 시간 목표 (RTO) 가 TTL 연장됩니다. 일반적으로 장애 조치에는 60초 값을 사용합니다. 모든 소프트웨어가 DNS TTL 만료를 완벽하게 준수하는 것은 아니며 운영 체제, 가상 시스템 및 애플리케이션과 같은 여러 수준의 DNS 캐싱이 있을 수 있습니다.

지역에 쓰기 모드에서는 컴퓨팅 레이어 요청 라우팅을 함께 사용하지 않는 한 Route 53을 사용하지 않는 것이 가장 좋습니다.

## Global Accelerator 요청 라우팅

[AWS Global Accelerator](#) 클라이언트를 사용하여 Route 53에서 잘 알려진 도메인 이름을 조회합니다. 그러나 클라이언트는 지역 엔드포인트에 해당하는 IP 주소를 다시 가져오는 대신 가장 가까운 AWS 엣지 로케이션으로 라우팅되는 애니캐스트 고정 IP 주소를 다시 가져옵니다. 이 엣지 로케이션에서 시작하여 모든 트래픽은 사실 AWS 네트워크를 통해 Global Accelerator 내에서 유지 관리되는 라우팅 규칙에 의해 선택된 지역의 일부 엔드포인트 (네트워크 로드 밸런서, 애플리케이션 로드 밸런서, EC2 인스턴스 또는 엘라스틱 IP 주소) 로 라우팅됩니다. Route 53 규칙에 기반한 라우팅과 비교할 때 Global Accelerator 요청 라우팅은 공용 인터넷의 트래픽 양을 줄이므로 지연 시간이 짧습니다. 또한 글로벌 액셀러레이터는 라우팅 규칙을 변경할 때 DNS TTL 만료에 의존하지 않으므로 라우팅을 더 빠르게 조정할 수 있습니다.



Global Accelerator는 모든 지역에 쓰기 모드를 사용하거나 백엔드의 컴퓨팅 계층 요청 라우팅과 결합할 경우 원활하게 작동합니다. 클라이언트는 가장 가까운 엣지 로케이션에 연결되므로 요청을 받는 지역에 대해 걱정할 필요가 없습니다.

단일 지역에 쓰기 모드에서는 글로벌 액셀러레이터 라우팅 규칙이 현재 활성화된 지역으로 요청을 보내야 합니다. 글로벌 시스템에서 활성 리전으로 간주되지 않는 리전의 장애를 인위적으로 보고하는 상

태 확인을 사용할 수 있습니다. 와 DNS 마찬가지로 어느 지역에서든 요청이 들어올 수 있는 경우 대체 DNS 도메인 이름을 사용하여 읽기 요청을 라우팅할 수 있습니다.

지역에 쓰기 모드에서는 컴퓨팅 계층 요청 라우팅을 함께 사용하는 경우가 아니라면 Global Accelerator를 사용하지 않는 것이 좋습니다.

## 글로벌 테이블에 대피 프로세스

지역 대피는 활동 (보통 쓰기 활동, 읽기 활동 가능) 을 해당 지역 외부로 이전하는 과정입니다.

### 라이브 리전 대피

일반적인 비즈니스 활동의 일환으로 (예: 한 리전에 쓰기 모드를 사용하는 경우), 현재 활성화된 리전을 변경하기로 한 비즈니스 결정 follow-the-sun, DynamoDB 외부 소프트웨어 스택에 장애가 발생한 경우, 또는 리전 내에서 평소보다 긴 지연 시간과 같은 일반적인 문제가 발생한 경우 등 여러 가지 이유로 라이브 리전을 제거하기로 결정할 수 있습니다.

임의의 리전에 쓰기 모드에서는 라이브 리전 대피가 간단합니다. 원하는 라우팅 시스템을 사용하여 트래픽을 대체 지역으로 라우팅하고 대피한 지역에서 이미 발생한 쓰기 작업을 평소처럼 복제할 수 있습니다.

한 지역에 쓰기 및 지역에 쓰기 모드에서는 새 활성 지역에서 쓰기 작업을 시작하기 전에 활성 지역에 대한 모든 쓰기 작업이 완전히 기록되고, 스트리밍 처리되고, 전역으로 전파되었는지 확인해야 future 쓰기 작업이 최신 버전의 데이터에 대해 처리됩니다.

리전 A는 활성이고 지역 B는 비활성이라고 가정해 보겠습니다(전체 테이블 또는 리전 A에 있는 항목의 경우). 대피를 수행하는 일반적인 메커니즘은 A에 대한 쓰기 작업을 일시 중지하고 이러한 작업이 B로 완전히 전파될 때까지 충분히 기다린 후 B를 활성으로 인식하도록 아키텍처 스택을 업데이트한 다음 B에 쓰기 작업을 재개하는 것입니다. 리전 A의 데이터가 리전 B에 완전히 복제되었음을 100% 확실하게 나타내는 지표는 없습니다. 리전 A가 정상인 경우 리전 A에 대한 쓰기 작업을 일시 중지하고 ReplicationLatency 지표의 최근 최대값의 10배를 기다리면 일반적으로 복제가 완료되었는지 확인하는 데 충분합니다. 리전 A가 비정상이고 다른 영역에서 지연 시간이 길어지면 대기 시간을 더 큰 배수로 설정할 수 있습니다.

### 오프라인 리전 대피

고려해야 할 특별한 경우가 있습니다. 지역 A가 예고 없이 완전히 오프라인 상태가 되면 어떻게 될까요? 그럴 가능성은 극히 낮지만 그래도 고려해봐야 합니다. 이런 경우에는 아직 전파되지 않은 리전 A의 모든 쓰기 작업은 보관되었다가 리전 A가 다시 온라인 상태가 된 후에 전파됩니다. 쓰기 작업은 손실되지 않지만 전파는 무기한 지연됩니다.

이 경우 어떻게 진행할지는 애플리케이션이 결정합니다. 비즈니스 연속성을 위해서는 새 기본 리전 B에 쓰기 작업을 계속해야 할 수도 있습니다. 하지만 리전 A로부터 항목에 대한 쓰기 작업 전파가 보류

중인 동안 리전 B의 해당 항목이 업데이트를 수신하는 경우, 최종 쓰기 우선 모델에서는 전파가 억제됩니다. 리전 B에서의 모든 업데이트는 수신되는 쓰기 요청을 억제할 수 있습니다.

모든 지역에 쓰기 모드를 사용하면 지역 A의 항목이 결국 지역 B로 전파될 것이라고 믿고 지역 A가 다시 온라인 상태가 될 때까지 항목이 누락될 가능성을 인식하여 지역 B에서 읽기 및 쓰기 작업을 계속할 수 있습니다. 중복 쓰기 작업과 같이 가능하면 최근 쓰기 트래픽을 재생 (예: 업스트림 이벤트 소스 사용) 하여 잠재적으로 누락될 수 있는 쓰기 작업의 간격을 메우고 마지막 작성자가 충돌을 해결하여 수신되는 쓰기 작업의 최종 전파를 방지할 수 있도록 하는 것이 좋습니다.

다른 쓰기 모드에서는 세상을 약간 out-of-date 바라보면서 작업을 계속할 수 있는 정도를 고려해야 합니다. ReplicationLatency로 추적되는 짧은 기간 동안의 일부 쓰기 작업은 리전 A가 다시 온라인 상태가 될 때까지 누락됩니다. 비즈니스를 계속 진행할 수 있을까요? 진행 가능한 사용 사례도 있겠지만 추가 완화 메커니즘 없이는 가능하지 않을 수도 있습니다.

예를 들어 리전이 완전히 중단된 후에도 중단 없이 사용 가능한 크레딧 밸런스를 유지해야 한다고 가정해 보겠습니다. 잔액을 지역 A와 지역 B에 각각 하나씩 두 개의 다른 아이টে으로 나누어 사용 가능한 잔고의 절반으로 각각 시작할 수 있습니다. 이렇게 하면 사용자 리전에 쓰기 모드를 사용하는 것입니다. 각 리전에서 처리되는 트랜잭션 업데이트는 잔액의 로컬 사본에 기록됩니다. 리전 A가 완전히 오프라인 상태가 되더라도 리전 B에서 트랜잭션 처리를 계속 진행할 수 있으며, 쓰기 작업은 리전 B에 보관된 잔액 부분으로만 제한됩니다. 이렇게 잔액을 분할하면 잔액이 낮아지거나 크레딧을 재조정해야 할 때 복잡성이 발생하지만, 보류 중인 쓰기 작업에 불확실성이 있더라도 비즈니스를 안전하게 복구할 수 있는 한 가지 예가 됩니다.

또 다른 예로, 웹 양식 데이터를 캡처한다고 가정해 보겠습니다. [OCC \(낙관적 동시성 제어\)](#) 를 사용하여 데이터 항목에 버전을 할당하고 최신 버전을 웹 양식에 숨겨진 필드로 포함할 수 있습니다. 제출할 때마다 데이터베이스에 있는 버전이 양식의 작성 기준 버전과 일치하는 경우에만 쓰기 작업이 성공합니다. 버전이 일치하지 않는 경우 데이터베이스에 있는 현재 버전을 기반으로 웹 양식을 새로 고치거나 신중하게 병합할 수 있고, 사용자는 다시 진행할 수 있습니다. OCC 모델은 일반적으로 다른 클라이언트가 데이터를 덮어쓰고 새 버전의 데이터를 생성하지 못하도록 보호하지만, 클라이언트가 이전 버전의 데이터를 발견할 수 있는 장애 조치 중에도 도움이 될 수 있습니다. 타임스탬프를 버전으로 사용하고 있다고 가정해 보겠습니다. 이 양식은 12:00 에 지역 A를 대상으로 처음 작성되었지만 (장애 조치 이후) 지역 B에 쓰려고 시도한 결과 데이터베이스의 최신 버전이 11:59 라는 메시지가 표시됩니다. 이 시나리오에서 클라이언트는 12:00 버전이 리전 B로 전파될 때까지 기다린 다음 이 버전을 기반으로 쓰거나, 11:59를 기반으로 빌드하고 새 12:01 버전(쓰기 후에 리전 A가 복구된 후 수신 버전을 억제)을 생성할 수 있습니다.

세 번째 예로, 금융 서비스 회사는 DynamoDB 데이터베이스에 고객 계정 및 금융 거래에 대한 데이터를 보관합니다. 지역 A가 완전히 중단되는 경우 해당 계정과 관련된 모든 쓰기 활동이 지역 B에서 완전히 사용 가능한지 확인하거나 지역 A가 다시 온라인 상태가 될 때까지 계정을 부분적 상태로 격리하려

고 합니다. 이 회사는 모든 업무를 일시 중지하는 대신 트랜잭션이 전파되지 않은 것으로 판단되는 극히 일부의 계정만 업무를 일시 중지하기로 결정했습니다. 이를 위해 리전 C라고 부르는 세 번째 리전을 사용했습니다. 리전 A에서 쓰기 작업을 처리하기 전에 보류 중인 작업(예: 계정의 새 트랜잭션 수)을 간략하게 요약하여 리전 C에 배치했습니다. 이 요약만으로도 리전 B가 해당 뷰가 최신 상태인지 판단하기에 충분했습니다. 이 조치로 인해 리전 C에서의 쓰기 시점부터 리전 A가 쓰기 작업을 수락하고 지역 B가 쓰기 작업을 수신할 때까지 계정이 사실상 잠겼습니다. 리전 C에 있는 데이터는 장애 조치 프로세스의 일부인 경우를 제외하고는 사용되지 않았습니다. 장애 조치 후 리전 B는 리전 C와 데이터를 교환 검증하여 최신 상태가 아닌 계정이 있는지 확인할 수 있었습니다. 해당 계정은 지역 A 복구 시 일부 데이터가 지역 B로 전파될 때까지 격리된 것으로 표시되었습니다. 지역 C에 장애가 발생할 경우 새 지역 D를 가동하여 대신 사용할 수 있습니다. 지역 C의 데이터는 매우 일시적이었으며 몇 분 후 지역 D에는 진행 중인 쓰기 작업에 대한 충분한 up-to-date 기록이 남았으므로 충분히 유용하게 사용할 수 있었습니다. 리전 B에 장애가 발생할 경우 리전 A는 리전 C와 협력하여 쓰기 요청을 계속 수락할 수 있었습니다. 이 회사는 지연 시간이 더 긴 쓰기(리전 C와 리전 A에 대한 쓰기)를 받아들일 용의가 있었고, 다행히도 계정 상태를 간략하게 요약할 수 있는 데이터 모델이 있었습니다.

## 글로벌 테이블의 처리량 용량 계획

한 리전에서 다른 리전으로 트래픽을 마이그레이션하려면 용량과 관련된 DynamoDB 테이블 설정을 신중하게 고려해야 합니다.

쓰기 용량 관리에 대한 몇 가지 고려 사항은 다음과 같습니다.

- 글로벌 테이블은 온디맨드 모드이거나 Auto Scaling이 활성화된 상태로 프로비저닝되어야 합니다.
- Auto Scaling으로 프로비저닝된 경우 쓰기 설정(최소, 최대, 목표 사용률)이 여러 리전에 복제됩니다. Auto Scaling 설정은 동기화되지만 실제로 프로비저닝되는 쓰기 용량은 리전 간에 독립적으로 변동될 수 있습니다.
- 프로비저닝된 쓰기 용량이 다르게 나타나는 한 가지 이유는 time to live (TTL) 기능 때문입니다. TTL DynamoDB에서 활성화하면 값이 항목의 만료 시간을 나타내는 속성 이름을 [Unix epoch](#) 시간 형식 (초) 으로 지정할 수 있습니다. 이 시간이 지나면 DynamoDB가 쓰기 비용 없이 항목을 삭제할 수 있습니다. 글로벌 테이블을 사용하면 어느 TTL 지역에서나 구성할 수 있으며, 해당 설정은 글로벌 테이블과 연결된 다른 지역에 자동으로 복제됩니다. TTL 규칙을 통해 항목을 삭제할 수 있는 경우 해당 작업은 어느 지역에서나 수행할 수 있습니다. 삭제 작업은 원본 테이블의 쓰기 단위를 소비하지 않고 수행되지만 복제 테이블에는 해당 삭제 작업에 대한 복제 쓰기가 발생하며 복제된 쓰기 단위 비용이 발생합니다.
- Auto Scaling을 사용하는 경우 프로비저닝된 최대 쓰기 용량 설정이 모든 쓰기 작업과 모든 잠재적 TTL 삭제 작업을 처리할 수 있을 만큼 충분히 높은지 확인하십시오. Auto Scaling은 쓰기 사용량에 따라 각 리전을 조정합니다. 온디맨드 테이블에는 프로비저닝된 최대 쓰기 용량 설정이 없지만 테이블 수준 최대 쓰기 처리량 제한이 온디맨드 테이블이 허용하는 최대 지속 쓰기 용량을 지정합니다. 기본 제한은 40,000이지만 조정할 수 있습니다. 온디맨드 테이블에 필요할 수 있는 모든 쓰기 작업 (쓰기 작업 포함 TTL) 을 처리할 수 있을 만큼 충분히 높게 설정하는 것이 좋습니다. 글로벌 테이블을 설정할 때는 모든 참여 리전에서 이 값이 동일해야 합니다.

읽기 용량 관리에 대한 몇 가지 고려 사항은 다음과 같습니다.

- 리전마다 읽기 패턴이 독립적일 수 있다고 가정하기 때문에 읽기 용량 관리 설정이 리전마다 다를 수 있습니다. 테이블에 처음 글로벌 복제본을 추가하면 소스 리전의 용량이 전파됩니다. 생성 후에 읽기 용량 설정을 조정할 수 있으며, 이 설정은 반대쪽으로 전송되지 않습니다.
- DynamoDB Auto Scaling을 사용할 때는 프로비저닝된 최대 읽기 용량 설정이 모든 리전에서 모든 읽기 작업을 처리할 수 있을 만큼 충분히 높아야 합니다. 표준 운영 중에는 읽기 용량이 여러 지역에 분산될 수 있지만 장애 조치 중에는 테이블이 증가된 읽기 워크로드에 맞게 자동으로 조정될 수 있어야 합니다. 온디맨드 테이블에는 프로비저닝된 최대 읽기 용량 설정이 없지만 테이블 수준 최대 읽기

처리량 제한이 온디맨드 테이블이 허용하는 최대 지속 읽기 용량을 지정합니다. 기본 제한은 40,000 이지만 조정할 수 있습니다. 모든 읽기 작업이 이 단일 리전으로 라우팅되는 경우 테이블에 필요할 수 있는 모든 읽기 작업을 처리할 수 있을 만큼 이 제한을 충분히 높게 설정하는 것이 좋습니다.

- 한 리전의 테이블이 일반적으로 읽기 트래픽을 수신하지 못하지만 장애 조치 후 많은 양의 읽기 트래픽을 흡수해야 하는 경우, 테이블의 프로비저닝된 읽기 용량을 늘리고 테이블 업데이트가 완료될 때까지 기다린 다음 테이블을 다시 프로비저닝할 수 있습니다. 테이블을 프로비저닝된 모드로 두거나 온디맨드 모드로 전환할 수 있습니다. 이렇게 하면 테이블이 사전 워밍되어 더 높은 수준의 읽기 트래픽을 수용할 수 있습니다.

ARC요청을 라우팅하는 데 Route 53을 사용하는지 여부에 관계없이 DynamoDB 리전의 테이블 설정 및 계정 할당량이 비슷한지 확인하는 데 유용할 수 있는 [준비 검사](#)가 있습니다. 이러한 준비 상태 검사는 계정 수준 할당량을 조정하여 일치하도록 하는 데도 도움이 됩니다.

# 글로벌 테이블 준비 체크리스트

글로벌 테이블을 배포할 때 의사 결정 및 작업에 다음 체크리스트를 사용하세요.

- 글로벌 테이블에 참여해야 하는 리전과 리전 수를 결정합니다.
- [애플리케이션의 쓰기 모드를 결정하십시오.](#)
- 쓰기 모드에 따라 [라우팅 전략](#)을 계획하세요.
- 쓰기 모드와 라우팅 전략에 따라 [대피 계획](#)을 정의하세요.
- 각 리전의 상태, 지연 시간, 오류에 대한 지표를 캡처합니다. DynamoDB 메트릭 목록은 운영 인식을 위한 [Amazon DynamoDB 모니터링](#) 블로그 게시물을 참조하십시오 AWS . 또한 [가상 카나리아](#) (장애를 감지하도록 설계된 인위적인 요청)와 고객 트래픽의 실시간 관찰을 사용해야 합니다. 모든 문제가 DynamoDB 지표에 나타나는 것은 아닙니다.
- ReplicationLatency의 지속적 증가에 대한 경보를 설정하세요. 증가는 글로벌 테이블의 쓰기 설정이 리전마다 다른 잘못된 구성을 나타낼 수 있습니다. 이는 복제된 요청 실패와 지연 시간 증가로 이어질 수 있습니다. 리전 중단이 있음을 나타낼 수도 있습니다. [좋은 예](#)는 최근 평균이 180,000밀리초를 초과할 경우 알림을 생성하는 것입니다. ReplicationLatency가 0으로 떨어지는 것을 관찰할 수도 있습니다. 이는 복제가 중단되었음을 나타냅니다.
- 각 글로벌 테이블에 충분한 최대 읽기 및 쓰기 설정을 할당합니다.
- 지역에서 대피해야 할 조건을 파악하십시오. 결정에 사람의 판단이 수반되는 경우 모든 고려 사항을 문서화합니다. 이 작업은 압박을 받지 않는 상태에서 사전에 신중하게 수행해야 합니다.
- 리전 대피 시 취해야 하는 모든 조치를 위한 런북을 유지 관리합니다. 일반적으로 글로벌 테이블에 필요한 작업은 거의 없지만 나머지 스택을 이동하는 작업은 복잡할 수 있습니다.

## Note

장애 조치 절차를 사용할 때는 데이터 플레인 작업에만 의존하고 컨트롤 플레인 작업에는 의존하지 않는 것이 좋습니다. 지역 장애 발생 시 일부 컨트롤 플레인 작업이 저하될 수 있기 때문입니다. 자세한 내용은 AWS 블로그 게시물 [Amazon DynamoDB 글로벌 테이블을 사용하여 복원력이 뛰어난 애플리케이션 구축](#): 4부를 참조하십시오.

- 리전 대피를 포함하여 런북의 모든 측면을 정기적으로 테스트합니다. 테스트되지 않은 런북은 신뢰할 수 없는 런북입니다.
- 전체 애플리케이션 (글로벌 테이블 포함)의 복원력을 평가하는 [AWS Resilience Hub](#)에 사용하는 것을 고려해 보십시오. 이 서비스는 대시보드를 통해 애플리케이션 포트폴리오의 복원력 상태를 포괄적으로 보여줍니다.

- [ARC](#) 준비 상태 검사를 사용하여 애플리케이션의 현재 구성을 평가하고 모범 사례와의 편차를 추적해 보십시오.
- Route 53 또는 글로벌 액셀러레이터와 함께 사용할 상태 확인을 작성할 때는 전체 데이터베이스 흐름을 포괄하는 일련의 호출을 수행하십시오. DynamoDB 엔드포인트가 작동하는지만 확인하도록 검사를 제한하면 IAM () 구성 오류, 코드 배포 문제, DynamoDB 외부 스택에서의 장애, 평균보다 긴 읽기 또는 쓰기 지연 시간 등과 AWS Identity and Access Management 같은 여러 장애 모드를 해결할 수 없습니다.

## 글로벌 테이블 FAQ

이 섹션에서는 DynamoDB 글로벌 테이블에 대한 질문과 답을 제공합니다.

### 글로벌 테이블의 요금은 어떻게 되나요?

- 기존 DynamoDB 테이블에 대한 쓰기 작업 요금은 쓰기 용량 단위(WCU, 프로비저닝된 테이블의 경우) 또는 쓰기 요청 단위(WRU, 온디맨드 테이블의 경우)를 기준으로 합니다. 5KB 항목을 쓰면 5단위의 요금이 부과됩니다. 글로벌 테이블에 대한 쓰기 요금은 복제된 쓰기 용량 단위(rWCU, 프로비저닝된 테이블의 경우) 또는 복제된 쓰기 요청 단위(rWRU, 온디맨드 테이블의 경우)로 책정됩니다.
- rWCU와 rWRU에는 복제를 관리하는 데 필요한 스트리밍 인프라 비용이 포함됩니다. 따라서 WCU 및 WRU보다 50% 더 높은 요금이 책정됩니다. 리전 간 데이터 전송 요금이 적용됩니다.
- RWCu 및 RWRu 요금은 항목이 직접 작성되거나 복제를 통해 작성된 모든 지역에서 발생합니다.
- 글로벌 보조 인덱스(GSI)에 대한 쓰기 작업은 로컬 쓰기 작업으로 간주되며, 일반 쓰기 단위를 사용합니다.
- 현재 rWCU에는 예약 용량을 사용할 수 없습니다. GSI가 쓰기 단위를 소비하는 테이블의 경우 WCU에 대해 예약 용량을 구매하는 것이 여전히 유리할 수 있습니다.
- 글로벌 테이블에 새 리전을 추가하면 DynamoDB는 새 리전을 자동으로 부트스트랩하고, 테이블 복원과 마찬가지로 테이블의 GB 크기를 기준으로 요금을 청구합니다. 리전 간 데이터 전송 요금도 적용됩니다.

### 글로벌 테이블은 어떤 리전을 지원하나요?

글로벌 테이블은 모든 AWS 리전을 지원합니다.

### 글로벌 테이블에서는 GSI를 어떻게 처리하나요?

글로벌 테이블(현재 버전 2019)에서는 한 리전에서 생성한 GSI가 다른 참여 리전에도 자동으로 생성되고 자동으로 백업됩니다.

### 글로벌 테이블의 복제를 중지하려면 어떻게 해야 하나요?

다른 테이블을 삭제하는 것과 같은 방식으로 복제본 테이블을 삭제할 수 있습니다. 글로벌 테이블을 삭제하면 해당 리전으로의 복제가 중지되고 해당 리전에 보관된 테이블 사본이 삭제됩니다. 하지만 테이블

블의 사본을 독립적 엔터티로 유지하는 동안에는 복제를 중지할 수 없으며 복제를 일시 중지할 수도 없습니다.

## Amazon DynamoDB Streams는 글로벌 테이블과 어떻게 상호 작용 하나요?

각 글로벌 테이블은 시작 위치에 관계없이 모든 쓰기 작업을 기반으로 독립적인 스트림을 생성합니다. 이 DynamoDB 스트림을 한 리전 또는 모든 리전에서 독립적으로 사용하도록 선택할 수 있습니다. 로컬 쓰기 작업은 처리되 복제된 쓰기 작업은 처리하지 않으려는 경우 쓰기 리전을 식별하는 고유한 리전 속성을 각 항목에 추가할 수 있습니다. 그런 다음 AWS Lambda 이벤트 필터를 사용하여 로컬 리전에서 쓰기 작업만을 위한 Lambda 함수를 직접적으로 호출할 수 있습니다. 이렇게 하면 삽입 및 업데이트 작업에 도움이 되지만 삭제 작업에는 도움이 되지 않습니다.

## 글로벌 테이블은 트랜잭션을 어떻게 처리 하나요?

트랜잭션 작업은 쓰기 작업이 원래 이루어진 리전에서만 ACID(원자성, 일관성, 격리 및 내구성) 보장을 제공합니다. 전역 테이블에서는 트랜잭션이 리전 간에 지원되지 않습니다. 예를 들어 미국 동부(오하이오) 및 미국 서부(오레곤) 리전에 복제본이 있는 글로벌 테이블이 있고 미국 동부(오하이오)에서 TransactWriteItems 작업을 수행할 경우, 변경 내용이 복제될 때 미국 서부(오레곤)에서 부분적으로 완료된 트랜잭션을 관찰할 수 있습니다. 변경 사항은 소스 리전에서 커밋된 이후에만 다른 리전에 복제됩니다.

## 글로벌 테이블은 DynamoDB Accelerator(DAX) 캐시와 어떻게 상호 작용 하나요?

글로벌 테이블은 DynamoDB를 직접 업데이트하여 DAX를 우회하므로 DAX는 오래된 데이터를 보유하고 있다는 사실을 인식하지 못합니다. DAX 캐시는 캐시의 TTL이 만료되는 경우에만 새로 고쳐집니다.

## 테이블의 태그가 전파되나요?

아니요, 태그는 자동으로 전파되지 않습니다.

## 테이블을 모든 리전에 백업해야 하나요 아니면 한 리전에만 백업해야 하나요?

대답은 백업의 목적에 따라 달라집니다.

- 데이터 내구성을 보장하려는 경우 DynamoDB는 이미 그러한 보호 장치를 제공합니다. 이 서비스는 내구성을 보장합니다.
- 기록 레코드의 스냅샷을 보관하려는 경우(예: 규정 요구 사항 충족) 한 리전에 백업하는 것으로 충분합니다. [AWS Backup](#)를 사용하여 백업을 추가 리전에 복사할 수 있습니다.
- 잘못 삭제되거나 수정된 데이터를 복구하려면 한 지역에서 [DynamoDB point-in-time 복구 \(PITR\)](#)를 사용하십시오.

## AWS CloudFormation을 사용하여 글로벌 테이블을 배포하려면 어떻게 해야 하나요?

- CloudFormation DynamoDB 테이블과 글로벌 테이블을 별도의 두 리소스인 `Table` 및 `GlobalTable`로 나타냅니다. `AWS::DynamoDB::Table`와 `AWS::DynamoDB::GlobalTable` 한 가지 방법은 `GlobalTable` 구성을 사용하여 글로벌 테이블일 수 있는 모든 테이블을 생성하고, 처음에는 해당 테이블을 독립형 테이블로 유지하다가 나중에 필요에 따라 리전을 추가하는 것입니다.
- 여기서 CloudFormation 각 글로벌 테이블은 복제본 수에 관계없이 단일 리전의 단일 스택으로 제어됩니다. 템플릿을 배포할 때 단일 스택 작업의 일부로 모든 복제본을 CloudFormation 생성하고 업데이트합니다. 동일한 `AWS::DynamoDB::GlobalTable` 리소스를 여러 리전에 배포하면 안 됩니다. 이런 배포는 오류를 유발하며 지원되지 않습니다. 여러 리전에 애플리케이션 템플릿을 배포하는 경우 조건을 사용하여 단일 리전에서 `AWS::DynamoDB::GlobalTable` 리소스를 생성할 수 있습니다. 또는 `AWS::DynamoDB::GlobalTable` 리소스를 애플리케이션 스택과 별도의 스택에 정의하고 단일 리전에 배포되도록 선택할 수 있습니다.
- 일반 테이블을 사용하여 관리하면서 글로벌 테이블로 변환하려는 경우 CloudFormation: [삭제 정책](#)을 설정하고 스택에서 테이블을 제거한 다음 콘솔에서 테이블을 글로벌 테이블로 변환한 다음 글로벌 테이블을 스택에 새 리소스로 가져오십시오. Retain 자세한 내용은 AWS GitHub 리포지토리 [amazon-dynamodb-table-to-global-table-cdk](#)를 참조하십시오.
- 교차 계정 복제는 현재 지원되지 않습니다.

## 결론 및 리소스

DynamoDB 글로벌 테이블에는 컨트롤이 거의 없지만 여전히 신중하게 고려해야 합니다. 쓰기 모드, 라우팅 모델, 대피 프로세스를 결정해야 합니다. 글로벌 상태 유지를 위해서는 모든 리전에서 애플리케이션을 계측하고 경로를 조정하거나 대피를 수행할 준비가 되어 있어야 합니다. 그 보상은 99.999% 가용성을 제공하도록 설계된 지연 시간이 짧은 읽기 및 쓰기 작업을 갖춘 전 세계에 분산된 데이터 세트를 갖게 된다는 것입니다.

DynamoDB 글로벌 테이블에 대한 자세한 내용은 다음 리소스를 참조하십시오.

- [Amazon DynamoDB 설명서](#)
- [Amazon Route 53 애플리케이션 복구 컨트롤러](#)
- [ARC준비 검사 \(설명서\)AWS](#)
- [Route 53 라우팅 정책 \(AWS 설명서\)](#)
- [AWS Global Accelerator](#)
- [DynamoDB 서비스 수준 계약](#)
- [AWS 다중 지역 기초 \(백서\)AWS](#)
- [AWS\(AWS re:Invent 2022 프레젠테이션\) 을 사용한 데이터 레질리언스 설계 패턴](#)
- [Amazon DynamoDB로 피델리티 인베스트먼트와 리티오를 현대화한 방법 \(re:Invent 2022 프레젠테이션\)AWS](#)
- [다중 지역 설계 패턴 및 모범 사례 \(re:Invent 2022 프레젠테이션\)AWS](#)
- [재해 복구 \(DR\) 아키텍처 기반 AWS, 일부III: 파일럿 라이트 및 워 스탠바이 \(블로그 게시물\)AWS](#)
- [지역 고정을 사용하여 Amazon DynamoDB 글로벌AWS 테이블의 항목에 홈 지역을 설정합니다 \(블로그 게시물\).](#)
- [운영 인식을AWS 위한 Amazon DynamoDB 모니터링 \(블로그 게시물\)](#)
- [DynamoDB 크기 조정: 열에 따른 파티션, 핫키 및 분할이 성능에AWS 미치는 영향 \(블로그 게시물\)](#)

## 문서 기록

아래 표에 이 가이드의 주요 변경 사항이 설명되어 있습니다. 향후 업데이트에 대한 알림을 받으려면 [RSS 피드](#)를 구독하십시오.

변경 사항	설명	날짜
<a href="#">업데이트된 AWS Global Accelerator 정보</a>	<a href="#">글로벌 액셀러레이터</a> 요청 라우팅의 엔드포인트를 수정했습니다.	2024년 3월 14일
<a href="#">업데이트된 AWS 리전 지원 정보</a>	이제 글로벌 테이블에서 모든 AWS 리전을 지원한다는 사실을 반영하도록 <a href="#">FAQ</a> 를 업데이트했습니다.	2023년 11월 15일
<a href="#">최초 게시</a>	—	2023년 5월 19일

# AWS 규범적 지침 용어집

다음은 규범적 지침에서 제공하는 AWS 전략, 가이드 및 패턴에서 일반적으로 사용되는 용어입니다. 용어집 항목을 제안하려면 용어집 끝에 있는 피드백 제공 링크를 사용하십시오.

## 숫자

### 7가지 전략

애플리케이션을 클라우드로 이전하기 위한 7가지 일반적인 마이그레이션 전략 이러한 전략은 Gartner가 2011년에 파악한 5가지 전략을 기반으로 하며 다음으로 구성됩니다.

- 리팩터링/리아키텍트 - 클라우드 네이티브 기능을 최대한 활용하여 애플리케이션을 이동하고 해당 아키텍처를 수정함으로써 민첩성, 성능 및 확장성을 개선합니다. 여기에는 일반적으로 운영 체제와 데이터베이스 이식이 포함됩니다. 예: 온프레미스 Oracle 데이터베이스를 Amazon Aurora PostgreSQL 호환 에디션으로 마이그레이션하십시오.
- 리플랫폼(리프트 앤드 리세이프) - 애플리케이션을 클라우드로 이동하고 일정 수준의 최적화를 도입하여 클라우드 기능을 활용합니다. 예: 온프레미스 Oracle 데이터베이스를 오라클용 Amazon RDS (Amazon RDS) 로 마이그레이션합니다. AWS 클라우드
- 재구매(드롭 앤드 쇼프) - 일반적으로 기존 라이선스에서 SaaS 모델로 전환하여 다른 제품으로 전환합니다. 예: 고객 관계 관리 (CRM) 시스템을 Salesforce.com으로 마이그레이션하십시오.
- 리호스팅(리프트 앤드 시프트) - 애플리케이션을 변경하지 않고 클라우드로 이동하여 클라우드 기능을 활용합니다. 예: 온프레미스 Oracle 데이터베이스를 의 EC2 인스턴스에서 Oracle로 마이그레이션합니다. AWS 클라우드
- 재배포(하이퍼바이저 수준의 리프트 앤 시프트) - 새 하드웨어를 구매하거나, 애플리케이션을 다시 작성하거나, 기존 운영을 수정하지 않고도 인프라를 클라우드로 이동합니다. 온프레미스 플랫폼에서 동일한 플랫폼의 클라우드 서비스로 서버를 마이그레이션합니다. 예: Microsoft Hyper-V 애플리케이션을 다음으로 마이그레이션하십시오. AWS
- 유지(보관) - 소스 환경에 애플리케이션을 유지합니다. 대규모 리팩터링이 필요하고 해당 작업을 나중에 연기하려는 애플리케이션과 비즈니스 차원에서 마이그레이션할 이유가 없어 유지하려는 레거시 애플리케이션이 여기에 포함될 수 있습니다.
- 사용 중지 - 소스 환경에서 더 이상 필요하지 않은 애플리케이션을 폐기하거나 제거합니다.

# A

## ABAC

[속성 기반 액세스](#) 제어를 참조하십시오.

## 추상화된 서비스

[관리형 서비스를](#) 참조하십시오.

## 산

[원자성, 일관성, 격리성, 내구성을](#) 참조하십시오.

## 능동-능동 마이그레이션

양방향 복제 도구 또는 이중 쓰기 작업을 사용하여 소스 데이터베이스와 대상 데이터베이스가 동기화된 상태로 유지되고, 두 데이터베이스 모두 마이그레이션 중 연결 애플리케이션의 트랜잭션을 처리하는 데이터베이스 마이그레이션 방법입니다. 이 방법은 일회성 전환이 필요한 대신 소규모의 제어된 배치로 마이그레이션을 지원합니다. [더 유연하지만 액티브-패시브 마이그레이션보다 더 많은 작업이 필요합니다.](#)

## 능동-수동 마이그레이션

소스 데이터베이스와 대상 데이터베이스가 동기화된 상태로 유지되지만 소스 데이터베이스만 연결 애플리케이션의 트랜잭션을 처리하고 데이터는 대상 데이터베이스로 복제되는 데이터베이스 마이그레이션 방법입니다. 대상 데이터베이스는 마이그레이션 중 어떤 트랜잭션도 허용하지 않습니다.

## 집계 함수

행 그룹에서 연산을 수행하고 그룹에 대한 단일 반환값을 계산하는 SQL 함수입니다. 집계 함수의 예로는 `MAX` 및 `SUM` 등이 있습니다.

## AI

[인공 지능을](#) 참조하십시오.

## AIOps

[인공 지능 운영을](#) 참조하십시오.

## 익명화

데이터세트에서 개인 정보를 영구적으로 삭제하는 프로세스입니다. 익명화는 개인 정보 보호에 도움이 될 수 있습니다. 익명화된 데이터는 더 이상 개인 데이터로 간주되지 않습니다.

## 안티 패턴

솔루션이 다른 솔루션보다 비생산적이거나 비효율적이거나 덜 효과적이어서 반복되는 문제에 자주 사용되는 솔루션입니다.

### 애플리케이션 제어

시스템을 멀웨어로부터 보호하기 위해 승인된 애플리케이션만 사용할 수 있는 보안 접근 방식입니다.

### 애플리케이션 포트폴리오

애플리케이션 구축 및 유지 관리 비용과 애플리케이션의 비즈니스 가치를 비롯하여 조직에서 사용하는 각 애플리케이션에 대한 세부 정보 모음입니다. 이 정보는 [포트폴리오 검색 및 분석 프로세스](#)의 핵심이며 마이그레이션, 현대화 및 최적화할 애플리케이션을 식별하고 우선순위를 정하는 데 도움이 됩니다.

### 인공 지능

컴퓨터 기술을 사용하여 학습, 문제 해결, 패턴 인식 등 일반적으로 인간과 관련된 인지 기능을 수행하는 것을 전문으로 하는 컴퓨터 과학 분야입니다. 자세한 내용은 [What is Artificial Intelligence?](#)를 참조하십시오.

### 인공 지능 운영(AIOps)

기계 학습 기법을 사용하여 운영 문제를 해결하고, 운영 인시던트 및 사용자 개입을 줄이고, 서비스 품질을 높이는 프로세스입니다. AWS 마이그레이션 전략에서 AIOps가 사용되는 방법에 대한 자세한 내용은 [운영 통합 가이드](#)를 참조하십시오.

### 비대칭 암호화

한 쌍의 키, 즉 암호화를 위한 퍼블릭 키와 복호화를 위한 프라이빗 키를 사용하는 암호화 알고리즘입니다. 퍼블릭 키는 복호화에 사용되지 않으므로 공유할 수 있지만 프라이빗 키에 대한 액세스는 엄격히 제한되어야 합니다.

### 원자성, 일관성, 격리성, 내구성(ACID)

오류, 정전 또는 기타 문제가 발생한 경우에도 데이터베이스의 데이터 유효성과 운영 신뢰성을 보장하는 소프트웨어 속성 세트입니다.

### ABAC(속성 기반 액세스 제어)

부서, 직무, 팀 이름 등의 사용자 속성을 기반으로 세분화된 권한을 생성하는 방식입니다. 자세한 내용은 AWS Identity and Access Management (IAM) [설명서의 AWS ABAC](#) for를 참조하십시오.

## 신뢰할 수 있는 데이터 소스

가장 신뢰할 수 있는 정보 소스로 간주되는 기본 버전의 데이터를 저장하는 위치입니다. 익명화, 편집 또는 가명화와 같은 데이터 처리 또는 수정의 목적으로 신뢰할 수 있는 데이터 소스의 데이터를 다른 위치로 복사할 수 있습니다.

### 가용 영역

다른 가용 영역의 장애로부터 격리되고 동일한 지역 내 다른 가용 영역에 저렴하고 지연 시간이 짧은 네트워크 연결을 제공하는 별도의 위치. AWS 리전

### AWS 클라우드 채택 프레임워크 (AWS CAF)

조직이 클라우드로 성공적으로 AWS 전환하기 위한 효율적이고 효과적인 계획을 개발하는 데 도움이 되는 지침 및 모범 사례 프레임워크입니다. AWS CAF는 지침을 관점이라고 하는 6가지 중점 영역, 즉 비즈니스, 사람, 거버넌스, 플랫폼, 보안, 운영으로 분류합니다. 비즈니스, 사람 및 거버넌스 관점은 비즈니스 기술과 프로세스에 초점을 맞추고, 플랫폼, 보안 및 운영 관점은 전문 기술과 프로세스에 중점을 둡니다. 예를 들어, 사람 관점은 인사(HR), 직원 배치 기능 및 인력 관리를 담당하는 이해관계자를 대상으로 합니다. 이러한 관점에서 AWS CAF는 조직이 성공적인 클라우드 채택을 준비할 수 있도록 인력 개발, 교육 및 커뮤니케이션에 대한 지침을 제공합니다. 자세한 내용은 [AWS CAF 웹 사이트](#)와 [AWS CAF 백서](#)를 참조하십시오.

### AWS 워크로드 검증 프레임워크 (AWS WQF)

데이터베이스 마이그레이션 워크로드를 평가하고 마이그레이션 전략을 권장하며 작업 예상치를 제공하는 도구입니다. AWS WQF는 () 에 포함됩니다. AWS Schema Conversion Tool AWS SCT 데이터베이스 스키마 및 코드 객체, 애플리케이션 코드, 종속성 및 성능 특성을 분석하고 평가 보고서를 제공합니다.

## B

### 배드 봇

개인이나 조직을 방해하거나 피해를 입히려는 의도를 가진 [봇입니다](#).

### BCP

[비즈니스 연속성 계획을](#) 참조하십시오.

## 동작 그래프

리소스 동작과 시간 경과에 따른 상호 작용에 대한 통합된 대화형 뷰입니다. Amazon Detective에서 동작 그래프를 사용하여 실패한 로그인 시도, 의심스러운 API 호출 및 유사한 작업을 검사할 수 있습니다. 자세한 내용은 Detective 설명서의 [Data in a behavior graph](#)를 참조하십시오.

## 빅 엔디안 시스템

가장 중요한 바이트를 먼저 저장하는 시스템입니다. [엔디안도](#) 참조하십시오.

## 바이너리 분류

바이너리 결과(가능한 두 클래스 중 하나)를 예측하는 프로세스입니다. 예를 들어, ML 모델이 “이 이메일이 스팸인가요, 스팸이 아닌가요?”, ‘이 제품은 책임가요, 자동차인가요?’ 등의 문제를 예측해야 할 수 있습니다.

## 블룸 필터

요소가 세트의 멤버인지 여부를 테스트하는 데 사용되는 메모리 효율성이 높은 확률론적 데이터 구조입니다.

## 블루/그린(Blue/Green) 배포

서로 다르지만 동일한 환경을 두 개 만드는 배포 전략입니다. 현재 애플리케이션 버전을 한 환경 (파란색) 에서 실행하고 다른 환경 (녹색) 에서 새 애플리케이션 버전을 실행합니다. 이 전략을 사용하면 영향을 최소화하면서 신속하게 롤백할 수 있습니다.

## bot

인터넷을 통해 자동화된 작업을 실행하고 사람의 활동이나 상호 작용을 시뮬레이션하는 소프트웨어 애플리케이션입니다. 인터넷에서 정보를 인덱싱하는 웹 크롤러와 같은 일부 봇은 유용하거나 유용합니다. 배드 봇으로 알려진 일부 다른 봇은 개인이나 조직을 방해하거나 피해를 입히기 위한 것입니다.

## 봇넷

[멀웨어에 감염되어 봇 허더 또는 봇 운영자로 알려진 단일 당사자의 통제 하에 있는 봇 네트워크](#). 봇넷은 봇과 그 영향을 확장하는 가장 잘 알려진 메커니즘입니다.

## 브랜치

코드 리포지토리의 포함된 영역입니다. 리포지토리에 생성되는 첫 번째 브랜치가 기본 브랜치입니다. 기존 브랜치에서 새 브랜치를 생성한 다음 새 브랜치에서 기능을 개발하거나 버그를 수정할 수 있습니다. 기능을 구축하기 위해 생성하는 브랜치를 일반적으로 기능 브랜치라고 합니다. 기능을 출시할 준비가 되면 기능 브랜치를 기본 브랜치에 다시 병합합니다. 자세한 내용은 [브랜치 정보](#) (문서) 를 참조하십시오. GitHub

## 브레이크 글래스 액세스

예외적인 상황에서 승인된 프로세스를 통해 사용자가 일반적으로 액세스 권한이 없는 데이터에 빠르게 액세스할 수 있는 AWS 계정 있는 수단입니다. 자세한 내용은 Well-Architected AWS 지침의 [브레이크 글래스 절차 구현](#) 표시기를 참조하십시오.

## 브라운필드 전략

사용자 환경의 기존 인프라 시스템 아키텍처에 브라운필드 전략을 채택할 때는 현재 시스템 및 인프라의 제약 조건을 중심으로 아키텍처를 설계합니다. 기존 인프라를 확장하는 경우 브라운필드 전략과 [그린필드](#) 전략을 혼합할 수 있습니다.

## 버퍼 캐시

가장 자주 액세스하는 데이터가 저장되는 메모리 영역입니다.

## 사업 역량

기업이 가치를 창출하기 위해 하는 일(예: 영업, 고객 서비스 또는 마케팅)입니다. 마이크로서비스 아키텍처 및 개발 결정은 비즈니스 역량에 따라 이루어질 수 있습니다. 자세한 내용은 백서의 [AWS에서 컨테이너화된 마이크로서비스 실행의 비즈니스 역량 중심의 구성화](#) 섹션을 참조하십시오.

## 비즈니스 연속성 계획(BCP)

대규모 마이그레이션과 같은 중단 이벤트가 운영에 미치는 잠재적 영향을 해결하고 비즈니스가 신속하게 운영을 재개할 수 있도록 지원하는 계획입니다.

# C

## CAF

[클라우드 채택 프레임워크를 참조하십시오AWS](#).

## 카나리아 배포

최종 사용자에게 버전을 느리고 점진적으로 릴리스하는 것입니다. 확신이 들면 새 버전을 배포하고 현재 버전을 완전히 교체합니다.

## CCoE

[클라우드 센터 오브 엑셀런스를 참조하십시오](#).

## CDC

[변경 데이터 캡처를 참조하십시오](#).

## 변경 데이터 캡처(CDC)

데이터베이스 테이블과 같은 데이터 소스의 변경 내용을 추적하고 변경 사항에 대한 메타데이터를 기록하는 프로세스입니다. 대상 시스템의 변경 내용을 감사하거나 복제하여 동기화를 유지하는 등의 다양한 용도로 CDC를 사용할 수 있습니다.

## 카오스 엔지니어링

시스템의 복원력을 테스트하기 위해 의도적으로 장애나 장애를 일으키는 이벤트를 발생시키는 행위 [AWS Fault Injection Service \(AWS FIS\)](#) 를 사용하여 AWS 워크로드에 스트레스를 주는 실험을 수행하고 응답을 평가할 수 있습니다.

## CI/CD

[지속적 통합 및 지속적 전달](#)을 참조하십시오.

## 분류

예측을 생성하는 데 도움이 되는 분류 프로세스입니다. 분류 문제에 대한 ML 모델은 이산 값을 예측합니다. 이산 값은 항상 서로 다릅니다. 예를 들어, 모델이 이미지에 자동차가 있는지 여부를 평가해야 할 수 있습니다.

## 클라이언트측 암호화

대상이 데이터를 AWS 서비스 수신하기 전에 데이터를 로컬로 암호화합니다.

## 클라우드 혁신 센터(CCoE)

클라우드 모범 사례 개발, 리소스 동원, 마이그레이션 타임라인 설정, 대규모 혁신을 통한 조직 선도 등 조직 전체에서 클라우드 채택 노력을 추진하는 다분야 팀입니다. 자세한 내용은 AWS 클라우드 기업 전략 [블로그의 CCoE 게시물을](#) 참조하십시오.

## 클라우드 컴퓨팅

원격 데이터 스토리지와 IoT 디바이스 관리에 일반적으로 사용되는 클라우드 기술 클라우드 컴퓨팅은 일반적으로 [엣지 컴퓨팅 기술과](#) 연결됩니다.

## 클라우드 운영 모델

IT 조직에서 하나 이상의 클라우드 환경을 구축, 성숙화 및 최적화하는 데 사용되는 운영 모델입니다. 자세한 내용은 [클라우드 운영 모델 구축](#)을 참조하십시오.

## 클라우드 채택 단계

조직이 마이그레이션할 때 일반적으로 거치는 4단계는 다음과 같습니다. AWS 클라우드

- 프로젝트 - 개념 증명 및 학습 목적으로 몇 가지 클라우드 관련 프로젝트 실행
- 기반 - 클라우드 채택 확장을 위한 기초 투자(예: 랜딩 존 생성, CCoE 정의, 운영 모델 구축)
- 마이그레이션 - 개별 애플리케이션 마이그레이션
- Re-invention - 제품 및 서비스 최적화와 클라우드 혁신

Stephen Orban은 기업 전략 블로그의 [클라우드 우선주의를 향한 여정 및 채택 단계에 대한 블로그 게시물](#)에서 이러한 단계를 정의했습니다. AWS 클라우드 [이들이 AWS 마이그레이션 전략과 어떤 관련이 있는지에 대한 자세한 내용은 마이그레이션 준비 가이드를 참조하십시오.](#)

## CMDB

[구성 관리 데이터베이스](#)를 참조하십시오.

## 코드 리포지토리

소스 코드와 설명서, 샘플, 스크립트 등의 기타 자산이 버전 관리 프로세스를 통해 저장되고 업데이트되는 위치입니다. 일반 클라우드 리포지토리에는 또는 이 포함됩니다 GitHub . AWS CodeCommit코드의 각 버전을 브랜치라고 합니다. 마이크로서비스 구조에서 각 리포지토리는 단일 기능 전용입니다. 단일 CI/CD 파이프라인은 여러 리포지토리를 사용할 수 있습니다.

## 콜드 캐시

비어 있거나, 제대로 채워지지 않았거나, 오래되었거나 관련 없는 데이터를 포함하는 버퍼 캐시입니다. 주 메모리나 디스크에서 데이터베이스 인스턴스를 읽어야 하기 때문에 성능에 영향을 미치며, 이는 버퍼 캐시에서 읽는 것보다 느립니다.

## 콜드 데이터

거의 액세스되지 않고 일반적으로 과거 데이터인 데이터. 이런 종류의 데이터를 쿼리할 때는 일반적으로 느린 쿼리가 허용됩니다. 이 데이터를 성능이 낮고 비용이 저렴한 스토리지 계층 또는 클래스로 옮기면 비용을 절감할 수 있습니다.

## 컴퓨터 비전 (CV)

기계 학습을 사용하여 디지털 이미지 및 비디오와 같은 시각적 형식에서 정보를 분석하고 추출하는 [AI](#) 분야. 예를 들어 AWS Panorama 는 온프레미스 카메라 네트워크에 CV를 추가하는 디바이스를 제공하고, SageMaker Amazon은 CV용 이미지 처리 알고리즘을 제공합니다.

## 구성 드리프트

워크로드의 경우 구성이 예상 상태에서 변경됩니다. 이로 인해 워크로드가 규정을 준수하지 않게 될 수 있으며, 일반적으로 점진적이고 의도하지 않은 방식으로 진행됩니다.

## 구성 관리 데이터베이스(CMDB)

하드웨어 및 소프트웨어 구성 요소와 해당 구성을 포함하여 데이터베이스와 해당 IT 환경에 대한 정보를 저장하고 관리하는 리포지토리입니다. 일반적으로 마이그레이션의 포트폴리오 검색 및 분석 단계에서 CMDB의 데이터를 사용합니다.

### 규정 준수 팩

AWS Config 규정 준수 및 보안 검사를 사용자 지정하기 위해 조합할 수 있는 규칙 및 수정 조치 모음입니다. YAML 템플릿을 사용하여 한 AWS 계정 및 지역 또는 조직 전체에 단일 엔티티로 적합성 팩을 배포할 수 있습니다. 자세한 내용은 설명서의 [적합성 팩](#)을 참조하십시오. AWS Config

### 지속적 통합 및 지속적 전달(CI/CD)

소프트웨어 릴리스 프로세스의 소스, 빌드, 테스트, 스테이징 및 프로덕션 단계를 자동화하는 프로세스입니다. CI/CD는 일반적으로 파이프라인으로 설명됩니다. CI/CD를 통해 프로세스를 자동화하고, 생산성을 높이고, 코드 품질을 개선하고, 더 빠르게 제공할 수 있습니다. 자세한 내용은 [지속적 전달의 이점](#)을 참조하십시오. CD는 지속적 배포를 의미하기도 합니다. 자세한 내용은 [지속적 전달\(Continuous Delivery\)](#)과 [지속적인 개발](#)을 참조하십시오.

## CV

[컴퓨터 비전을 참조하십시오.](#)

## D

### 저장 데이터

스토리지에 있는 데이터와 같이 네트워크에 고정되어 있는 데이터입니다.

### 데이터 분류

중요도와 민감도를 기준으로 네트워크의 데이터를 식별하고 분류하는 프로세스입니다. 이 프로세스는 데이터에 대한 적절한 보호 및 보존 제어를 결정하는 데 도움이 되므로 사이버 보안 위험 관리 전략의 중요한 구성 요소입니다. 데이터 분류는 AWS Well-Architected 프레임워크의 보안 핵심 요소입니다. 자세한 내용은 [데이터 분류](#)를 참조하십시오.

### 데이터 드리프트

프로덕션 데이터와 ML 모델 학습에 사용된 데이터 간의 상당한 차이 또는 시간 경과에 따른 입력 데이터의 의미 있는 변화. 데이터 드리프트는 ML 모델 예측의 전반적인 품질, 정확성 및 공정성을 저하시킬 수 있습니다.

## 전송 중 데이터

네트워크를 통과하고 있는 데이터입니다. 네트워크 리소스 사이를 이동 중인 데이터를 예로 들 수 있습니다.

## 데이터 메시

중앙 집중식 관리 및 거버넌스와 함께 분산되고 분산된 데이터 소유권을 제공하는 아키텍처 프레임워크입니다.

## 데이터 최소화

꼭 필요한 데이터만 수집하고 처리하는 원칙입니다. 에서 데이터 최소화를 실천하면 개인 정보 보호 위험, 비용 및 분석에 따른 탄소 발자국을 줄일 AWS 클라우드 수 있습니다.

## 데이터 경계

신뢰할 수 있는 ID만 예상 네트워크에서 신뢰할 수 있는 리소스에 액세스하도록 하는 데 도움이 되는 AWS 환경 내 일련의 예방 가드레일입니다. 자세한 내용은 [데이터 경계 구축을 참조하십시오](#).

AWS

## 데이터 사전 처리

원시 데이터를 ML 모델이 쉽게 구문 분석할 수 있는 형식으로 변환하는 것입니다. 데이터를 사전 처리한다는 것은 특정 열이나 행을 제거하고 누락된 값, 일관성이 없는 값 또는 중복 값을 처리함을 의미할 수 있습니다.

## 데이터 출처

라이프사이클 전반에 걸쳐 데이터의 출처와 기록을 추적하는 프로세스(예: 데이터 생성, 전송, 저장 방법).

## 데이터 주체

데이터를 수집 및 처리하는 개인입니다.

## 데이터 웨어하우스

분석과 같은 비즈니스 인텔리전스를 지원하는 데이터 관리 시스템. 데이터 웨어하우스에는 일반적으로 대량의 과거 데이터가 포함되며 일반적으로 쿼리 및 분석에 사용됩니다.

## 데이터 정의 언어(DDL)

데이터베이스에서 테이블 및 객체의 구조를 만들거나 수정하기 위한 명령문 또는 명령입니다.

## 데이터베이스 조작 언어(DML)

데이터베이스에서 정보를 수정(삽입, 업데이트 및 삭제)하기 위한 명령문 또는 명령입니다.

## DDL

[데이터베이스 정의 언어](#)를 참조하십시오.

### 딥 앙상블

예측을 위해 여러 딥 러닝 모델을 결합하는 것입니다. 딥 앙상블을 사용하여 더 정확한 예측을 얻거나 예측의 불확실성을 추정할 수 있습니다.

### 딥 러닝

여러 계층의 인공 신경망을 사용하여 입력 데이터와 관심 대상 변수 간의 매핑을 식별하는 ML 하위 분야입니다.

### defense-in-depth

네트워크와 그 안의 데이터 기밀성, 무결성 및 가용성을 보호하기 위해 컴퓨터 네트워크 전체에 일련의 보안 메커니즘과 제어를 신중하게 계층화하는 정보 보안 접근 방식입니다. 이 전략을 채택하면 AWS Organizations 구조의 여러 계층에 AWS 여러 컨트롤을 추가하여 리소스를 보호하는 데 도움이 됩니다. 예를 들어 다단계 인증, 네트워크 세분화, 암호화를 결합한 defense-in-depth 접근 방식을 사용할 수 있습니다.

### 위임된 관리자

에서 AWS Organizations 호환 가능한 서비스는 AWS 구성원 계정을 등록하여 조직의 계정을 관리하고 해당 서비스에 대한 권한을 관리할 수 있습니다. 이러한 계정을 해당 서비스의 위임된 관리자라고 합니다. 자세한 내용과 호환되는 서비스 목록은 AWS Organizations 설명서의 [AWS Organizations와 함께 사용할 수 있는 AWS 서비스](#)를 참조하십시오.

### 배포

대상 환경에서 애플리케이션, 새 기능 또는 코드 수정 사항을 사용할 수 있도록 하는 프로세스입니다. 배포에는 코드 베이스의 변경 사항을 구현한 다음 애플리케이션 환경에서 해당 코드베이스를 구축하고 실행하는 작업이 포함됩니다.

### 개발 환경

[환경](#)을 참조하십시오.

### 탐지 제어

이벤트 발생 후 탐지, 기록 및 알림을 수행하도록 설계된 보안 제어입니다. 이러한 제어는 기존의 예방적 제어를 우회한 보안 이벤트를 알리는 2차 방어선입니다. 자세한 내용은 Implementing security controls on AWS의 [Detective controls](#)를 참조하십시오.

## 개발 가치 흐름 매핑 (DVSM)

소프트웨어 개발 라이프사이클에서 속도와 품질에 부정적인 영향을 미치는 제약 조건을 식별하고 우선 순위를 지정하는 데 사용되는 프로세스입니다. DVSM은 원래 린 제조 방식을 위해 설계된 가치 흐름 매핑 프로세스를 확장합니다. 소프트웨어 개발 프로세스를 통해 가치를 창출하고 이동하는 데 필요한 단계와 팀에 중점을 둡니다.

## 디지털 트윈

건물, 공장, 산업 장비 또는 생산 라인과 같은 실제 시스템을 가상으로 표현한 것입니다. 디지털 트윈은 예측 유지 보수, 원격 모니터링, 생산 최적화를 지원합니다.

## 치수 표

[스타 스키마에서](#) 팩트 테이블의 양적 데이터에 대한 데이터 속성을 포함하는 작은 테이블입니다. 차원 테이블 속성은 일반적으로 텍스트처럼 동작하는 텍스트 필드 또는 불연속형 숫자입니다. 이러한 속성은 일반적으로 쿼리 제한, 필터링 및 결과 집합 레이블 지정에 사용됩니다.

## 재해

워크로드 또는 시스템이 기본 배포 위치에서 비즈니스 목표를 달성하지 못하게 방해하는 이벤트입니다. 이러한 이벤트는 자연재해, 기술적 오류, 의도하지 않은 구성 오류 또는 멀웨어 공격과 같은 사람의 행동으로 인한 결과일 수 있습니다.

## 재해 복구(DR)

[재해로 인한 다운타임과 데이터 손실을 최소화하기 위해 사용하는 전략과 프로세스입니다.](#) 자세한 내용은 [워크로드의 재해 복구 AWS: AWS Well-Architected 프레임워크에서의 클라우드 복구를 참조하십시오.](#)

## DML

[데이터베이스](#) 조작 언어를 참조하십시오.

## 도메인 기반 설계

구성 요소를 각 구성 요소가 제공하는 진화하는 도메인 또는 핵심 비즈니스 목표에 연결하여 복잡한 소프트웨어 시스템을 개발하는 접근 방식입니다. 이 개념은 에릭 에반스에 의해 그의 저서인 도메인 기반 디자인: 소프트웨어 중심의 복잡성 해결(Boston: Addison-Wesley Professional, 2003)에서 소개되었습니다. Strangler Fig 패턴과 함께 도메인 기반 설계를 사용하는 방법에 대한 자세한 내용은 [컨테이너 및 Amazon API Gateway를 사용하여 기존의 Microsoft ASP.NET\(ASMX\) 웹 서비스를 점진적으로 현대화하는 방법](#)을 참조하십시오.

## DR

[재해 복구](#)를 참조하십시오.

## 드리프트 감지

기존 구성으로부터의 편차 추적. 예를 들어 [시스템 리소스의 편차를 감지하는 AWS CloudFormation](#) 데 사용하거나 거버넌스 요구 사항 준수에 영향을 미칠 수 있는 [착륙 지대의 변경 사항을 탐지하는 AWS Control Tower](#) 데 사용할 수 있습니다.

## DVSM

[개발 가치 흐름 매핑](#) 참조하십시오.

## E

### EDA

[탐색적 데이터 분석](#) 참조하십시오.

### 엣지 컴퓨팅

IoT 네트워크의 엣지에서 스마트 디바이스의 컴퓨팅 성능을 개선하는 기술 [클라우드 컴퓨팅과](#) 비교할 때 엣지 컴퓨팅은 통신 대기 시간을 줄이고 응답 시간을 개선할 수 있습니다.

### 암호화

사람이 읽을 수 있는 일반 텍스트 데이터를 암호문으로 변환하는 컴퓨팅 프로세스입니다.

### 암호화 키

암호화 알고리즘에 의해 생성되는 무작위 비트의 암호화 문자열입니다. 키의 길이는 다양할 수 있으며 각 키는 예측할 수 없고 고유하게 설계되었습니다.

### 엔디안

컴퓨터 메모리에 바이트가 저장되는 순서입니다. 빅 엔디안 시스템은 가장 중요한 바이트를 먼저 저장합니다. 리틀 엔디안 시스템은 가장 덜 중요한 바이트를 먼저 저장합니다.

### 엔드포인트

[서비스](#) 엔드포인트를 참조하십시오.

### 엔드포인트 서비스

Virtual Private Cloud(VPC)에서 호스팅하여 다른 사용자와 공유할 수 있는 서비스입니다. 다른 주체 AWS 계정 또는 AWS Identity and Access Management (IAM) 보안 주체에 권한을 부여하여 엔드포인트 서비스를 생성하고 권한을 부여할 수 있습니다. AWS PrivateLink 이러한 계정 또는 보안 주체는 인터페이스 VPC 엔드포인트를 생성하여 엔드포인트 서비스에 비공개로 연결할 수 있습니다.

다. 자세한 내용은 Amazon Virtual Private Cloud(VPC) 설명서의 [엔드포인트 서비스 생성](#)을 참조하십시오.

## ERP (전사적 자원 관리)

기업의 주요 비즈니스 프로세스 (예: 회계, [MES](#), 프로젝트 관리) 를 자동화하고 관리하는 시스템입니다.

## 봉투 암호화

암호화 키를 다른 암호화 키로 암호화하는 프로세스입니다. 자세한 내용은 AWS Key Management Service (AWS KMS) [설명서의 봉투 암호화](#)를 참조하십시오.

## 환경

실행 중인 애플리케이션의 인스턴스입니다. 다음은 클라우드 컴퓨팅의 일반적인 환경 유형입니다.

- 개발 환경 - 애플리케이션 유지 관리를 담당하는 핵심 팀만 사용할 수 있는 실행 중인 애플리케이션의 인스턴스입니다. 개발 환경은 변경 사항을 상위 환경으로 승격하기 전에 테스트하는 데 사용됩니다. 이러한 유형의 환경을 테스트 환경이라고도 합니다.
- 하위 환경 - 초기 빌드 및 테스트에 사용되는 환경을 비롯한 애플리케이션의 모든 개발 환경입니다.
- 프로덕션 환경 - 최종 사용자가 액세스할 수 있는 실행 중인 애플리케이션의 인스턴스입니다. CI/CD 파이프라인에서 프로덕션 환경이 마지막 배포 환경입니다.
- 상위 환경 - 핵심 개발 팀 이외의 사용자가 액세스할 수 있는 모든 환경입니다. 프로덕션 환경, 프로덕션 이전 환경 및 사용자 수용 테스트를 위한 환경이 여기에 포함될 수 있습니다.

## 에픽

애자일 방법론에서 작업을 구성하고 우선순위를 정하는 데 도움이 되는 기능적 범주입니다. 에픽은 요구 사항 및 구현 작업에 대한 개괄적인 설명을 제공합니다. 예를 들어 AWS CAF 보안 에픽에는 ID 및 액세스 관리, 탐지 제어, 인프라 보안, 데이터 보호, 사고 대응 등이 포함됩니다. AWS 마이그레이션 전략의 에픽에 대한 자세한 내용은 [프로그램 구현 가이드](#)를 참조하십시오.

## ERP

[엔터프라이즈 리소스 계획을](#) 참조하십시오.

## 탐색 데이터 분석(EDA)

데이터 세트를 분석하여 주요 특성을 파악하는 프로세스입니다. 데이터를 수집 또는 집계한 다음 초기 조사를 수행하여 패턴을 찾고, 이상을 탐지하고, 가정을 확인합니다. EDA는 요약 통계를 계산하고 데이터 시각화를 생성하여 수행됩니다.

## F

### 팩트 테이블

[스타 스키마의](#) 중앙 테이블. 비즈니스 운영에 대한 정량적 데이터를 저장합니다. 일반적으로 팩트 테이블에는 측정값이 포함된 열과 차원 테이블의 외부 키가 포함된 열 등 두 가지 유형의 열이 포함됩니다.

### 빨리 실패하세요

빈번하고 점진적인 테스트를 통해 개발 라이프사이클을 단축하는 철학. 이는 애자일 접근 방식의 중요한 부분입니다.

### 장애 격리 경계

장애 영향을 제한하고 워크로드의 복원력을 개선하는 데 도움이 되는 가용 영역 AWS 리전, 컨트를 플레인 또는 데이터 플레인과 같은 경계 AWS 클라우드자세한 내용은 [AWS 장애 격리](#) 경계를 참조하십시오.

### 기능 브랜치

[브랜치를](#) 참조하십시오.

### 기능

예측에 사용하는 입력 데이터입니다. 예를 들어, 제조 환경에서 기능은 제조 라인에서 주기적으로 캡처되는 이미지일 수 있습니다.

### 기능 중요도

모델의 예측에 특성이 얼마나 중요한지를 나타냅니다. 이는 일반적으로 SHAP(Shapley Additive Descriptions) 및 통합 그래디언트와 같은 다양한 기법을 통해 계산할 수 있는 수치 점수로 표현됩니다. 자세한 내용은 [다음은AWS사용한 기계 학습 모델 해석 가능성을](#) 참조하십시오.

### 기능 변환

추가 소스로 데이터를 보강하거나, 값을 조정하거나, 단일 데이터 필드에서 여러 정보 세트를 추출하는 등 ML 프로세스를 위해 데이터를 최적화하는 것입니다. 이를 통해 ML 모델이 데이터를 활용할 수 있습니다. 예를 들어, 날짜 '2021-05-27 00:15:37'을 '2021년', '5월', '목', '15일'로 분류하면 학습 알고리즘이 다양한 데이터 구성 요소와 관련된 미묘한 패턴을 학습하는 데 도움이 됩니다.

### FGAC

[세분화된 액세스 제어](#)를 참조하십시오.

## 세분화된 액세스 제어(FGAC)

여러 조건을 사용하여 액세스 요청을 허용하거나 거부합니다.

## 플래시컷 마이그레이션

단계별 접근 방식 대신 [변경 데이터 캡처를 통한 지속적인 데이터](#) 복제를 통해 최단 시간에 데이터를 마이그레이션하는 데이터베이스 마이그레이션 방법입니다. 목표는 가동 중지 시간을 최소화하는 것입니다.

## G

### 지리적 차단

[지리적 제한](#)을 참조하십시오.

### 지리적 제한(지리적 차단)

CloudFrontAmazon에서는 특정 국가의 사용자가 콘텐츠 배포에 액세스하지 못하도록 하는 옵션을 제공합니다. 허용 목록 또는 차단 목록을 사용하여 승인된 국가와 차단된 국가를 지정할 수 있습니다. 자세한 내용은 [설명서의 콘텐츠의 지리적 배포 제한](#)을 참조하십시오. CloudFront

### Gitflow 워크플로

하위 환경과 상위 환경이 소스 코드 리포지토리의 서로 다른 브랜치를 사용하는 방식입니다. Gitflow 워크플로는 레거시로 간주되며 [트렁크 기반 워크플로는](#) 현대적이고 선호되는 접근 방식입니다.

### 브라운필드 전략

새로운 환경에서 기존 인프라의 부재 시스템 아키텍처에 대한 그린필드 전략을 채택할 때 [브라운필드](#)라고도 하는 기존 인프라와의 호환성 제한 없이 모든 새로운 기술을 선택할 수 있습니다. 기존 인프라를 확장하는 경우 브라운필드 전략과 그린필드 전략을 혼합할 수 있습니다.

### 가드레일

조직 단위(OU) 전체에서 리소스, 정책 및 규정 준수를 관리하는 데 도움이 되는 중요 규칙입니다. 예방 가드레일은 규정 준수 표준에 부합하도록 정책을 시행하며, 서비스 제어 정책과 IAM 권한 경계를 사용하여 구현됩니다. 탐지 가드레일은 정책 위반 및 규정 준수 문제를 감지하고 해결을 위한 알림을 생성하며, 이들은, Amazon AWS Config AWS Security Hub GuardDuty AWS Trusted Advisor, Amazon Inspector 및 사용자 지정 AWS Lambda 검사를 사용하여 구현됩니다.

# H

## 하

[고가용성을](#) 확인하세요.

### 이기종 데이터베이스 마이그레이션

다른 데이터베이스 엔진을 사용하는 대상 데이터베이스로 소스 데이터베이스 마이그레이션(예: Oracle에서 Amazon Aurora로) 이기종 마이그레이션은 일반적으로 리아키텍트 작업의 일부이며 스키마를 변환하는 것은 복잡한 작업일 수 있습니다. AWS 는 스키마 변환에 도움이 되는 [AWS SCT](#)를 제공합니다.

### 높은 가용성(HA)

문제나 재해 발생 시 개입 없이 지속적으로 운영할 수 있는 워크로드의 능력. HA 시스템은 자동으로 장애 조치되고, 지속적으로 고품질 성능을 제공하고, 성능에 미치는 영향을 최소화하면서 다양한 부하와 장애를 처리하도록 설계되었습니다.

### 히스토리언 현대화

제조 산업의 요구 사항을 더 잘 충족하도록 운영 기술(OT) 시스템을 현대화하고 업그레이드하는 데 사용되는 접근 방식입니다. 히스토리언은 공장의 다양한 출처에서 데이터를 수집하고 저장하는 데 사용되는 일종의 데이터베이스입니다.

### 동종 데이터베이스 마이그레이션

동일한 데이터베이스 엔진을 공유하는 대상 데이터베이스로 소스 데이터베이스 마이그레이션(예: Microsoft SQL Server에서 Amazon RDS for SQL Server로) 동종 마이그레이션은 일반적으로 리호스팅 또는 리플랫폼 작업의 일부입니다. 네이티브 데이터베이스 유틸리티를 사용하여 스키마를 마이그레이션할 수 있습니다.

### 핫 데이터

자주 액세스하는 데이터(예: 실시간 데이터 또는 최근 번역 데이터). 일반적으로 이 데이터에는 빠른 쿼리 응답을 제공하기 위한 고성능 스토리지 계층 또는 클래스가 필요합니다.

### 핫픽스

프로덕션 환경의 중요한 문제를 해결하기 위한 긴급 수정입니다. 긴급성 때문에 핫픽스는 일반적으로 일반적인 DevOps 릴리스 워크플로 외부에서 만들어집니다.

## 하이퍼케어 기간

전환 직후 마이그레이션 팀이 문제를 해결하기 위해 클라우드에서 마이그레이션된 애플리케이션을 관리하고 모니터링하는 기간입니다. 일반적으로 이 기간은 1~4일입니다. 하이퍼케어 기간이 끝나면 마이그레이션 팀은 일반적으로 애플리케이션에 대한 책임을 클라우드 운영 팀에 넘깁니다.

## I

### laC

[인프라를 코드로 보세요.](#)

#### 자격 증명 기반 정책

환경 내에서 권한을 정의하는 하나 이상의 IAM 보안 주체에 연결된 정책입니다. AWS 클라우드 유휴 애플리케이션

90일 동안 평균 CPU 및 메모리 사용량이 5~20%인 애플리케이션입니다. 마이그레이션 프로젝트에서는 이러한 애플리케이션을 사용 중지하거나 온프레미스에 유지하는 것이 일반적입니다.

### IIoT

[산업용 사물 인터넷을 참조하십시오.](#)

#### 불변의 인프라

기존 인프라를 업데이트, 패치 또는 수정하는 대신 프로덕션 워크로드용 새 인프라를 배포하는 모델입니다. [변경 불가능한 인프라는 기본적으로 변경 가능한 인프라보다 더 일관되고 안정적이며 예측 가능합니다.](#) 자세한 내용은 Well-Architected AWS 프레임워크의 [변경 불가능한 인프라를 사용한 배포](#) 모범 사례를 참조하십시오.

#### 인바운드(수신) VPC

AWS 다중 계정 아키텍처에서 VPC는 애플리케이션 외부에서 네트워크 연결을 허용, 검사 및 라우팅합니다. [AWS Security Reference Architecture](#)에서는 애플리케이션과 더 넓은 인터넷 간의 양방향 인터페이스를 보호하기 위해 인바운드, 아웃바운드 및 검사 VPC로 네트워크 계정을 설정할 것을 권장합니다.

#### 중분 마이그레이션

한 번에 전체 전환을 수행하는 대신 애플리케이션을 조금씩 마이그레이션하는 전환 전략입니다. 예를 들어, 처음에는 소수의 마이크로서비스나 사용자만 새 시스템으로 이동할 수 있습니다. 모든 것

이 제대로 작동하는지 확인한 후에는 레거시 시스템을 폐기할 수 있을 때까지 추가 마이크로서비스 또는 사용자를 점진적으로 이동할 수 있습니다. 이 전략을 사용하면 대규모 마이그레이션과 관련된 위험을 줄일 수 있습니다.

## Industry 4.0

[Klaus Schwab](#)이 연결성, 실시간 데이터, 자동화, 분석 및 AI/ML의 발전을 통한 제조 프로세스의 현대화를 지칭하기 위해 2016년 도입한 용어입니다.

## 인프라

애플리케이션의 환경 내에 포함된 모든 리소스와 자산입니다.

## 코드형 인프라(IaC)

구성 파일 세트를 통해 애플리케이션의 인프라를 프로비저닝하고 관리하는 프로세스입니다. IaC는 새로운 환경의 반복 가능성, 신뢰성 및 일관성을 위해 인프라 관리를 중앙 집중화하고, 리소스를 표준화하고, 빠르게 확장할 수 있도록 설계되었습니다.

## 산업용 사물 인터넷(IIoT)

제조, 에너지, 자동차, 의료, 생명과학, 농업 등의 산업 부문에서 인터넷에 연결된 센서 및 디바이스의 사용 자세한 내용은 [산업용 사물 인터넷\(IoT\) 디지털 트랜스포메이션 전략 구축](#)을 참조하십시오.

## 검사 VPC

AWS 다중 계정 아키텍처에서 VPC (동일하거나 AWS 리전다른), 인터넷 및 온프레미스 네트워크 간의 네트워크 트래픽 검사를 관리하는 중앙 집중식 VPC입니다. [AWS Security Reference Architecture](#)에서는 애플리케이션과 더 넓은 인터넷 간의 양방향 인터페이스를 보호하기 위해 인바운드, 아웃바운드 및 검사 VPC로 네트워크 계정을 설정할 것을 권장합니다.

## 사물 인터넷(IoT)

인터넷이나 로컬 통신 네트워크를 통해 다른 디바이스 및 시스템과 통신하는 센서 또는 프로세서가 내장된 연결된 물리적 객체의 네트워크 자세한 내용은 [IoT란?](#)을 참조하십시오.

## 해석력

모델의 예측이 입력에 따라 어떻게 달라지는지를 사람이 이해할 수 있는 정도를 설명하는 기계 학습 모델의 특성입니다. 자세한 내용은 [Machine learning model interpretability with AWS](#)를 참조하십시오.

## IoT

[사물 인터넷을 참조하십시오.](#)

## IT 정보 라이브러리(TIL)

IT 서비스를 제공하고 이러한 서비스를 비즈니스 요구 사항에 맞게 조정하기 위한 일련의 모범 사례 ITIL은 ITSM의 기반을 제공합니다.

## IT 서비스 관리(TSM)

조직의 IT 서비스 설계, 구현, 관리 및 지원과 관련된 활동 클라우드 운영을 ITSM 도구와 통합하는 방법에 대한 자세한 내용은 [운영 통합 가이드](#)를 참조하십시오.

## ITIL

[IT 정보 라이브러리를](#) 참조하십시오.

## ITSM

[IT 서비스 관리를](#) 참조하십시오.

## L

### 레이블 기반 액세스 제어(LBAC)

사용자 및 데이터 자체에 각각 보안 레이블 값을 명시적으로 할당하는 필수 액세스 제어(MAC)를 구현한 것입니다. 사용자 보안 레이블과 데이터 보안 레이블 간의 교차 부분에 따라 사용자가 볼 수 있는 행과 열이 결정됩니다.

### 랜딩 존

Landing Zone은 확장 가능하고 안전한 잘 설계된 다중 계정 AWS 환경입니다. 조직은 여기에서부터 보안 및 인프라 환경에 대한 확신을 가지고 워크로드와 애플리케이션을 신속하게 시작하고 배포할 수 있습니다. 랜딩 존에 대한 자세한 내용은 [안전하고 확장 가능한 다중 계정 AWS 환경 설정](#)을 참조하십시오.

### 대규모 마이그레이션

300대 이상의 서버 마이그레이션입니다.

## LBAC

[레이블 기반 액세스 제어를](#) 참조하십시오.

### 최소 권한

작업을 수행하는 데 필요한 최소 권한을 부여하는 보안 모범 사례입니다. 자세한 내용은 IAM 설명서의 [최소 권한 적용](#)을 참조하십시오.

## 리프트 앤드 시프트

[7 R](#)을 참조하십시오.

## 리틀 엔디안 시스템

가장 덜 중요한 바이트를 먼저 저장하는 시스템입니다. [엔디안](#) 참조.

## 하위 환경

[환경 참조](#).

# M

## 기계 학습(ML)

패턴 인식 및 학습에 알고리즘과 기법을 사용하는 인공지능의 한 유형입니다. ML은 사물 인터넷 (IoT) 데이터와 같은 기록된 데이터를 분석하고 학습하여 패턴을 기반으로 통계 모델을 생성합니다. 자세한 내용은 [기계 학습](#)을 참조하십시오.

## 기본 브랜치

[브랜치](#) 참조.

## 악성 코드

컴퓨터 보안 또는 개인 정보를 침해하도록 설계된 소프트웨어 멀웨어는 컴퓨터 시스템을 방해하거나, 민감한 정보를 유출하거나, 무단 액세스를 얻을 수 있습니다. 멀웨어의 예로는 바이러스, 웜, 랜섬웨어, 트로이 목마, 스파이웨어, 키로거 등이 있습니다.

## 매니지드 서비스

AWS 서비스 인프라 계층, 운영 체제 및 플랫폼을 AWS 운영하며 사용자는 엔드포인트에 액세스하여 데이터를 저장하고 검색합니다. 관리형 서비스의 예로는 아마존 심플 스토리지 서비스 (Amazon S3) 와 아마존 DynamoDB가 있습니다. 이러한 서비스를 추상화된 서비스라고도 합니다.

## 제조 실행 시스템 (MES)

제조 현장에서 원자재를 완제품으로 전환하는 생산 프로세스를 추적, 모니터링, 문서화 및 제어하기 위한 소프트웨어 시스템입니다.

## MAP

[Migration Acceleration 프로그램](#)을 참조하십시오.

## 기구

도구를 만들고 도구 채택을 유도한 다음 결과를 검토하여 조정하는 전체 프로세스입니다. 메커니즘은 작동하면서 자체적으로 강화되고 개선되는 사이클입니다. 자세한 내용은 [AWS Well-Architected 프레임워크에서의 메커니즘 구축을](#) 참조하십시오.

## 멤버 계정

조직의 일부인 관리 계정을 AWS 계정 제외한 모든 계정 AWS Organizations 하나의 계정은 한 번에 하나의 조직 멤버만 될 수 있습니다.

## MES

[제조 실행 시스템을](#) 참조하십시오.

## 메시지 큐 텔레메트리 전송 (MQTT)

[퍼블리시/구독 패턴을 기반으로 하는 리소스가 제한된 IoT 디바이스를 위한 경량 machine-to-machine \(M2M\) 통신 프로토콜입니다.](#)

## 마이크로서비스

잘 정의된 API를 통해 통신하고 일반적으로 소규모 자체 팀이 소유하는 소규모 독립 서비스입니다. 예를 들어, 보험 시스템에는 영업, 마케팅 등의 비즈니스 역량이나 구매, 청구, 분석 등의 하위 영역에 매핑되는 마이크로 서비스가 포함될 수 있습니다. 마이크로서비스의 이점으로 민첩성, 유연한 확장, 손쉬운 배포, 재사용 가능한 코드, 복원력 등이 있습니다. [자세한 내용은 서버리스 서비스를 사용하여 마이크로서비스 통합을](#) 참조하십시오. [AWS](#)

## 마이크로서비스 아키텍처

각 애플리케이션 프로세스를 마이크로서비스로 실행하는 독립 구성 요소를 사용하여 애플리케이션을 구축하는 접근 방식입니다. 이러한 마이크로서비스는 경량 API를 사용하여 잘 정의된 인터페이스를 통해 통신합니다. 애플리케이션의 특정 기능에 대한 수요에 맞게 이 아키텍처의 각 마이크로 서비스를 업데이트, 배포 및 조정할 수 있습니다. 자세한 내용은 마이크로서비스 [구현을](#) 참조하십시오. [AWS](#)

## Migration Acceleration Program(MAP)

조직이 클라우드로 전환하기 위한 강력한 운영 기반을 구축하고 초기 마이그레이션 비용을 상쇄할 수 있도록 컨설팅 지원, 교육 및 서비스를 제공하는 AWS 프로그램입니다. MAP에는 레거시 마이그레이션을 체계적인 방식으로 실행하기 위한 마이그레이션 방법론과 일반적인 마이그레이션 시나리오를 자동화하고 가속화하는 도구 세트가 포함되어 있습니다.

## 대규모 마이그레이션

애플리케이션 포트폴리오의 대다수를 웨이브를 통해 클라우드로 이동하는 프로세스로, 각 웨이브에서 더 많은 애플리케이션이 더 빠른 속도로 이동합니다. 이 단계에서는 이전 단계에서 배운 모범 사례와 교훈을 사용하여 팀, 도구 및 프로세스의 마이그레이션 팩토리를 구현하여 자동화 및 민첩한 제공을 통해 워크로드 마이그레이션을 간소화합니다. 이것은 [AWS 마이그레이션 전략](#)의 세 번째 단계입니다.

### 마이그레이션 팩토리

자동화되고 민첩한 접근 방식을 통해 워크로드 마이그레이션을 간소화하는 다기능 팀입니다. 마이그레이션 팩토리 팀에는 일반적으로 운영, 비즈니스 분석가 및 소유자, 마이그레이션 엔지니어, 개발자 및 스프린트에서 일하는 DevOps 전문가가 포함됩니다. 엔터프라이즈 애플리케이션 포트폴리오의 20~50%는 공장 접근 방식으로 최적화할 수 있는 반복되는 패턴으로 구성되어 있습니다. 자세한 내용은 이 콘텐츠 세트의 [클라우드 마이그레이션 팩토리 가이드](#)와 [마이그레이션 팩토리에 대한 설명](#)을 참조하십시오.

### 마이그레이션 메타데이터

마이그레이션을 완료하는 데 필요한 애플리케이션 및 서버에 대한 정보 각 마이그레이션 패턴에는 서로 다른 마이그레이션 메타데이터 세트가 필요합니다. 마이그레이션 메타데이터의 예로는 대상 서브넷, 보안 그룹, 계정 등이 있습니다. AWS

### 마이그레이션 패턴

사용되는 마이그레이션 전략, 마이그레이션 대상, 마이그레이션 애플리케이션 또는 서비스를 자세히 설명하는 반복 가능한 마이그레이션 작업입니다. 예: 애플리케이션 마이그레이션 서비스를 사용하여 Amazon EC2로 AWS 마이그레이션을 재호스팅합니다.

### Migration Portfolio Assessment(MPA)

로 마이그레이션하기 위한 비즈니스 사례를 검증하기 위한 정보를 제공하는 온라인 도구입니다. AWS 클라우드 MPA는 상세한 포트폴리오 평가(서버 적정 규모 조정, 가격 책정, TCO 비교, 마이그레이션 비용 분석)와 마이그레이션 계획(애플리케이션 데이터 분석 및 데이터 수집, 애플리케이션 그룹화, 마이그레이션 우선순위 지정, 웨이브 계획)을 제공합니다. [MPA 도구](#) (로그인 필요) 는 모든 컨설턴트와 APN 파트너 AWS 컨설턴트에게 무료로 제공됩니다.

### 마이그레이션 준비 상태 평가(MRA)

CAF를 사용하여 조직의 클라우드 준비 상태에 대한 통찰력을 얻고, 강점과 약점을 파악하고, 식별된 격차를 해소하기 위한 실행 계획을 수립하는 프로세스입니다. AWS 자세한 내용은 [마이그레이션 준비 가이드](#)를 참조하십시오. MRA는 [AWS 마이그레이션 전략](#)의 첫 번째 단계입니다.

## 마이그레이션 전략

워크로드를 로 마이그레이션하는 데 사용된 접근 방식. AWS 클라우드자세한 내용은 이 용어집의 [7R 항목](#) 및 [대규모 마이그레이션 가속화를 위한 조직 동원을 참조하십시오](#).

## ML

[기계 학습을 참조하십시오](#).

## 현대화

비용을 절감하고 효율성을 높이고 혁신을 활용하기 위해 구식(레거시 또는 모놀리식) 애플리케이션과 해당 인프라를 클라우드의 민첩하고 탄력적이고 가용성이 높은 시스템으로 전환하는 것입니다. 자세한 내용은 [의 AWS 클라우드애플리케이션 현대화 전략](#)을 참조하십시오.

## 현대화 준비 상태 평가

조직 애플리케이션의 현대화 준비 상태를 파악하고, 이점, 위험 및 종속성을 식별하고, 조직이 해당 애플리케이션의 향후 상태를 얼마나 잘 지원할 수 있는지를 확인하는 데 도움이 되는 평가입니다. 평가 결과는 대상 아키텍처의 청사진, 현대화 프로세스의 개발 단계와 마일스톤을 자세히 설명하는 로드맵 및 파악된 격차를 해소하기 위한 실행 계획입니다. 자세한 내용은 [에서 애플리케이션의 현대화 준비 상태 평가를 참조하십시오](#). AWS 클라우드

## 모놀리식 애플리케이션(모놀리식 유형)

긴밀하게 연결된 프로세스를 사용하여 단일 서비스로 실행되는 애플리케이션입니다. 모놀리식 애플리케이션에는 몇 가지 단점이 있습니다. 한 애플리케이션 기능에 대한 수요가 급증하면 전체 아키텍처 규모를 조정해야 합니다. 코드 베이스가 커지면 모놀리식 애플리케이션의 기능을 추가하거나 개선하는 것도 더 복잡해집니다. 이러한 문제를 해결하기 위해 마이크로서비스 아키텍처를 사용할 수 있습니다. 자세한 내용은 [마이크로서비스로 모놀리식 유형 분해](#)를 참조하십시오.

## MPA

[마이그레이션 포트폴리오 평가](#)를 참조하십시오.

## MQTT

[메시지 큐 원격 분석 전송](#)을 참조하십시오.

## 멀티클래스 분류

여러 클래스에 대한 예측(2개 이상의 결과 중 하나 예측)을 생성하는 데 도움이 되는 프로세스입니다. 예를 들어, ML 모델이 '이 제품은 책인가요, 자동차인가요, 휴대폰인가요?' 또는 '이 고객이 가장 관심을 갖는 제품 범주는 무엇인가요?'라고 물을 수 있습니다.

## 변경 가능한 인프라

프로덕션 워크로드를 위해 기존 인프라를 업데이트하고 수정하는 모델입니다. 일관성, 안정성 및 예측 가능성을 개선하기 위해 AWS Well-Architected Framework는 [변경 불가능한](#) 인프라를 모범 사례로 사용할 것을 권장합니다.

## O

### OAC

[원본 액세스 제어를 참조하십시오.](#)

좋아요

[원본 액세스 ID를 참조하십시오.](#)

### OCM

[조직 변경 관리를 참조하십시오.](#)

### 오프라인 마이그레이션

마이그레이션 프로세스 중 소스 워크로드가 중단되는 마이그레이션 방법입니다. 이 방법은 가동 중지 증가를 수반하며 일반적으로 작고 중요하지 않은 워크로드에 사용됩니다.

## O

[운영 통합을 참조하십시오.](#)

안녕하세요.

[운영 수준 계약을 참조하십시오.](#)

### 온라인 마이그레이션

소스 워크로드를 오프라인 상태로 전환하지 않고 대상 시스템에 복사하는 마이그레이션 방법입니다. 워크로드에 연결된 애플리케이션은 마이그레이션 중에도 계속 작동할 수 있습니다. 이 방법은 가동 중지 차단 또는 최소화를 수반하며 일반적으로 중요한 프로덕션 워크로드에 사용됩니다.

### OPC-UA

[오픈 프로세스 커뮤니케이션 - 통합](#) 아키텍처를 참조하십시오.

### 오픈 프로세스 커뮤니케이션 - 통합 아키텍처 (OPC-UA)

산업 machine-to-machine 자동화를 위한 (M2M) 통신 프로토콜. OPC-UA는 데이터 암호화, 인증 및 권한 부여 체계와 함께 상호 운용성 표준을 제공합니다.

## 운영 수준 협약(OLA)

서비스 수준에 관한 계약(SLA)을 지원하기 위해 직무 IT 그룹이 서로에게 제공하기로 약속한 내용을 명확히 하는 계약입니다.

## 운영 준비 검토 (ORR)

인시던트 및 발생 가능한 실패의 범위를 이해, 평가, 예방 또는 줄이는 데 도움이 되는 질문 및 관련 모범 사례로 구성된 체크리스트입니다. 자세한 내용은 Well-Architected AWS 프레임워크의 [운영 준비 상태 검토 \(ORR\)](#) 를 참조하십시오.

## 운영 기술 (OT)

물리적 환경과 함께 작동하여 산업 운영, 장비 및 인프라를 제어하는 하드웨어 및 소프트웨어 시스템. 제조 분야에서는 OT와 정보 기술 (IT) 시스템의 통합이 [인더스트리 4.0](#) 혁신의 핵심 초점입니다.

## 운영 통합(OI)

클라우드에서 운영을 현대화하는 프로세스로 준비 계획, 자동화 및 통합을 수반합니다. 자세한 내용은 [운영 통합 가이드](#)를 참조하십시오.

## 조직 트레일

이를 통해 AWS CloudTrail 생성되는 트레일은 조직 AWS 계정 내 모든 사용자의 모든 이벤트를 기록합니다. AWS Organizations이 트레일은 조직에 속한 각 AWS 계정에 생성되고 각 계정의 활동을 추적합니다. 자세한 내용은 CloudTrail 설명서에서 [조직을 위한 트레일 만들기를](#) 참조하십시오.

## 조직 변경 관리(OCM)

사람, 문화 및 리더십 관점에서 중대하고 파괴적인 비즈니스 혁신을 관리하기 위한 프레임워크입니다. OCM은 변화 채택을 가속화하고, 과도기적 문제를 해결하고, 문화 및 조직적 변화를 주도함으로써 조직이 새로운 시스템 및 전략을 준비하고 전환할 수 있도록 지원합니다. 클라우드 채택 프로젝트에 필요한 변화 속도 때문에 AWS 마이그레이션 전략에서는 이 프레임워크를 사용자 가속화라고 합니다. 자세한 내용은 [사용 가이드](#)를 참조하십시오.

## 오리진 액세스 제어(OAC)

CloudFront에서는 Amazon Simple Storage Service (Amazon S3) 콘텐츠의 보안을 위해 액세스를 제한하는 향상된 옵션을 제공합니다. OAC는 모든 S3 버킷 AWS 리전, AWS KMS (SSE-KMS) 를 사용한 서버 측 암호화, S3 버킷에 대한 동적 및 요청을 모두 지원합니다. PUT DELETE

## 오리진 액세스 ID(OAI)

CloudFront에서는 Amazon S3 콘텐츠 보안을 위해 액세스를 제한하는 옵션입니다. OAI를 사용하면 Amazon S3가 인증할 수 있는 보안 주체를 CloudFront 생성합니다. 인증된 보안 주체는 특정 배

포를 통해서만 S3 버킷의 콘텐츠에 액세스할 수 있습니다. CloudFront 더 세분화되고 향상된 액세스 제어를 제공하는 [OAC](#)도 참조하십시오.

또는

[운영 준비 상태](#) 검토를 참조하십시오.

아니요

[운영 기술을](#) 참조하십시오.

아웃바운드(송신) VPC

AWS 다중 계정 아키텍처에서 애플리케이션 내에서 시작되는 네트워크 연결을 처리하는 VPC입니다. [AWS Security Reference Architecture](#)에서는 애플리케이션과 더 넓은 인터넷 간의 양방향 인터페이스를 보호하기 위해 인바운드, 아웃바운드 및 검사 VPC로 네트워크 계정을 설정할 것을 권장합니다.

## P

권한 경계

사용자나 역할이 가질 수 있는 최대 권한을 설정하기 위해 IAM 보안 주체에 연결되는 IAM 관리 정책입니다. 자세한 내용은 IAM 설명서의 [권한 경계](#)를 참조하십시오.

개인 식별 정보(PII)

직접 보거나 다른 관련 데이터와 함께 짝을 지을 때 개인의 신원을 합리적으로 추론하는 데 사용할 수 있는 정보입니다. PII의 예로는 이름, 주소, 연락처 정보 등이 있습니다.

PII

[개인 식별](#) 정보를 참조하십시오.

플레이북

클라우드에서 핵심 운영 기능을 제공하는 등 마이그레이션과 관련된 작업을 캡처하는 일련의 사전 정의된 단계입니다. 플레이북은 스크립트, 자동화된 런북 또는 현대화된 환경을 운영하는 데 필요한 프로세스나 단계 요약의 형태를 취할 수 있습니다.

PLC

[프로그래머블 로직 컨트롤러](#)를 참조하십시오.

## PLM

[제품 라이프사이클 관리](#)를 참조하십시오.

### 정책

권한을 정의 ([ID 기반 정책 참조](#)) 하거나, 액세스 조건을 지정 ([리소스 기반 정책 참조](#)) 하거나, 조직 내 모든 계정에 대한 최대 권한을 정의 AWS Organizations ([서비스 제어 정책 참조](#)) 할 수 있는 개체입니다.

### 다국어 지속성

데이터 액세스 패턴 및 기타 요구 사항을 기반으로 독립적으로 마이크로서비스의 데이터 스토리지 기술 선택. 마이크로서비스가 동일한 데이터 스토리지 기술을 사용하는 경우 구현 문제가 발생하거나 성능이 저하될 수 있습니다. 요구 사항에 가장 적합한 데이터 스토어를 사용하면 마이크로서비스를 더 쉽게 구현하고 성능과 확장성을 높일 수 있습니다. 자세한 내용은 [마이크로서비스에서 데이터 지속성 활성화](#)를 참조하십시오.

### 포트폴리오 평가

마이그레이션을 계획하기 위해 애플리케이션 포트폴리오를 검색 및 분석하고 우선순위를 정하는 프로세스입니다. 자세한 내용은 [마이그레이션 준비 상태 평가](#)를 참조하십시오.

### 조건자

일반적으로 조항에 있는 true false OR를 반환하는 쿼리 조건입니다. WHERE

### 조건부 푸시다운

전송하기 전에 쿼리의 데이터를 필터링하는 데이터베이스 쿼리 최적화 기법입니다. 이렇게 하면 관계형 데이터베이스에서 검색하고 처리해야 하는 데이터의 양이 줄어들고 쿼리 성능이 향상됩니다.

### 예방적 제어

이벤트 발생을 방지하도록 설계된 보안 제어입니다. 이 제어는 네트워크에 대한 무단 액세스나 원치 않는 변경을 방지하는 데 도움이 되는 1차 방어선입니다. 자세한 내용은 Implementing security controls on AWS의 [Preventative controls](#)를 참조하십시오.

### 보안 주체

작업을 수행하고 리소스에 액세스할 수 있는 AWS 있는 엔티티 이 엔티티는 일반적으로 IAM 역할의 루트 사용자 또는 사용자입니다. AWS 계정자세한 내용은 IAM 설명서의 [역할 용어 및 개념](#)의 보안 주체를 참조하십시오.

### 개인 정보 보호 중심 설계

전체 엔지니어링 프로세스에서 개인 정보를 고려하는 시스템 엔지니어링에서의 접근 방식입니다.

## 프라이빗 호스팅 영역

Amazon Route 53에서 하나 이상의 VPC 내 도메인과 하위 도메인에 대한 DNS 쿼리에 응답하는 방법에 대한 정보가 담긴 컨테이너입니다. 자세한 내용은 Route 53 설명서의 [프라이빗 호스팅 영역 작업을 참조하십시오](#).

## 사전 예방 제어

규정을 준수하지 않는 리소스의 배포를 방지하도록 설계된 [보안 제어입니다](#). 이러한 컨트롤은 리소스를 프로비저닝하기 전에 리소스를 스캔합니다. 리소스가 컨트롤과 호환되지 않으면 프로비저닝되지 않습니다. 자세한 내용은 AWS Control Tower 설명서의 [컨트롤 참조 안내서를 참조](#)하고 보안 제어 구현의 [사전 제어를 참조](#)하십시오. AWS

## 제품 라이프사이클 관리 (PLM)

설계, 개발, 출시부터 성장 및 성숙도, 폐기 및 제거에 이르는 전체 라이프사이클에 걸쳐 제품에 대한 데이터 및 프로세스를 관리하는 것입니다.

## 프로덕션 환경

[환경을 참조](#)하십시오.

## 프로그래머블 로직 컨트롤러 (PLC)

제조 분야에서 기계를 모니터링하고 제조 프로세스를 자동화하는 매우 안정적이고 적응력이 뛰어난 컴퓨터입니다.

## 가명화

데이터세트의 개인 식별자를 자리 표시자 값으로 바꾸는 프로세스입니다. 가명화는 개인 정보를 보호하는 데 도움이 될 수 있습니다. 가명화된 데이터는 여전히 개인 데이터로 간주됩니다.

## 게시/구독 (게시/구독)

마이크로서비스 간의 비동기 통신을 통해 확장성과 응답성을 개선할 수 있는 패턴입니다. 예를 들어 마이크로서비스 기반 [MES에서](#) 마이크로서비스는 다른 마이크로서비스가 구독할 수 있는 채널에 이벤트 메시지를 게시할 수 있습니다. 시스템은 게시 서비스를 변경하지 않고도 새 마이크로서비스를 추가할 수 있습니다.

## Q

### 쿼리 계획

SQL 관계형 데이터베이스 시스템의 데이터에 액세스하는 데 사용되는 일련의 단계 (예: 지침).

## 쿼리 계획 회귀

데이터베이스 서비스 최적화 프로그램이 데이터베이스 환경을 변경하기 전보다 덜 최적의 계획을 선택하는 경우입니다. 통계, 제한 사항, 환경 설정, 쿼리 파라미터 바인딩 및 데이터베이스 엔진 업데이트의 변경으로 인해 발생할 수 있습니다.

## R

### RACI 매트릭스

RACI ([책임, 책임, 상담, 정보 제공](#)) 를 참조하십시오.

### 랜섬웨어

결제 완료될 때까지 컴퓨터 시스템이나 데이터에 대한 액세스를 차단하도록 설계된 악성 소프트웨어입니다.

### RASCI 매트릭스

[책임, 책임, 상담, 정보 제공 \(RACI\)](#) 을 참조하십시오.

### RCAC

[행 및 열 액세스 제어](#) 를 참조하십시오.

### 읽기 전용 복제본

읽기 전용 용도로 사용되는 데이터베이스의 사본입니다. 쿼리를 읽기 전용 복제본으로 라우팅하여 기본 데이터베이스의 로드를 줄일 수 있습니다.

### 재설계

[7 R](#) 을 참조하십시오.

### Recovery Point Objective(RPO)

마지막 데이터 복구 시점 이후 허용되는 최대 시간입니다. 이에 따라 마지막 복구 시점과 서비스 중단 사이에 허용되는 데이터 손실로 간주되는 범위가 결정됩니다.

### Recovery Time Objective(RTO)

서비스 중단과 서비스 복원 사이의 허용 가능한 지연 시간입니다.

### 리팩터링

[7 R](#) 을 참조하십시오.

## 리전

지리적 AWS 영역별 리소스 모음. AWS 리전 각각은 격리되어 있고 서로 독립적이므로 내결함성, 안정성 및 복원력을 제공합니다. 자세한 내용은 [사용할 수 있는 AWS 리전 계정 지정을](#) 참조하십시오.

## 회귀

숫자 값을 예측하는 ML 기법입니다. 예를 들어, '이 집은 얼마에 팔릴까?'라는 문제를 풀기 위해 ML 모델은 선형 회귀 모델을 사용하여 주택에 대해 알려진 사실(예: 면적)을 기반으로 주택의 매매 가격을 예측할 수 있습니다.

## 리호스팅

[7 R을](#) 참조하십시오.

## release

배포 프로세스에서 변경 사항을 프로덕션 환경으로 승격시키는 행위입니다.

## 고쳐 놓다

[7 R을](#) 참조하십시오.

## 리플랫폼

[7 R을](#) 참조하십시오.

## 환매

[7 R을](#) 참조하십시오.

## 복원력

장애를 견디거나 장애를 복구할 수 있는 애플리케이션의 능력 [고가용성](#) 및 [재해 복구](#)는 복원력을 계획할 때 일반적으로 고려해야 할 사항입니다. AWS 클라우드 자세한 내용은 [AWS 클라우드 복원력을](#) 참조하십시오.

## 리소스 기반 정책

Amazon S3 버킷, 엔드포인트, 암호화 키 등의 리소스에 연결된 정책입니다. 이 유형의 정책은 액세스가 허용된 보안 주체, 지원되는 작업 및 충족해야 하는 기타 조건을 지정합니다.

## RACI(Responsible, Accountable, Consulted, Informed) 매트릭스

마이그레이션 활동 및 클라우드 운영에 참여하는 모든 당사자의 역할과 책임을 정의하는 매트릭스입니다. 매트릭스 이름은 매트릭스에 정의된 책임 유형에서 파생됩니다. 실무 담당자 (R), 의사 결

정권자 (A), 업무 수행 조연자 (C), 결과 통보 대상자 (I). 지원자는 (S) 선택사항입니다. 지원자를 포함하면 매트릭스를 RASCI 매트릭스라고 하고, 지원자를 제외하면 RACI 매트릭스라고 합니다.

## 대응 제어

보안 기준에서 벗어나거나 부정적인 이벤트를 해결하도록 설계된 보안 제어입니다. 자세한 내용은 [Implementing security controls on AWS의 Responsive controls](#)를 참조하십시오.

## retain

[7 R](#)을 참조하십시오.

## 은퇴

[7 R](#)을 참조하십시오.

## 회전

공격자가 자격 증명에 액세스하는 것을 더 어렵게 만들기 위해 [암호](#)를 주기적으로 업데이트하는 프로세스입니다.

## 행 및 열 액세스 제어(RCAC)

액세스 규칙이 정의된 기본적이고 유연한 SQL 표현식을 사용합니다. RCAC는 행 권한과 열 마스크로 구성됩니다.

## RPO

[복구 지점 목표를](#) 참조하십시오.

## RTO

[복구 시간 목표를](#) 참조하십시오.

## 런복

특정 작업을 수행하는 데 필요한 일련의 수동 또는 자동 절차입니다. 일반적으로 오류율이 높은 반복 작업이나 절차를 간소화하기 위해 런복을 만듭니다.

# S

## SAML 2.0

많은 ID 제공업체 (IdPs) 가 사용하는 개방형 표준입니다. 이 기능을 사용하면 페더레이션 싱글 사인온 (SSO) 이 가능하므로 조직의 모든 사용자를 위해 IAM에서 사용자를 생성하지 않고도 사용자가 AWS API 작업에 AWS Management Console 로그인하거나 API 작업을 호출할 수 있습니다.

SAML 2.0 기반 페더레이션에 대한 자세한 내용은 IAM 설명서의 [SAML 2.0 기반 페더레이션 정보](#)를 참조하십시오.

## SCADA

[감독 제어 및 데이터 수집](#)을 참조하십시오.

## SCP

[서비스 제어 정책](#)을 참조하십시오.

## secret

에는 AWS Secrets Manager 암호화된 형태로 저장하는 비밀번호나 사용자 자격 증명과 같은 기밀 또는 제한된 정보. 비밀 값과 해당 메타데이터로 구성됩니다. 비밀 값은 바이너리, 단일 문자열 또는 여러 문자열일 수 있습니다. 자세한 내용은 [Secrets Manager 시크릿에는 무엇이 들어 있나요?](#)를 참조하십시오. Secrets Manager 설명서에서 확인할 수 있습니다.

## 보안 제어

위험 행위자가 보안 취약성을 악용하는 능력을 방지, 탐지 또는 감소시키는 기술적 또는 관리적 가드레일입니다. [보안 제어에는 예방적, 탐정적, 대응적, 사전 예방적 제어의 네 가지 기본 유형이 있습니다.](#)

## 보안 강화

공격 표면을 줄여 공격에 대한 저항력을 높이는 프로세스입니다. 더 이상 필요하지 않은 리소스 제거, 최소 권한 부여의 보안 모범 사례 구현, 구성 파일의 불필요한 기능 비활성화 등의 작업이 여기에 포함될 수 있습니다.

## 보안 정보 및 이벤트 관리(SIEM) 시스템

보안 정보 관리(SIM)와 보안 이벤트 관리(SEM) 시스템을 결합하는 도구 및 서비스입니다. SIEM 시스템은 서버, 네트워크, 디바이스 및 기타 소스에서 데이터를 수집, 모니터링 및 분석하여 위협과 보안 침해를 탐지하고 알림을 생성합니다.

## 보안 대응 자동화

보안 이벤트에 자동으로 대응하거나 보안 이벤트를 해결하도록 설계된 사전 정의되고 프로그래밍된 조치입니다. 이러한 자동화는 보안 모범 사례를 구현하는 데 도움이 되는 [탐지](#) 또는 [대응형](#) 보안 제어 역할을 합니다. AWS 자동 응답 조치의 예로는 VPC 보안 그룹 수정, Amazon EC2 인스턴스 패치, 자격 증명 교체 등이 있습니다.

## 서버 측 암호화

수신자에 의한 목적지의 데이터 암호화 AWS 서비스

## 서비스 제어 정책(SCP)

AWS Organizations에 속한 조직의 모든 계정에 대한 권한을 중앙 집중식으로 제어하는 정책입니다. SCP는 관리자가 사용자 또는 역할에 위임할 수 있는 작업에 대해 제한을 설정하거나 가드레일을 정의합니다. SCP를 허용 목록 또는 거부 목록으로 사용하여 허용하거나 금지할 서비스 또는 작업을 지정할 수 있습니다. 자세한 내용은 AWS Organizations 설명서의 [서비스 제어 정책을](#) 참조하십시오.

## 서비스 엔드포인트

의 진입점 URL입니다 AWS 서비스. 엔드포인트를 사용하여 대상 서비스에 프로그래밍 방식으로 연결할 수 있습니다. 자세한 내용은 AWS 일반 참조의 [AWS 서비스 엔드포인트](#)를 참조하십시오.

## 서비스 수준에 관한 계약(SLA)

IT 팀이 고객에게 제공하기로 약속한 내용(예: 서비스 가동 시간 및 성능)을 명시한 계약입니다.

## 서비스 수준 표시기 (SLI)

오류율, 가용성 또는 처리량과 같은 서비스의 성능 측면을 측정하는 것입니다.

## 서비스 수준 목표 (SLO)

[서비스 수준 지표로 측정되는 서비스 상태를 나타내는 대상 지표입니다.](#)

## 공동 책임 모델

클라우드 보안 및 규정 준수에 AWS 대한 책임을 공유하는 것을 설명하는 모델입니다. AWS 클라우드의 보안을 책임지는 반면, 사용자는 클라우드에서의 보안을 담당합니다. 자세한 내용은 [공동 책임 모델](#)을 참조하십시오.

## 시앰

[보안 정보 및 이벤트 관리 시스템을](#) 참조하십시오.

## 단일 장애 지점 (SPOF)

응용 프로그램의 중요한 단일 구성 요소에서 발생한 오류로 인해 시스템이 중단될 수 있습니다.

## SLA

SLA ([서비스 수준 계약](#)) 를 참조하십시오.

## SLI

[서비스 수준 표시기](#) 참조.

## SLO

[서비스 수준 목표를](#) 참조하십시오.

## split-and-seed 모델

현대화 프로젝트를 확장하고 가속화하기 위한 패턴입니다. 새로운 기능과 제품 릴리스가 정의되면 핵심 팀이 분할되어 새로운 제품 팀이 만들어집니다. 이를 통해 조직의 역량과 서비스 규모를 조정하고, 개발자 생산성을 개선하고, 신속한 혁신을 지원할 수 있습니다. 자세한 내용은 [의 애플리케이션 현대화를 위한 단계별 접근 방식을 참조하십시오. AWS 클라우드](#)

## SPOF

[단일 장애 지점 보기.](#)

## 스타 스키마

하나의 큰 팩트 테이블을 사용하여 트랜잭션 또는 측정 데이터를 저장하고 하나 이상의 작은 차원 테이블을 사용하여 데이터 속성을 저장하는 데이터베이스 구성 구조입니다. 이 구조는 [데이터 웨어하우스에서](#) 사용하거나 비즈니스 인텔리전스 용도로 설계되었습니다.

## Strangler Fig 패턴

레거시 시스템을 폐기할 수 있을 때까지 시스템 기능을 점진적으로 다시 작성하고 교체하여 모놀리식 시스템을 현대화하기 위한 접근 방식. 이 패턴은 무화과 덩굴이 나무로 자라 결국 숙주를 압도하고 대체하는 것과 비슷합니다. [Martin Fowler](#)가 모놀리식 시스템을 다시 작성할 때 위험을 관리하는 방법으로 이 패턴을 도입했습니다. 이 패턴을 적용하는 방법의 예는 [컨테이너 및 Amazon API Gateway를 사용하여 기존의 Microsoft ASP.NET\(ASMX\) 웹 서비스를 점진적으로 현대화하는 방법을 참조하십시오.](#)

## 서브넷

VPC의 IP 주소 범위입니다. 서브넷은 단일 가용 영역에 상주해야 합니다.

## 감독 통제 및 데이터 수집 (SCADA)

제조 시 하드웨어와 소프트웨어를 사용하여 물리적 자산과 생산 작업을 모니터링하는 시스템입니다.

## 대칭 암호화

동일한 키를 사용하여 데이터를 암호화하고 복호화하는 암호화 알고리즘입니다.

## 합성 테스트

잠재적 문제를 감지하거나 성능을 모니터링하기 위해 사용자 상호 작용을 시뮬레이션하는 방식으로 시스템을 테스트합니다. [Amazon CloudWatch Synthetics](#)를 사용하여 이러한 테스트를 생성할 수 있습니다.

# T

## tags

리소스 구성을 위한 메타데이터 역할을 하는 키-값 쌍. AWS 태그를 사용하면 리소스를 손쉽게 관리, 식별, 정리, 검색 및 필터링할 수 있습니다. 자세한 내용은 [AWS 리소스에 태그 지정](#)을 참조하십시오.

## 대상 변수

지도 ML에서 예측하려는 값으로, 결과 변수라고도 합니다. 예를 들어, 제조 설정에서 대상 변수는 제품 결함일 수 있습니다.

## 작업 목록

런북을 통해 진행 상황을 추적하는 데 사용되는 도구입니다. 작업 목록에는 런북의 개요와 완료해야 할 일반 작업 목록이 포함되어 있습니다. 각 일반 작업에 대한 예상 소요 시간, 소유자 및 진행 상황이 작업 목록에 포함됩니다.

## 테스트 환경

[환경을 참조하십시오.](#)

## 훈련

ML 모델이 학습할 수 있는 데이터를 제공하는 것입니다. 훈련 데이터에는 정답이 포함되어야 합니다. 학습 알고리즘은 훈련 데이터에서 대상(예측하려는 답)에 입력 데이터 속성을 매핑하는 패턴을 찾고, 이러한 패턴을 캡처하는 ML 모델을 출력합니다. 그런 다음 ML 모델을 사용하여 대상을 모르는 새 데이터에 대한 예측을 할 수 있습니다.

## 전송 게이트웨이

VPC와 온프레미스 네트워크를 상호 연결하는 데 사용할 수 있는 네트워크 전송 허브입니다. 자세한 내용은 AWS Transit Gateway 설명서의 [트랜짓 게이트웨이란 무엇입니까?](#)를 참조하십시오.

## 트렁크 기반 워크플로

개발자가 기능 브랜치에서 로컬로 기능을 구축하고 테스트한 다음 해당 변경 사항을 기본 브랜치에 병합하는 접근 방식입니다. 이후 기본 브랜치는 개발, 프로덕션 이전 및 프로덕션 환경에 순차적으로 구축됩니다.

## 신뢰할 수 있는 액세스

조직 내 AWS Organizations 및 해당 계정에서 사용자를 대신하여 작업을 수행하도록 지정한 서비스에 권한 부여 신뢰할 수 있는 서비스는 필요할 때 각 계정에 서비스 연결 역할을 생성하여 관

리 작업을 수행합니다. 자세한 내용은 AWS Organizations 설명서의 [다른 AWS 서비스와 AWS Organizations 함께 사용](#)을 참조하십시오.

## 튜닝

ML 모델의 정확도를 높이기 위해 훈련 프로세스의 측면을 여러 변경하는 것입니다. 예를 들어, 레이블링 세트를 생성하고 레이블을 추가한 다음 다양한 설정에서 이러한 단계를 여러 번 반복하여 모델을 최적화하는 방식으로 ML 모델을 훈련할 수 있습니다.

## 피자 두 판 팀

피자 두 판만 들고 배블리 먹을 수 있는 소규모 DevOps 팀. 피자 두 판 팀 규모는 소프트웨어 개발에 있어 가능한 최상의 공동 작업 기회를 보장합니다.

# U

## 불확실성

예측 ML 모델의 신뢰성을 저해할 수 있는 부정확하거나 불완전하거나 알려지지 않은 정보를 나타내는 개념입니다. 불확실성에는 두 가지 유형이 있습니다. 인식론적 불확실성은 제한적이고 불완전한 데이터에 의해 발생하는 반면, 우연한 불확실성은 데이터에 내재된 노이즈와 무작위성에 의해 발생합니다. 자세한 내용은 [Quantifying uncertainty in deep learning systems](#) 가이드를 참조하십시오.

## 차별화되지 않은 작업

애플리케이션을 만들고 운영하는 데 필요하지만 최종 사용자에게 직접적인 가치를 제공하거나 경쟁 우위를 제공하지 못하는 작업을 헤비 리프팅이라고도 합니다. 차별화되지 않은 작업의 예로는 조달, 유지보수, 용량 계획 등이 있습니다.

## 상위 환경

[환경을](#) 보세요.

# V

## 정리

스토리지를 회수하고 성능을 향상시키기 위해 증분 업데이트 후 정리 작업을 수반하는 데이터베이스 유지 관리 작업입니다.

## 버전 제어

리포지토리의 소스 코드 변경과 같은 변경 사항을 추적하는 프로세스 및 도구입니다.

## VPC 피어링

프라이빗 IP 주소를 사용하여 트래픽을 라우팅할 수 있게 하는 두 VPC 간의 연결입니다. 자세한 내용은 Amazon VPC 설명서의 [VPC 피어링이란?](#)을 참조하십시오.

## 취약성

시스템 보안을 손상시키는 소프트웨어 또는 하드웨어 결함입니다.

# W

## 웹 캐시

자주 액세스하는 최신 관련 데이터를 포함하는 버퍼 캐시입니다. 버퍼 캐시에서 데이터베이스 인스턴스를 읽을 수 있기 때문에 주 메모리나 디스크에서 읽는 것보다 빠릅니다.

## 웹 데이터

자주 액세스하지 않는 데이터입니다. 이런 종류의 데이터를 쿼리할 때는 일반적으로 적절히 느린 쿼리가 허용됩니다.

## 윈도우 함수

현재 레코드와 어떤 식으로든 관련된 행 그룹에 대해 계산을 수행하는 SQL 함수입니다. 윈도우 함수는 이동 평균을 계산하거나 현재 행의 상대적 위치를 기반으로 행 값에 액세스하는 등의 작업을 처리하는 데 유용합니다.

## 워크로드

고객 대면 애플리케이션이나 백엔드 프로세스 같이 비즈니스 가치를 창출하는 리소스 및 코드 모음입니다.

## 워크스트림

마이그레이션 프로젝트에서 특정 작업 세트를 담당하는 직무 그룹입니다. 각 워크스트림은 독립적이지만 프로젝트의 다른 워크스트림을 지원합니다. 예를 들어, 포트폴리오 워크스트림은 애플리케이션 우선순위 지정, 웨이브 계획, 마이그레이션 메타데이터 수집을 담당합니다. 포트폴리오 워크스트림은 이러한 자산을 마이그레이션 워크스트림에 전달하고, 마이그레이션 워크스트림은 서버와 애플리케이션을 마이그레이션합니다.

## 원

한 번 쓰고, 많이 읽으세요.

### WQF

AWS 워크로드 검증 프레임워크를 참조하십시오.

#### 한 번 작성하고 여러 번 읽기 (WORM)

데이터를 한 번 쓰고 데이터가 삭제되거나 수정되지 않도록 하는 스토리지 모델입니다. 인증된 사용자는 필요한 만큼 데이터를 여러 번 읽을 수 있지만 변경할 수는 없습니다. 이 데이터 스토리지 인 프라는 변경할 수 없는 것으로 간주됩니다.

## Z

### 제로데이 익스플로잇

제로데이 취약점을 악용하는 공격 (일반적으로 멀웨어)입니다.

#### 제로데이 취약성

프로덕션 시스템의 명백한 결함 또는 취약성입니다. 위협 행위자는 이러한 유형의 취약성을 사용하여 시스템을 공격할 수 있습니다. 개발자는 공격의 결과로 취약성을 인지하는 경우가 많습니다.

#### 좀비 애플리케이션

평균 CPU 및 메모리 사용량이 5% 미만인 애플리케이션입니다. 마이그레이션 프로젝트에서는 이러한 애플리케이션을 사용 중지하는 것이 일반적입니다.

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