



Resilience analysis framework

# AWS 규범적 지침



# AWS 규범적 지침: Resilience analysis framework

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
프레임워크 개요 .....	2
워크로드 이해 .....	5
프레임워크 적용 .....	7
잠재적 장애 완화 .....	10
장단점 및 위험에 대한 이해 .....	10
장애 모드 관찰 가능성 .....	12
일반적인 완화 전략 .....	12
지속적 개선 .....	17
결론 및 리소스 .....	18
문서 기록 .....	19
용어집 .....	20
# .....	20
A .....	21
B .....	23
C .....	25
D .....	28
E .....	32
F .....	34
G .....	35
H .....	36
I .....	38
L .....	40
M .....	41
O .....	45
P .....	47
Q .....	50
R .....	50
S .....	53
T .....	56
U .....	58
V .....	58
W .....	59
Z .....	60

---

..... lxi

# 레질리언스 분석 프레임워크

존 포르멘토, 브루노 에머, 스티븐 후퍼, 제이슨 바르토, 마이클 하켄, 아마존 웹 서비스 (AWS)

2023년 9월([문서 기록](#))

일관되고 반복 가능한 표준 및 프로세스는 지속적인 개선의 중요한 부분입니다. 이는 분산 시스템의 복원력에서도 마찬가지입니다. 이 지침의 목적은 장애 모드를 분석하고 워크로드에 미치는 영향을 일관되게 분석하는 방법을 제공하는 복원력 분석 프레임워크를 소개하는 것입니다. 설계에서 운영에 이르기까지 워크로드의 라이프사이클 전반에 걸쳐 이 프레임워크를 사용하면 일관되고 반복 가능한 방식으로 광범위한 잠재적 장애 모드에 대한 워크로드의 복원력을 지속적으로 개선할 수 있습니다. 이를 통해 레질리언스 목표를 달성하고 워크로드의 원하는 레질리언스 속성을 유지할 수 있습니다.

이 프레임워크는 여러 업계의 고객과 함께 작업한 AWS 솔루션 아키텍처 현장 팀의 경험을 바탕으로 개발되었습니다. 제품 관리자, 소프트웨어 개발자, 시스템 엔지니어, 운영 팀, 설계자 등 다양한 직함을 가질 수 있는 건축업자를 대상으로 합니다. 이들은 분석 대상 시스템, 서비스 또는 제품에 대해 가장 잘 아는 사람들입니다. 지속적인 연습에서 프레임워크를 사용하면 점진적인 진전을 이루고 장기적인 탄력성 목표를 달성하는 데 도움이 될 수 있습니다.

프레임워크의 초점은 잠재적 장애 모드와 그 영향을 완화하는 데 사용할 수 있는 예방 및 수정 조치를 식별하는 것입니다. 종속성의 오류율 증가와 같이 사용자가 직접 제어할 수 없는 구성 요소에서 장애가 발생하더라도 이러한 장애가 워크로드에 미치는 영향과 이러한 장애에 대응하도록 해당 워크로드를 설계하는 방법을 고려해야 합니다. 궁극적으로는 다음 사항에 집중해야 합니다. 대응할 수 있는 장애 제어할 수 있는 완화 방법을 사용함으로써

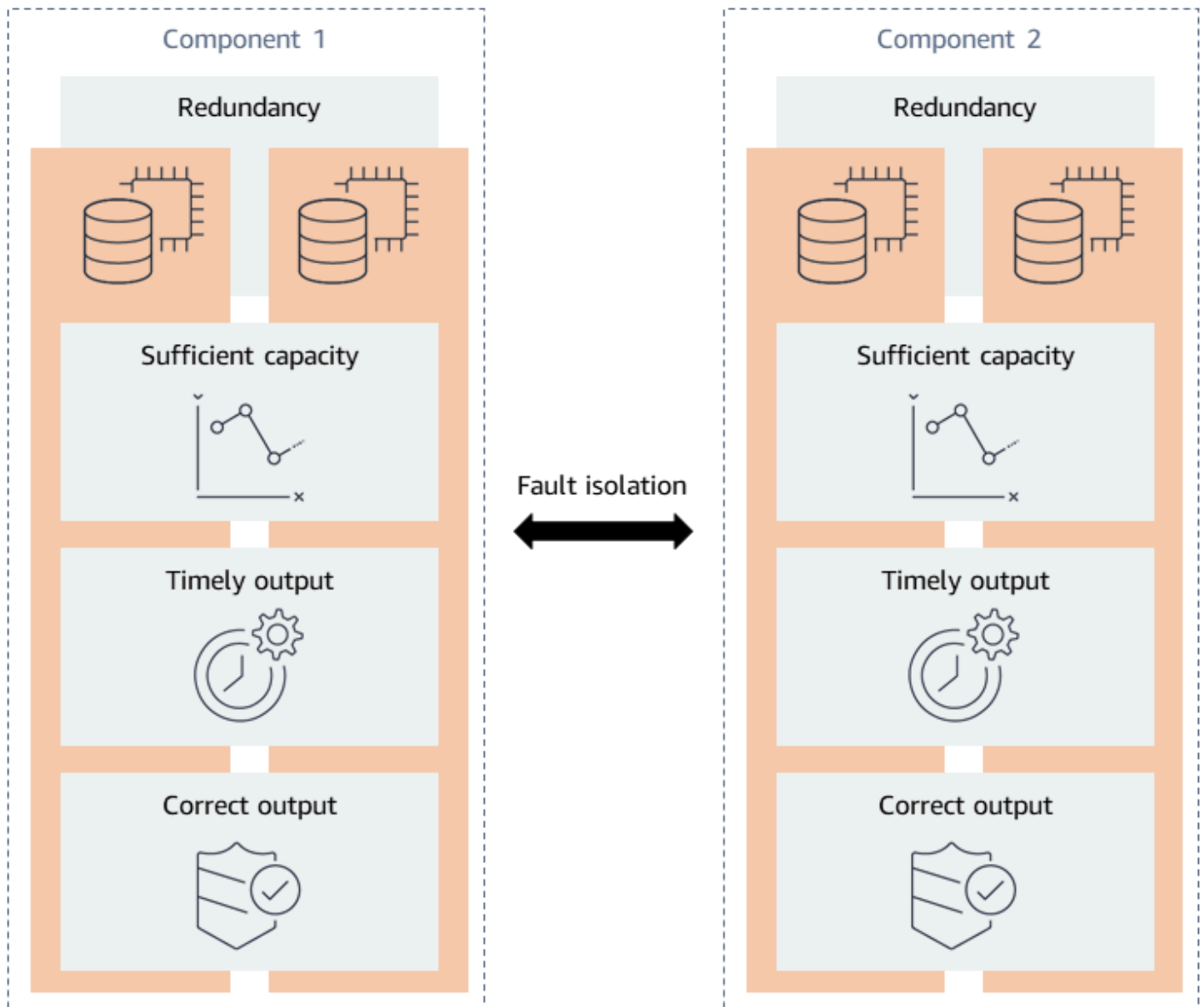
이 가이드에서는 프레임워크를 개괄적으로 설명한 다음 워크로드를 식별하고 문서화하는 방법, 해당 워크로드에 프레임워크를 적용하는 방법, 발견된 잠재적 장애에 대한 완화 전략을 평가하는 방법을 설명합니다.

## 목차

- [프레임워크 개요](#)
- [워크로드에 대한 이해](#)
- [프레임워크 적용](#)
- [잠재적 장애 완화](#)
- [결론 및 리소스](#)

## 프레임워크 개요

복원력 분석 프레임워크는 워크로드의 원하는 복원력 속성을 식별하여 개발되었습니다. 원하는 속성이란 시스템에서 사실로 표현하고자 하는 내용을 말합니다. 복원력은 일반적으로 가용성으로 측정되므로 가용성이 높은 분산 시스템의 특징은 이중화, 충분한 용량, 적시 출력, 올바른 출력, 오류 격리 등 5가지 속성입니다. 이러한 속성은 다음 다이어그램에 나와 있습니다.



- 리던던시— 단일 장애 지점 (SPOF) 을 제거하는 이중화를 통해 내결함성이 달성됩니다. 중복성은 워크로드의 예비 구성 요소부터 전체 애플리케이션 스택의 전체 복제본에 이르기까지 다양합니다. 애플리케이션의 중복성을 고려할 때는 사용하는 인프라, 데이터 저장소 및 종속성이 제공하는 중복

성 수준을 고려하는 것이 중요합니다. 예를 들어, Amazon DynamoDB 및 Amazon 심플 스토리지 서비스 (Amazon S3) 는 한 지역의 여러 가용 영역에 데이터를 복제하여 중복성을 제공합니다. AWS Lambda 여러 가용 영역의 여러 작업자 노드에서 함수를 실행합니다. 사용하는 각 서비스에 대해 서비스에서 제공하는 내용과 설계에 필요한 사항을 고려하십시오.

- 충분한 용량— 워크로드가 의도한 대로 작동하려면 충분한 리소스가 필요합니다. 리소스에는 메모리, CPU 사이클, 스투드, 스토리지, 처리량, 서비스 할당량 등이 포함됩니다.
- 적시 출력— 고객은 워크로드를 사용할 때 적절한 시간 내에 워크로드가 의도한 기능을 수행할 것으로 기대합니다. 서비스가 지연 시간에 대한 SLA (서비스 수준 계약) 를 제공하지 않는 한 고객의 기대치는 일반적으로 경험적 증거, 즉 고객 경험을 기반으로 합니다. 이는 평균적인 고객 경험 일반적으로 시스템의 평균 지연 시간 (P50) 으로 간주됩니다. 워크로드가 예상보다 오래 걸리는 경우 이 지연 시간이 고객 경험에 영향을 미칠 수 있습니다.
- 올바른 출력— 의도한 기능을 제공하려면 워크로드 소프트웨어의 올바른 출력이 필요합니다. 부정확하거나 불완전한 결과는 응답이 없는 것보다 더 나쁠 수 있습니다.
- 장애 격리— 장애 격리는 장애 발생 시 의도한 장애 컨테이너로 영향 범위를 제한합니다. 이를 통해 워크로드의 특정 구성 요소가 함께 장애를 일으키도록 하는 동시에 장애가 의도하지 않은 다른 구성 요소로 연쇄적으로 발생하는 것을 방지합니다. 또한 워크로드가 고객에게 미치는 영향의 범위를 제한하는 데도 도움이 됩니다. 장애 격리는 장애가 이미 발생했지만 억제되어야 한다는 점을 인정하기 때문에 이전의 네 가지 속성과는 다소 다릅니다. 인프라, 종속성 및 소프트웨어 기능에 장애 격리를 만들 수 있습니다.

원하는 속성을 위반하면 워크로드를 사용할 수 없거나 사용할 수 없는 것으로 인식될 수 있습니다. 이러한 원하는 복원력 속성 및 여러 업체와 협력한 경험을 기반으로 합니다. AWS 고객을 대상으로 단일 장애 지점, 과도한 부하, 과도한 지연 시간, 잘못된 구성 및 버그, 공유 운명 등 5가지 일반적인 장애 범주를 식별했습니다. 이를 줄여서 SEEMS라고 부릅니다. 이는 잠재적 장애 모드를 분류하는 일관된 방법을 제공하며 다음 표에 설명되어 있습니다.

실패 카테고리	위반	정의
단일 장애 지점 (SPOF)	중복성	단일 구성 요소에 장애가 발생하면 구성 요소의 이중화 부족으로 인해 시스템이 중단됩니다.
과도한 부하	충분한 용량	과도한 수요나 트래픽으로 인해 리소스가 과도하게 소비되면 리소스가 예상한 기능을 수

행할 수 없습니다. 여기에는 한도 및 할당량 도달이 포함될 수 있으며, 이로 인해 병목 현상 및 요청 거부가 발생할 수 있습니다.

과도한 지연 시간

적시 출력

시스템 처리 또는 네트워크 트래픽 지연 시간이 예상 시간, SLO (서비스 수준 목표) 또는 SLA (서비스 수준 계약) 를 초과합니다.

잘못된 구성 및 버그

올바른 출력

소프트웨어 버그 또는 시스템 구성 오류로 인해 출력이 잘못될 수 있습니다.

공유 운명

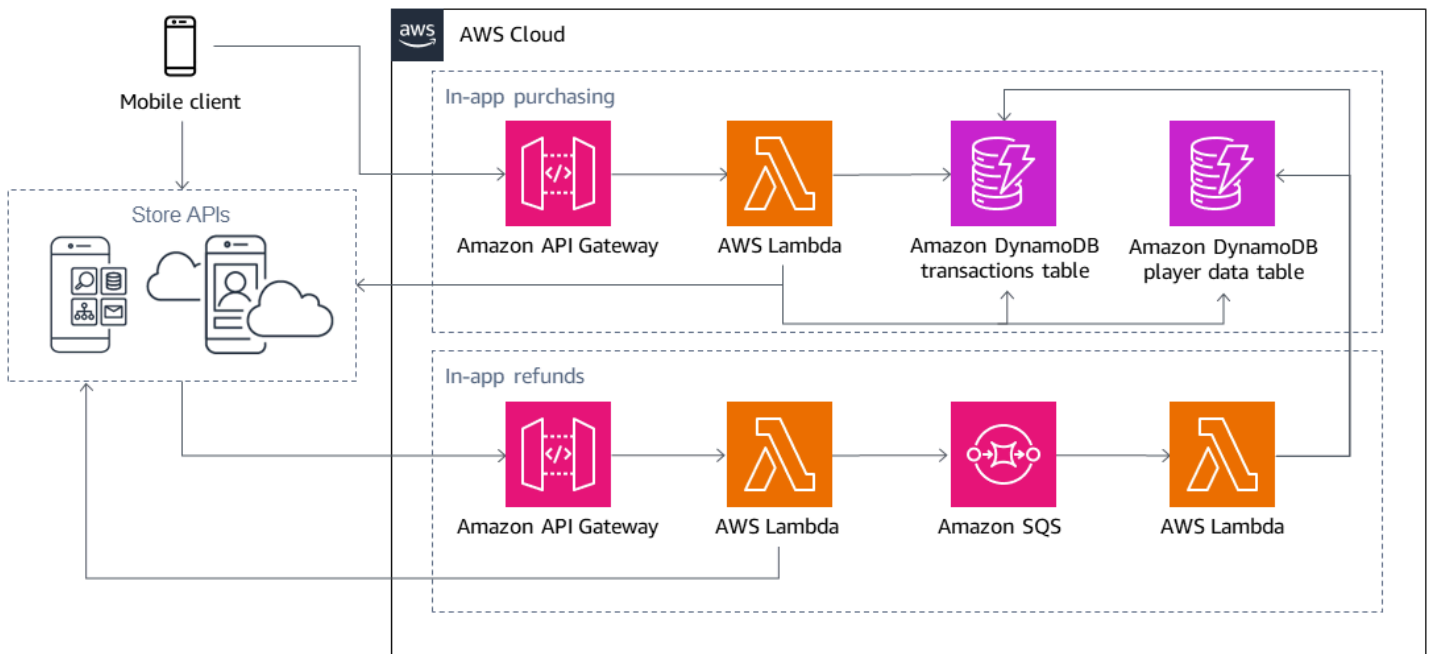
장애 격리

이전 장애 범주 중 하나로 인해 발생한 장애는 의도된 장애 격리 범위를 넘어 시스템의 다른 부분이나 다른 고객에게 영향을 미칩니다.



# 워크로드 이해

프레임워크를 적용하려면 먼저 분석하려는 워크로드를 이해해야 합니다. 시스템 아키텍처 다이어그램은 시스템의 가장 관련성이 높은 세부 정보를 문서화하기 위한 출발점을 제공합니다. 그러나 많은 시스템에 수많은 구성 요소와 상호 작용이 있기 때문에 전체 워크로드를 분석하는 것은 복잡할 수 있습니다. 대신 다음 사항에 집중하는 것이 좋습니다. **사용자 스토리**, 최종 사용자의 관점에서 작성된 소프트웨어 기능에 대한 비공식적이고 일반적인 설명. 소프트웨어 기능이 고객에게 가치를 제공하는 방법을 설명하는 것이 목적입니다. 그런 다음 아키텍처 다이어그램과 데이터 흐름 다이어그램을 사용하여 이러한 사용자 스토리를 모델링하여 설명된 비즈니스 기능을 제공하는 기술 구성 요소를 더 쉽게 평가할 수 있습니다. 예를 들어, 다음 다이어그램과 같이 인앱 모바일 게임 구매 솔루션에는 '인앱 크레딧 구매'와 '인앱 환불 받기'라는 두 개의 사용자 스토리가 있을 수 있습니다. (이 예제 아키텍처는 시스템을 사용자 스토리로 분해하는 방법을 중점적으로 설명하지만 복원력이 뛰어난 애플리케이션을 나타내기 위한 것은 아닙니다.)



각 사용자 스토리는 코드 및 구성, 인프라, 데이터 저장소, 외부 종속성 등 4가지 공통 구성 요소로 구성됩니다. 다이어그램에는 이러한 구성 요소가 모두 포함되어야 하며 구성 요소 간의 상호 작용이 반영되어야 합니다. 예를 들어, Amazon API Gateway 엔드포인트에 과도한 부하가 발생하는 경우 해당 부하가 시스템의 다른 구성 요소 (예:) 에 어떻게 캐스케이드되는지 생각해 보십시오. AWS Lambda 함수 또는 Amazon DynamoDB 테이블. 이러한 상호 작용을 추적하면 장애 모드가 사용자 스토리에 어떤 영향을 미칠 수 있는지 이해하는 데 도움이 됩니다. 이전 그림과 같이 데이터 흐름도를 사용하거나 아키텍처 다이어그램에서 간단한 흐름 화살표를 사용하여 이 흐름을 시각적으로 캡처할 수 있습니다. 각 구성 요소에 대해 전송되는 정보의 유형, 수신되는 정보, 통신이 동기식인지 비동기식인지 여부, 넘어가는

장애 경계와 같은 세부 정보를 캡처해 보세요. 이 예제에서 DynamoDB 테이블은 두 사용자 스토리에서 공유됩니다. 화살표를 보면 인앱 환불 스토리의 Lambda 구성 요소가 인앱 구매 스토리의 DynamoDB 테이블에 액세스한다는 것을 알 수 있습니다. 즉, 인앱 구매 사용자 스토리로 인한 실패가 운명 공유의 결과로 인앱 환불 사용자 스토리로 연쇄적으로 이어질 수 있습니다.

또한 각 구성 요소의 기본 구성을 이해하는 것도 중요합니다. 기본 구성은 초당 평균 및 최대 트랜잭션 수, 페이로드의 최대 크기, 클라이언트 제한 시간, 리소스의 기본 또는 현재 서비스 할당량과 같은 제약 조건을 식별합니다. 새 설계를 모델링하는 경우 설계의 기능 요구 사항을 문서화하고 한계를 고려하는 것이 좋습니다. 이렇게 하면 구성 요소에서 고장 모드가 어떻게 나타날 수 있는지 이해하는 데 도움이 됩니다.

마지막으로, 사용자 스토리가 제공하는 비즈니스 가치를 기반으로 사용자 스토리의 우선 순위를 정해야 합니다. 이 우선 순위를 지정하면 워크로드의 가장 중요한 기능에 먼저 집중할 수 있습니다. 그런 다음 해당 기능의 중요 경로에 속하는 워크로드 구성 요소에 분석을 집중하여 프레임워크를 활용하여 더 빠르게 가치를 실현할 수 있습니다. 프로세스를 반복하면서 우선 순위가 다른 추가 사용자 사례를 검토할 수 있습니다.

## 프레임워크 적용

복원력 분석 프레임워크를 적용하는 가장 좋은 방법은 분석 중인 사용자 스토리의 각 구성 요소에 대해 질문해야 하는 실패 범주별로 정리된 표준 질문 집합으로 시작하는 것입니다. 일부 질문이 워크로드의 모든 구성 요소에 해당되지 않는 경우 가장 적절한 질문을 사용하세요.

장애 모드에 대해 다음과 같은 두 가지 관점에서 생각해 볼 수 있습니다.

- 실패는 사용자 스토리를 지원하는 구성 요소의 기능에 어떤 영향을 미칩니까?
- 실패는 구성 요소와 다른 구성 요소의 상호 작용에 어떤 영향을 미칩니까?

예를 들어 데이터 저장소와 과도한 부하를 고려할 때 데이터베이스에 과도한 부하가 걸리고 쿼리 제한 시간이 초과되는 장애 모드를 생각할 수 있습니다. 또한 데이터베이스 클라이언트가 재시도로 인해 데이터베이스 과부하를 일으키거나 데이터베이스 연결을 닫지 못해 연결 풀이 고갈될 수도 있다는 점을 생각해 볼 수도 있습니다. 또 다른 예로 여러 단계로 구성된 인증 프로세스를 들 수 있습니다. 다단계 인증 (MFA) 애플리케이션 또는 타사 ID 공급자 (IdP) 의 장애가 이 인증 시스템의 사용자 스토리에 어떤 영향을 미칠 수 있는지 생각해 봐야 합니다.

다음 질문에 답할 때는 실패의 원인을 고려해야 합니다. 예를 들어 고객 급증이나 유지 보수 작업 중에 너무 많은 노드를 서비스 중단시킨 작업자 때문에 과부하가 발생했습니까? 각 질문에서 여러 장애 원인을 식별할 수 있으며, 이를 위해서는 각기 다른 완화 조치가 필요할 수 있습니다. 질문을 할 때 발견한 잠재적 고장 모드, 해당 모드가 적용되는 구성 요소, 각 고장의 원인을 기록해 두십시오.

### 단일 장애 지점

- 구성 요소가 이중화를 위해 설계되었나요?
- 구성 요소에 장애가 발생하면 어떻게 됩니까?
- 애플리케이션이 단일 가용 영역의 부분적 또는 전체 손실을 감내할 수 있습니까?

### 과도한 지연 시간

- 이 구성 요소의 지연 시간이 길어지거나 상호 작용하는 구성 요소의 대기 시간이 길어지면 (또는 TCP 재설정과 같은 네트워크 중단) 어떻게 됩니까?
- 재시도 전략을 통해 제한 시간을 적절하게 구성했습니까?
- 실패가 빠르나요, 느리게 실패하나요? 빠른 속도로 장애가 발생하여 손상된 리소스로 모든 트래픽이 의도치 않게 전송되는 등의 연쇄적인 영향이 있습니까?

- 이 구성 요소에 대한 요청 중 가장 비용이 많이 드는 것은 무엇입니까?

### 과도한 부하

- 이 구성 요소를 압도할 수 있는 요인은 무엇입니까? 이 구성 요소가 다른 구성 요소를 어떻게 압도할 수 있습니까?
- 결코 성공하지 못할 일에 자원을 낭비하는 것을 어떻게 방지할 수 있을까요?
- 구성 요소에 맞게 구성된 회로 차단기가 있습니까?
- 무언가로 인해 극복할 수 없는 백로그가 생길 수 있을까요?
- 이 구성 요소가 바이모달 동작을 경험할 수 있는 곳은 어디입니까?
- 초과할 수 있는 한도 또는 서비스 할당량 (스토리지 용량 포함) 은 무엇입니까?
- 구성 요소는 부하 상태에서 어떻게 확장됩니까?

### 잘못된 구성 및 버그

- 잘못된 구성과 버그가 프로덕션에 배포되는 것을 방지하려면 어떻게 해야 할까요?
- 잘못된 배포를 자동으로 롤백하거나 업데이트 또는 변경 사항이 배포된 장애 컨테이너로부터 트래픽을 이동할 수 있습니까?
- 운영자 오류를 방지하기 위해 어떤 가드레일을 갖추고 있습니까?
- 어떤 항목 (예: 자격 증명 또는 인증서) 이 만료될 수 있습니까?

### 공유 요금

- 장애 격리 경계는 어떻게 되나요?
- 배포 유닛에 대한 변경 사항이 적어도 의도한 것만큼 작습니까? [장애 격리 경계](#) 하지만 원박스 환경 (장애 격리 경계 내의 단일 인스턴스) 과 같이 규모가 더 작은 것이 가장 이상적일까요?
- 이 구성 요소가 사용자 스토리 또는 다른 워크로드 간에 공유되나요?
- 이 구성 요소와 밀접하게 연결된 다른 구성 요소는 무엇입니까?
- 이 구성 요소나 해당 종속 항목에 부분적 또는 회색 오류가 발생하면 어떻게 됩니까?

이러한 질문을 한 후 SEEMS를 사용하여 워크로드 및 각 구성 요소와 관련된 다른 질문을 작성할 수도 있습니다. EEMS는 장애 모드에 대해 생각할 수 있는 구조화된 방법으로, 그리고 복원력 분석을 수행할 때 영감을 얻을 수 있는 방법으로 사용하는 것이 가장 좋습니다. 엄격한 분류 체계는 아닙니다. 특정 장애 모드가 어떤 범주에 속하는지 고민하느라 시간을 허비하지 마세요. 문제는 중요하지 않습니다. 무

엇입니다. 중요한 것은 실패를 생각하고 기록해 두었다는 것입니다. 틀린 답은 없습니다. 창의력을 발휘하고 틀에서 벗어나 사고하는 것이 좋습니다. 또한 고장 모드가 이미 완화되었다고 가정하지 말고 생각할 수 있는 모든 잠재적 고장 모드를 포함하세요.

첫 번째 연습에서 모든 잠재적 고장 모드를 예상할 가능성은 거의 없습니다. 프레임워크를 여러 번 반복하면 더 완전한 모델을 생성할 수 있으므로 첫 번째 단계에서 모든 문제를 해결하려고 하지 않아도 됩니다. 분석을 정기적으로, 매주 또는 격주로 실행할 수 있습니다. 각 세션에서 특정 장애 모드 또는 구성 요소에 초점을 맞추십시오. 이를 통해 워크로드의 복원력을 꾸준히 점진적으로 개선할 수 있습니다. 사용자 스토리의 잠재적 장애 모드 목록을 수집한 후 이에 대한 조치를 결정할 수 있습니다.

## 잠재적 장애 완화

이제 사용자 스토리의 구성 요소에 잠재적 장애가 발생했으니 완화 조치에 집중할 수 있습니다. 먼저, 발견한 각 실패의 잠재적 영향 및 가능성과 관련된 잠재적 절충점을 검토하십시오. 그런 다음 필요한 관찰 가능성 수준을 결정하고 완화 전략을 선택하십시오. 절충점에는 적절한 수준의 관찰 가능성과 완화 전략을 수립하기 위한 노력이 포함되어야 합니다. 마지막으로, 정기적인 복원력 분석 검토를 수행할 적절한 케이던스를 결정하십시오.

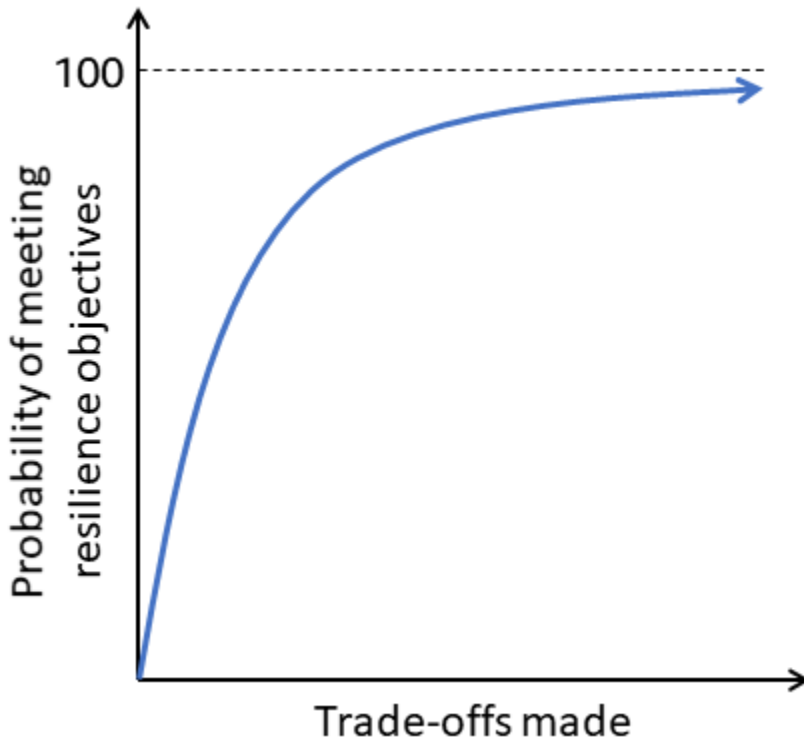
### Sections

- [트레이드오프와 위험에 대한 이해](#)
- [장애 모드 관찰 가능성](#)
- [일반적인 완화 전략](#)
- [지속적 개선](#)

## 장단점 및 위험에 대한 이해

레질리언스 아키텍처는 잘 테스트되고 간단하며 신뢰할 수 있는 몇 가지 메커니즘을 사용하여 장애에 대응해야 합니다. 최고 수준의 복원력을 달성하려면 워크로드가 최대한 많은 장애 모드를 자동으로 감지하고 복구해야 합니다. 이를 위해서는 복원력 분석을 수행하기 위한 광범위한 투자가 필요합니다. 즉, 복원력 수준을 높이려면 절충해야 합니다. 하지만 트레이드 오프를 계속하다 보면 회복력 목표에 비해 수익이 감소하는 지점에 도달하게 됩니다. 가장 일반적인 장단점은 다음과 같습니다.

- 비용 — 중복된 구성 요소, 향상된 관찰 가능성, 추가 도구 또는 리소스 활용률 증가로 인해 비용이 증가합니다.
- 시스템 복잡성 — 완화 솔루션을 포함하여 장애 모드를 감지하고 이에 대응하고 관리형 서비스를 사용하지 않을 경우 시스템 복잡성이 증가합니다.
- 엔지니어링 노력 — 장애 모드를 감지하고 이에 대응하기 위한 솔루션을 구축하려면 개발자 시간이 추가로 필요합니다.
- 운영 오버헤드 — 더 많은 장애 모드를 처리하는 시스템을 모니터링하고 운영하면 운영 오버헤드가 증가할 수 있습니다. 특히 관리형 서비스를 사용하여 특정 장애 모드를 완화할 수 없는 경우에는 더욱 그렇습니다.
- 지연 시간 및 일관성 — [가용성에 유리한 분산 시스템을 구축하려면 PACELC 정리에 설명된 대로 일관성과 지연 시간을 절충해야 합니다.](#)



사용자 사례에서 식별된 장애 모드의 완화 방법을 고려하면서 절충해야 할 사항도 생각해 보세요. 보안과 마찬가지로 복원력도 최적화 문제입니다. 식별된 장애로 인한 위험을 방지, 완화, 이전 또는 수용할지 여부를 결정해야 합니다. 피할 수 있는 장애 모드도 있고, 승낙하는 모드도 있고, 이전할 수 있는 장애 모드도 몇 개 있을 수 있습니다. 파악한 여러 장애 모드를 완화하는 방법을 선택할 수도 있습니다. 어떤 접근 방식을 취해야 할지 결정하려면 다음 두 가지 질문을 통해 평가를 수행하십시오. 장애가 발생할 가능성은 얼마나 됩니까? 문제가 발생할 경우 워크로드에 미치는 영향은 무엇입니까?

가능성은 이벤트가 발생할 가능성이 얼마나 높은지를 나타냅니다. 예를 들어 사용자 스토리에 단일 Amazon Elastic Compute Cloud (Amazon EC2) 인스턴스에서 작동하는 구성 요소가 있는 경우 패치 절차 또는 운영 체제 오류로 인해 시스템 운영 중 특정 시점에서 구성 요소가 중단될 수 있습니다. 또는 기본 인스턴스와 보조 인스턴스 간에 데이터를 동기화하는 Amazon RDS (Amazon RDS) 에서 관리하는 데이터베이스를 완전히 사용할 수 없게 될 가능성은 낮습니다.

영향은 이벤트로 인해 발생할 수 있는 피해의 추정치입니다. 재무 및 평판 측면에서 모두 평가해야 하며, 영향을 미치는 사용자 스토리의 가치와 관련이 있습니다. 예를 들어, 데이터베이스가 과부하되면 전자 상거래 시스템의 신규 주문 접수 능력에 중대한 영향을 미칠 수 있습니다. 하지만 로드 밸런서 뒤에 있는 20개의 인스턴스 중 단일 인스턴스가 손실되더라도 영향은 거의 없을 것입니다.

이러한 질문에 대한 답을 위험을 줄이기 위해 절충해야 하는 비용과 비교해 볼 수 있습니다. 위험 임계값과 복원력 목표를 고려하여 이 정보를 고려하면 적극적으로 완화하려는 장애 모드를 결정하는 데 도움이 됩니다.

## 장애 모드 관찰 가능성

장애 모드를 완화하려면 먼저 해당 장애가 현재 워크로드에 영향을 미치고 있거나 곧 영향을 미칠 것임을 감지해야 합니다. 완화 조치는 조치를 취해야 한다는 신호가 있는 경우에만 효과적입니다. 즉, 완화 조치를 마련할 때는 최소한 장애로 인한 영향을 감지하는 데 필요한 가시성을 확보했거나 구축하고 있는지 확인하는 것이 포함됩니다.

장애 모드의 관찰 가능한 증상을 두 가지 차원에서 고려해야 합니다.

- 시스템이 곧 영향을 받을 수 있는 상태에 가까워지고 있음을 알려주는 선행 지표는 무엇입니까?
- 장애 모드가 발생한 후 최대한 빨리 장애 모드의 영향을 보여줄 수 있는 지연 지표는 무엇입니까?

예를 들어, 데이터베이스 요소에 과도한 로드 장애가 발생한 경우 연결 수가 선행 지표로 간주될 수 있습니다. 연결 수가 꾸준히 증가하는 것은 데이터베이스가 곧 연결 제한을 초과할 수 있다는 선행 지표로 볼 수 있으므로 가장 최근에 사용하지 않은 연결을 종료하는 등의 조치를 취하여 연결 수를 줄일 수 있습니다. 지연 표시기는 데이터베이스 연결 제한이 초과되고 데이터베이스 연결 오류가 증가하는 시기를 나타냅니다. 애플리케이션 및 인프라 지표를 수집하는 것 외에도 장애가 고객 경험에 영향을 미치는 시기를 감지하기 위해 [핵심 성과 지표 \(KPI\)](#) 를 수집하는 것도 고려해 보십시오.

가능하면 옹저버빌리티 전략에 두 가지 유형의 지표를 모두 포함하는 것이 좋습니다. 경우에 따라 선행 지표를 만들지 못할 수도 있지만, 완화하려는 각 장애에 대해 항상 지연 지표를 마련하도록 계획해야 합니다. 적절한 완화 방법을 선택하려면 장애를 감지한 선행 지표가 있는지 지연 지표가 있는지도 고려해야 합니다. 예를 들어 웹 사이트 트래픽이 갑자기 급증했다고 가정해 보겠습니다. 지연 표시기만 표시될 가능성이 높습니다. 이 경우 자동 크기 조정만으로는 새 리소스를 배포하는 데 시간이 걸리기 때문에 최선의 완화 방법이 아닐 수 있지만, 전송률 조절은 과부하를 거의 즉시 방지하고 애플리케이션이 부하를 확장하거나 줄일 시간을 줄 수 있기 때문입니다. 반대로 트래픽이 점진적으로 증가하는 경우에는 선행 지표가 표시됩니다. 이 경우에는 시스템을 자동으로 확장하여 대응할 시간이 있기 때문에 전송률 조절이 적절하지 않을 수 있습니다.

## 일반적인 완화 전략

먼저 예방적 완화 방법을 사용하여 장애 모드가 사용자 스토리에 영향을 미치지 않도록 하는 방안을 생각해 보세요. 그런 다음 수정 방안에 대해 생각해 봐야 합니다. 시정 완화 조치는 시스템이 스스로 치유되거나 변화하는 조건에 적응하는 데 도움이 됩니다. 다음은 복원력 속성에 맞는 각 장애 범주의 일반적인 완화 조치 목록입니다.

실패 카테고리

원하는 복원력 속성

완화 조치



## 단일 장애 지점 (SPOF)

## 이중화 및 내결함성

- 예를 들어 Elastic Load Balancing (ELB) 기반의 여러 EC2 인스턴스를 사용하여 이중화를 구현합니다.
- AWS글로벌 서비스 컨트롤 플레인에 대한 종속성을 제거하고 글로벌 서비스 데이터 플레인에 대한 종속성만 취하십시오.
- 리소스를 사용할 수 없는 경우 적절한 성능 저하를 사용하여 시스템이 단일 장애 지점까지 정적으로 안정되도록 하세요.

## 과도한 부하

## 충분한 용량

- 주요 완화 전략으로는 속도 제한, 부하 차단 및 작업 우선 순위 지정, 지속적인 작업, 지터가 발생하거나 재시도하지 않는 지수 백오프 및 재시도, 소규모 서비스 제어, 대기열 깊이 관리, 자동 조정, 콜드 캐시 방지, 회로 차단기 방지 등이 있습니다.
- 또한 용량 계획을 고려하고 향후 AWS 리소스와 관련된 용량 및 규모 조정 한도 (들다) 에 도달할 수 있는 시스템 내 한도에 대해서도 생각해야 합니다.

과도한 지연 시간

적시 출력

- 적절하게 구성된 타임아웃 또는 적응형 타임아웃을 구현하세요 (현재 및 예측된 지연 시간 조건에 따라 타임아웃 값을 변경하여 느린 요청을 포기하는 대신 느린 종속성을 진행할 수 있도록 함).
- 온프레미스 환경에서 클라우드 서비스에 연결하고 특정 경로에서 지연 시간이 발생하는 경우 다중 경로 TCP와 같은 기술을 사용하고, 느슨하게 연결된 시스템과의 비동기 상호 작용을 사용하고, 작업을 낭비하지 않고, 캐싱을 사용하여 지터, 헤징을 통해 기하급수적 백오프 및 재시도를 구현하십시오.

## 잘못된 구성 및 버그

## 올바른 출력

- 소프트웨어에서 반복 가능한 기능 오류를 포착하는 주요 방법은 정적 분석, 단위 테스트, 통합 테스트, 회귀 테스트, 부하 테스트 및 복원력 테스트와 같은 메커니즘을 통한 엄격한 테스트입니다.
- 코드형 인프라 (IaC), 지속적 통합 및 지속적 전달 (CI/CD) 자동화와 같은 전략을 구현하여 잘못된 구성 위협을 완화하는 데 도움이 됩니다.
- 원박스 배포, 카나리아 배포, 장애 격리 경계에 맞게 조정된 부분 배포 또는 블루/그린 배포와 같은 배포 기법을 사용하여 구성 오류와 버그를 줄이십시오.

## 공유 운명

## 장애 격리

- 시스템에 [내결합성을](#) 구현하고 여러 컴퓨팅 또는 컨테이너 클러스터, 여러 AWS 계정, 다중 AWS Identity and Access Management (IAM) 보안 주체, 다중 가용 영역, 다중 가용 영역 등 논리적이고 물리적인 장애 격리 경계를 사용하십시오. AWS 리전
- [셀 기반 아키텍처](#) 및 [셔플 샤딩과](#) 같은 기술도 결합 격리를 개선할 수 있습니다.
- 캐스케이딩 실패를 방지하려면 [느슨한 결합 및 점진적](#) 성능 저하와 같은 패턴을 고려하십시오. 사용자 스토리의 우선 순위를 지정할 때 해당 우선 순위를 사용하여 주요 비즈니스 기능에 필수적인 사용자 스토리와 적절하게 저하될 수 있는 사용자 스토리를 구분할 수도 있습니다. 예를 들어 전자 상거래 사이트에서는 웹 사이트의 프로모션 위젯이 손상되어도 새 주문을 처리하는 데 영향을 미치지 않는 것이 좋습니다.

이러한 완화 조치 중 일부는 구현에 최소한의 노력만 필요하지만, 예측 가능한 장애 격리를 위한 셀 기반 아키텍처 채택 및 운명 공유 장애 최소화를 위한 셀 기반 아키텍처 채택 등의 경우에는 특정 사용자 스토리의 구성 요소뿐만 아니라 전체 워크로드를 재설계해야 할 수도 있습니다. 앞서 설명했듯이 장애 모드의 가능성과 영향을 비교해서 장애 모드를 완화하는 데 따르는 장단점을 비교하는 것이 중요합니다.

각 장애 모드 범주에 적용되는 완화 기법 외에도 사용자 스토리 또는 전체 시스템의 복구에 필요한 완화 조치를 고려해야 합니다. 예를 들어, 오류가 발생하면 워크플로가 중단되어 데이터가 의도한 대상에

기록되지 않을 수 있습니다. 이 경우 워크플로를 다시 실행하거나 데이터를 수동으로 수정하기 위한 운영 도구가 필요할 수 있습니다. 장애 발생 시 데이터 손실을 방지하기 위해 워크로드에 체크포인트 메커니즘을 구축해야 할 수도 있습니다. 또는 추가 피해를 방지하기 위해 워크플로를 일시 중지하고 새 작업 수락을 중단하는 안돈 코드를 만들어야 할 수도 있습니다. 이런 경우에는 필요한 운영 도구와 가드레일에 대해 생각해 봐야 합니다.

마지막으로, 완화 전략을 개발할 때는 사람이 실수를 할 것이라는 점을 항상 염두에 두어야 합니다. 현대의 DevOps 관행은 운영 자동화를 추구하지만 인간은 여전히 다양한 이유로 워크로드와 상호 작용해야 합니다. 사람의 행동이 잘못되면 유지 관리 중에 너무 많은 노드를 제거하여 과부하를 일으키거나 기능 플래그를 잘못 설정하는 등 SEELS 범주에서 오류가 발생할 수 있습니다. 이러한 시나리오는 예방적 가드레일을 제대로 활용하지 못하는 실정입니다. 근본 원인 분석은 “인간이 실수를 저질렀다”는 결론으로 끝나서는 안 됩니다. 대신 애초에 실수가 가능했던 이유를 설명해야 합니다. 따라서 완화 전략에서는 작업자가 워크로드 구성 요소와 상호 작용하는 방법과 안전 가드레일을 통해 작업자 실수로 인한 영향을 방지하거나 최소화하는 방법을 고려해야 합니다.

## 지속적 개선

레질리언스는 지속적인 프로세스입니다. 시스템 수명 주기 동안 시스템이 운영되는 환경은 변화할 것입니다. 시스템의 복원력을 유지하려면 프레임워크를 주기적인 운영 및 아키텍처 검토에 통합해야 합니다. 처음에는 식별하지 못했던 새로운 장애 모드가 발견될 수도 있고, 새로 도입하거나 이전에는 생각하지 못했던 완화 방법이 있을 수도 있습니다. 복원력 분석은 일회성이 아니라 반복적인 프로세스여야 합니다.

카오스 엔지니어링이나 게임 데이와 같은 프로세스를 통해 완화 전략을 경험적으로 테스트하여 예상대로 작동하는지 검증해야 합니다. 엄격한 테스트 메커니즘이 없다면 필요할 때 완화 조치가 예상대로 작동할지 확신할 수 없을 것입니다. 복원력 분석 중에 특정 완화 방법으로 장애 모드가 이미 처리되었다고 판단할 수 있지만 이러한 가정을 테스트해 보는 것도 중요합니다. 기존 완화 기능과 복원력 분석 프레임워크를 사용하여 만든 새 완화 기능을 모두 테스트해야 합니다.

또한 팀 회고를 통해 분석을 얼마나 잘 수행했는지 평가해야 합니다. 분석 중에 어떤 작업을 하고 있는지 모두가 알고 있었나요? 복원력 분석을 통해 발견한 장애 모드의 수가 팀의 기대치에 부합했나요? 발견한 모든 장애 모드의 완화 방법을 찾아낼 수 있습니까? 팀에서 이 프로세스가 유용하다고 생각했나요? 이를 통해 워크로드의 복원력이 향상될 것이라고 생각하시나요?

워크로드의 가용성에 영향을 미치는 실제 장애 이벤트가 발생하는 경우 특정 장애 모드, 오류의 일부였던 구성 요소, 사용된 완화 패턴을 기록해 두십시오. 사후 분석 도구에서 이 메타데이터를 검색할 수 있도록 하면 향후 어떤 고장 모드와 구성 요소에 중점을 두어야 할지 결정할 수 있습니다. 이 프로세스 전반에 걸쳐 AWS 계정 팀 및 솔루션 설계자를 참여시킬 수 있습니다.

## 결론 및 리소스

이 가이드에서는 지속적이고 일관된 방식으로 복원력 분석을 수행하기 위한 프레임워크를 제공합니다. 이 프레임워크를 사용하면 단일 장애 지점, 과도한 부하, 과도한 지연 시간, 잘못된 구성 및 버그, 공유 운명이 워크로드의 구성 요소에 어떤 영향을 미칠 수 있는지 식별할 수 있습니다. 이러한 장애 모드를 식별하면 복구 지향 아키텍처 구축의 일환으로 적절한 완화 전략을 결정하는 데 도움이 됩니다.

복원력 분석에 대한 추가 정보는 다음 링크를 참조하십시오.

- [레질리언스 라이프사이클 프레임워크](#) (AWS 규범적 지침)
- [레질리언스 솔루션 \(솔루션 라이브러리\)](#) AWS
- [지속적인 복원력을 향하여](#) (애드리안 혼스비, 클라우드 아키텍트, 2021년 3월 24일)

## 문서 기록

다음 표에는 이 가이드의 주요 변경 사항이 설명되어 있습니다. 향후 업데이트에 대한 알림을 받으려면 구독할 수 있습니다.[RSS 피드](#).

변경 사항	설명	날짜
<a href="#">초기 출판</a>	—	2023년 9월 5일

# AWS 규범적 지침 용어집

다음은 AWS 규범적 지침에서 제공하는 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 마이그레이션합니다.
- 리플랫폼(리프트 앤드 리세이프) - 애플리케이션을 클라우드로 이동하고 일정 수준의 최적화를 도입하여 클라우드 기능을 활용합니다. 예:에서 온프레미스 Oracle 데이터베이스를 Oracle용 Amazon Relational Database Service(Amazon RDS)로 마이그레이션합니다 AWS 클라우드.
- 재구매(드롭 앤드 슝) - 일반적으로 기존 라이선스에서 SaaS 모델로 전환하여 다른 제품으로 전환합니다. 예: 고객 관계 관리(CRM) 시스템을 Salesforce.com 마이그레이션합니다.
- 리호스팅(리프트 앤드 시프트) - 애플리케이션을 변경하지 않고 클라우드로 이동하여 클라우드 기능을 활용합니다. 예:의 EC2 인스턴스에서 온프레미스 Oracle 데이터베이스를 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: 마이그레이션 Microsoft Hyper-V 에 대한 애플리케이션 AWS입니다.
- 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](#) 참조하세요.

### 추상화된 서비스

[관리형 서비스를](#) 참조하세요.

## ACID

[원자성, 일관성, 격리, 내구성](#)을 참조하세요.

### 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. 유연성이 뛰어나지만 [능동 수동 마이그레이션](#)보다 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 작동하고 그룹의 단일 반환 값을 계산하는 SQL 함수입니다. 집계 함수의 예로는 SUM 및 MAX가 있습니다.

## AI

[인공 지능](#)을 참조하세요.

### AIOps

[인공 지능 작업을](#) 참조하세요.

### 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

맬웨어로부터 시스템을 보호하기 위해 승인된 애플리케이션만 사용할 수 있도록 허용하는 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 작업(AIOps)

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

## 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, 격리, 내구성(ACID)

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

## 속성 기반 액세스 제어(ABAC)

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

## 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 절연 AWS 리전 되고 동일한 리전의 다른 가용 영역에 저렴하고 지연 시간이 짧은 네트워크 연결을 제공하는 내 고유 위치입니다.

## AWS 클라우드 채택 프레임워크(AWS CAF)

조직이 클라우드로 성공적으로 전환 AWS 하기 위한 효율적이고 효과적인 계획을 개발하는 데 도움이 되는 지침 및 모범 사례 프레임워크입니다. AWS CAF는 지침을 비즈니스, 사람, 거버넌스, 플랫폼, 보안 및 운영이라는 6가지 중점 영역으로 구성합니다. 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 성공적인 클라우드 채택을 위해 조직을 준비하는 데 도움이 되는 인력 개발, 훈련 및 커뮤니케이션에 대한 지침을 제공합니다. 자세한 내용은 [AWS CAF 웹 사이트](#) 및 [AWS CAF 백서](#)를 참조하세요.

## AWS 워크로드 검증 프레임워크(AWS WQF)

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

## B

### 잘못된 봇

개인 또는 조직을 방해하거나 해를 입히기 위한 [봇](#)입니다.

### BCP

[비즈니스 연속성 계획](#)을 참조하세요.

## 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. [Endianness](#)도 참조하세요.

## 바이너리 분류

바이너리 결과(가능한 두 클래스 중 하나)를 예측하는 프로세스입니다. 예를 들어, ML 모델이 “이 이메일이 스팸인가요, 스팸이 아닌가요?”, ‘이 제품은 책인가요, 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

별개의 동일한 두 환경을 생성하는 배포 전략입니다. 현재 애플리케이션 버전은 한 환경(파란색)에서 실행하고 새 애플리케이션 버전은 다른 환경(녹색)에서 실행합니다. 이 전략을 사용하면 영향을 최소화하면서 빠르게 롤백할 수 있습니다.

## bot

인터넷을 통해 자동화된 작업을 실행하고 인적 활동 또는 상호 작용을 시뮬레이션하는 소프트웨어 애플리케이션입니다. 인터넷에서 정보를 인덱싱하는 웹 크롤러와 같은 일부 봇은 유용하거나 유용합니다. 잘못된 봇이라고 하는 다른 봇은 개인 또는 조직에 방해가 되거나 피해를 입히기 위한 것입니다.

## 봇넷

[맬웨어](#)에 감염되어 [있고 봇](#) 세더 또는 봇 운영자라고 하는 단일 당사자의 제어 하에 있는 봇 네트워크입니다. Botnet은 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](#)(GitHub) .

## 브레이크 글라스 액세스

예외적인 상황에서 승인된 프로세스를 통해 사용자가 일반적으로 액세스할 권한이 없는데 액세스할 수 있는 빠른 방법입니다. 자세한 내용은 Well-Architected 지침의 [브레이크 글라스 절차 구현](#) 표시기를 AWS 참조하세요.

## 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

[AWS 클라우드 채택 프레임워크](#)를 참조하세요.

## canary 배포

최종 사용자에게 버전의 느린 증분 릴리스입니다. 확신이 드는 경우 새 버전을 배포하고 현재 버전을 완전히 교체합니다.

## CCoE

[Cloud Center of Excellence](#)를 참조하세요.

## CDC

[데이터 캡처 변경을](#) 참조하세요.

## 데이터 캡처 변경(CDC)

데이터베이스 테이블과 같은 데이터 소스의 변경 내용을 추적하고 변경 사항에 대한 메타데이터를 기록하는 프로세스입니다. CDC를 사용하여 대상 시스템의 변경 사항을 감사하거나 복제하여 동기화를 유지할 수 있습니다.

## 카오스 엔지니어링

시스템 복원력을 테스트하기 위해 의도적으로 장애 또는 중단 이벤트를 도입합니다. [AWS Fault Injection Service \(AWS FIS\)](#)를 사용하여 AWS 워크로드에 스트레스를 가하고 응답을 평가하는 실험을 수행할 수 있습니다.

## CI/CD

[지속적 통합 및 지속적 전달](#)을 참조하세요.

## 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

대상에서 데이터를 AWS 서비스 수신하기 전에 로컬에서 데이터를 암호화합니다.

## Cloud Center of Excellence(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 엔터프라이즈 전략 블로그의 [CCoE 게시물](#)을 참조하세요.

## 클라우드 컴퓨팅

원격 데이터 스토리지와 IoT 디바이스 관리에 일반적으로 사용되는 클라우드 기술 클라우드 컴퓨팅은 일반적으로 [엣지 컴퓨팅](#) 기술에 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 로 마이그레이션할 때 일반적으로 거치는 4단계: AWS 클라우드

- 프로젝트 - 개념 증명 및 학습 목적으로 몇 가지 클라우드 관련 프로젝트 실행
- 파운데이션 - 클라우드 채택을 확장하기 위한 기본 투자(예: 랜딩 존 생성, CCoE 정의, 운영 모델 설정)
- 마이그레이션 - 개별 애플리케이션 마이그레이션
- Re-invention - 제품 및 서비스 최적화와 클라우드 혁신

이러한 단계는 Stephen Orban이 블로그 게시물 [The Journey Toward Cloud-First 및 Enterprise Strategy 블로그의 채택 단계에](#) 정의했습니다. AWS 클라우드 AWS 마이그레이션 전략과 어떤 관련이 있는지에 대한 자세한 내용은 [마이그레이션 준비 가이드](#)를 참조하세요.

## CMDB

[구성 관리 데이터베이스](#)를 참조하세요.

## 코드 리포지토리

소스 코드와 설명서, 샘플, 스크립트 등의 기타 자산이 버전 관리 프로세스를 통해 저장되고 업데이트되는 위치입니다. 일반적인 클라우드 리포지토리에는 다음이 포함됩니다. GitHub or Bitbucket Cloud. 코드의 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](#) 필드입니다. 예를 들어 온프레미스 카메라 네트워크에 CV를 추가하는 디바이스를 AWS Panorama 제공하고 Amazon SageMaker 는 CV에 대한 이미지 처리 알고리즘을 제공합니다.

## 구성 드리프트

워크로드의 경우 구성이 예상 상태에서 변경됩니다. 이로 인해 워크로드가 규정 미준수가 될 수 있으며 일반적으로 점진적이고 의도하지 않습니다.

## 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

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

### 지속적 통합 및 지속적 전달(CI/CD)

소프트웨어 릴리스 프로세스의 소스, 빌드, 테스트, 스테이징 및 프로덕션 단계를 자동화하는 프로세스입니다. CI/CD is commonly described as a pipeline. CI/CD를 사용하면 프로세스를 자동화하고, 생산성을 개선하고, 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](#) 참조하세요.

## 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 Framework의 보안 원칙 구성 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 거버넌스를 통해 분산되고 분산된 데이터 소유권을 제공하는 아키텍처 프레임워크입니다.

## 데이터 최소화

꼭 필요한 데이터만 수집하고 처리하는 원칙입니다. 에서 데이터를 최소화하면 프라이버시 위험, 비용 및 분석 탄소 발자국을 줄일 AWS 클라우드 수 있습니다.

## 데이터 경계

신뢰할 수 있는 자격 증명만 예상 네트워크에서 신뢰할 수 있는 리소스에 액세스할 수 있도록 하는 AWS 환경의 예방 가드레일 세트입니다. 자세한 내용은 [데이터 경계 구축을 참조하세요 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 시스템입니다. 데이터 웨어하우스에는 일반적으로 많은 양의 기록 데이터가 포함되며 일반적으로 쿼리 및 분석에 사용됩니다.

## 데이터베이스 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 참조하세요.

## 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 여러 제어를 AWS 추가하여 리소스를 보호할 수 있습니다. 예를 들어 a defense-in-depth 접근 방식은 다중 인증, 네트워크 세분화 및 암호화를 결합할 수 있습니다.

## 위임된 관리자

에서 AWS Organizations 호환되는 서비스는 AWS 멤버 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 Organizations 와 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 참조하세요.

## 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 스트림 매핑(DVSM)

소프트웨어 개발 라이프사이클에서 속도와 품질에 부정적인 영향을 미치는 제약 조건을 식별하고 우선 순위를 지정하는 데 사용되는 프로세스입니다. DVSM는 원래 린 제조 관행을 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.

## 차원 테이블

[스타 스키마](#)에서는 팩트 테이블의 정량적 데이터에 대한 데이터 속성을 포함하는 더 작은 테이블입니다. 차원 테이블 속성은 일반적으로 텍스트 필드 또는 텍스트처럼 동작하는 개별 숫자입니다. 이러한 속성은 쿼리 제약, 필터링 및 결과 세트 레이블 지정에 일반적으로 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)

[재해](#)로 인한 가동 중지 시간과 데이터 손실을 최소화하기 위해 사용하는 전략 및 프로세스입니다. 자세한 내용은 AWS Well-Architected Framework의 [Disaster Recovery of Workloads on AWS: Recovery in the Cloud](#)를 참조하세요.

## DML

[데이터베이스 조작 언어](#)를 참조하세요.

## 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 패턴으로 도메인 기반 설계를 사용하는 방법에 대한 자세한 내용은 [컨테이너 및 Amazon Word Gateway를 사용하여 레거시 Microsoft ASP.NET\(ASMX\) 웹 서비스 중분 현대화API](#) 참조하세요.

## DR

[재해 복구](#)를 참조하세요.

## 드리프트 감지

기존 구성과의 편차 추적. 예를 들어 AWS CloudFormation 를 사용하여 [시스템 리소스의 드리프트를 감지](#)하거나 사용하여 AWS Control Tower 거버넌스 요구 사항 준수에 영향을 미칠 수 있는 [랜딩 존의 변경 사항을 감지](#)할 수 있습니다.

## DVSM

[개발 값 스트림 매핑](#)을 참조하세요.

## E

### EDA

[탐색적 데이터 분석](#)을 참조하세요.

### EDI

[전자 데이터 교환](#)을 참조하세요.

## 엣지 컴퓨팅

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

## 전자 데이터 교환(EDI)

조직 간의 비즈니스 문서 자동 교환입니다. 자세한 내용은 [Electronic Data Interchange란 무엇입니까?](#)를 참조하세요.

## 암호화

사람이 읽을 수 있는 일반 텍스트 데이터를 암호 텍스트로 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 엔드포인트](#)를 참조하세요.

## 엔드포인트 서비스

Virtual Private Cloud(VPC)에서 호스팅하여 다른 사용자와 공유할 수 있는 서비스입니다. 를 사용하여 엔드포인트 서비스를 생성하고 다른 AWS 계정 또는 AWS Identity and Access Management (IAM) 보안 주체에 권한을 AWS PrivateLink 부여할 수 있습니다. 이러한 계정 또는 보안 주체는 인터페이스 VPC 엔드포인트를 생성하여 엔드포인트 서비스에 비공개로 연결할 수 있습니다. 자세한 내용은 Amazon Virtual Private Cloud(Amazon VPC) 설명서의 [엔드포인트 서비스 생성](#)을 참조하세요.

## 엔터프라이즈 리소스 계획(ERP)

엔터프라이즈의 주요 비즈니스 프로세스(예: 회계, [MES](#) 및 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 참조하세요.

## 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 자격 증명 및 액세스 관리, 탐지 제어, 인프라 보안, 데이터 보호 및 인시던트 대응이 포함됩니다. AWS 마이그레이션 전략의 에픽에 대한 자세한 내용은 [프로그램 구현 가이드](#)를 참조하십시오.

## ERP

[엔터프라이즈 리소스 계획을](#) 참조하세요.

## 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

[별표 스키마](#)의 중앙 테이블입니다. 비즈니스 운영에 대한 정량적 데이터를 저장합니다. 일반적으로 팩트 테이블에는 측정값이 포함된 열과 차원 테이블에 대한 외래 키가 포함된 열의 두 가지 유형이 있습니다.

### 빠른 실패

자주 증분 테스트를 사용하여 개발 수명 주기를 줄이는 철학입니다. 애자일 접근 방식의 중요한 부분입니다.

### 장애 격리 경계

에서 장애의 영향을 제한하고 워크로드의 복원력을 개선하는 데 도움이 되는 가용 영역, AWS 리전 제어 영역 또는 데이터 영역과 같은 AWS 클라우드경계입니다. 자세한 내용은 [AWS 장애 격리 경계를 참조하세요](#).

### 기능 브랜치

[브랜치를 참조하세요](#).

### 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 Shapley Additive Explanations(SHAP) 및 통합 그라데이션과 같은 다양한 기술을 통해 계산할 수 있는 수치 점수로 표현됩니다. 자세한 내용은 [AWS를 사용한 기계 학습 모델 해석 가능성을 참조하세요](#).

### 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 활용

할 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.

### 몇 번의 샷 프롬프트

유사한 작업을 수행하도록 요청하기 전에 작업과 원하는 출력을 보여주는 몇 가지 예를 [LLM](#)에 제공합니다. 이 기법은 컨텍스트 내 학습을 적용하여 모델이 프롬프트에 포함된 예제(샷)에서 학습합니다. 소수의 샷 프롬프트는 특정 형식 지정, 추론 또는 도메인 지식이 필요한 작업에 효과적일 수 있습니다. [제로샷 프롬프트도 참조하세요.](#)

### FGAC

[세분화된 액세스 제어를 참조하세요.](#)

#### 세분화된 액세스 제어(FGAC)

여러 조건을 사용하여 액세스 요청을 허용하거나 거부합니다.

#### 플래시컷 마이그레이션

단계적 접근 방식을 사용하는 대신 [변경 데이터 캡처](#)를 통해 지속적인 데이터 복제를 사용하여 가능한 최단 시간 내에 데이터를 마이그레이션하는 데이터베이스 마이그레이션 방법입니다. 목표는 가동 중지 시간을 최소화하는 것입니다.

### FM

[파운데이션 모델을 참조하세요.](#)

#### 파운데이션 모델(FM)

일반화된 데이터와 레이블이 지정되지 않은 데이터의 대규모 데이터 세트에 대해 훈련된 대규모 딥러닝 신경망입니다. FMs는 언어 이해, 텍스트 및 이미지 생성, 자연어 대화와 같은 다양한 일반 작업을 수행할 수 있습니다. 자세한 내용은 [파운데이션 모델이란 무엇입니까?](#)를 참조하세요.

## G

### 생성형 AI

대량의 데이터에 대해 훈련되었으며 간단한 텍스트 프롬프트를 사용하여 이미지, 비디오, 텍스트 및 오디오와 같은 새로운 콘텐츠 및 아티팩트를 생성할 수 있는 [AI](#) 모델의 하위 집합입니다. 자세한 내용은 [생성형 AI란 무엇입니까?](#)를 참조하세요.

#### 지리적 차단

[지리적 제한을 참조하세요.](#)

## 지리적 제한(지리적 차단)

Amazon CloudFront에서는 특정 국가의 사용자가 콘텐츠 배포에 액세스하지 못하도록 하는 옵션입니다. 허용 목록 또는 차단 목록을 사용하여 승인된 국가와 차단된 국가를 지정할 수 있습니다. 자세한 내용은 CloudFront 설명서 [의 콘텐츠의 지리적 배포 제한을 참조하세요](#).

## Gitflow 워크플로

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

## 골든 이미지

해당 시스템 또는 소프트웨어의 새 인스턴스를 배포하기 위한 템플릿으로 사용되는 시스템 또는 소프트웨어의 스냅샷입니다. 예를 들어 제조에서 황금색 이미지를 사용하여 여러 디바이스에 소프트웨어를 프로비저닝할 수 있으며 디바이스 제조 작업의 속도, 확장성 및 생산성을 개선할 수 있습니다.

## 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(OUs) 전반의 리소스, 정책 및 규정 준수를 관리하는 데 도움이 되는 상위 수준 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, 이는 AWS Config, Amazon GuardDuty AWS Security Hub, , Amazon AWS Trusted Advisor Amazon Inspector 및 사용자 지정 AWS Lambda 검사를 사용하여 구현됩니다.

# H

## HA

[고가용성을 참조하세요](#).

## 이기종 데이터베이스 마이그레이션

다른 데이터베이스 엔진을 사용하는 대상 데이터베이스로 소스 데이터베이스 마이그레이션(예: Oracle에서 Amazon Aurora로) 이기종 마이그레이션은 일반적으로 리아키텍트 작업의 일부이며 스



키마를 변환하는 것은 복잡한 작업일 수 있습니다. AWS 는 스키마 변환에 도움이 되는 [AWS SCT를](#) [제공](#)합니다.

## 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.

## 홀드아웃 데이터

[기계 학습](#) 모델을 훈련하는 데 사용되는 데이터 세트에서 보류된 레이블이 지정된 기록 데이터의 일부입니다. 홀드아웃 데이터를 사용하여 모델 예측을 홀드아웃 데이터와 비교하여 모델 성능을 평가할 수 있습니다.

## 동종 데이터베이스 마이그레이션

소스 데이터베이스를 동일한 데이터베이스 엔진을 공유하는 대상 데이터베이스로 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. 긴급성으로 인해 핫픽스는 일반적으로 typical DevOps 릴리스 워크플로 외부에서 이루어집니다.

## 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

## IaC

[인프라를 코드로](#) 참조하세요.

## 자격 증명 기반 정책

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

## 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 인터넷](#)을 참조하세요.

## 변경할 수 없는 인프라

기존 인프라를 업데이트, 패치 또는 수정하는 대신 프로덕션 워크로드를 위해 새 인프라를 배포하는 모델입니다. 변경 가능한 인프라는 본질적으로 [변경 가능한 인프라](#)보다 더 일관되고 안정적이며 예측 가능합니다. 자세한 내용은 AWS Well-Architected Framework의 [변경할 수 없는 인프라를 사용한 배포](#) 모범 사례를 참조하세요.

## 인바운드(수신) VPC

AWS 다중 계정 아키텍처에서 애플리케이션 외부에서 네트워크 연결을 수락, 검사 및 라우팅하는 VPC입니다. [AWS 보안 참조 아키텍처](#)는 애플리케이션과 더 넓은 인터넷 간의 양방향 인터페이스를 보호하기 위해 인바운드, 아웃바운드 및 검사 VPCs로 네트워크 계정을 설정하는 것이 좋습니다.

## 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 것이 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

연결성, 실시간 데이터, 자동화, 분석 및 AI/ML의 발전을 통한 제조 프로세스의 현대화를 지칭하기 위해 2016년에 [Klaus Schwab](#)에서 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\(IIoT\) 디지털 혁신 전략 구축을](#) 참조하세요.

### 검사 VPC

AWS 다중 계정 아키텍처에서는 VPC(동일하거나 다른 AWS 리전), 인터넷 및 온프레미스 네트워크 간의 네트워크 트래픽 검사를 관리하는 중앙 집중식 VPCs입니다. [AWS Security Reference Architecture](#)는 애플리케이션과 더 넓은 인터넷 간의 양방향 인터페이스를 보호하기 위해 인바운드, 아웃바운드 및 검사 VPCs로 네트워크 계정을 설정하는 것이 좋습니다.

### 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 [AWS를 사용한 기계 학습 모델 해석 가능성을](#) 참조하세요.

### IoT

[사물 인터넷을](#) 참조하세요.

### IT 정보 라이브러리(ITIL)

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

### IT 서비스 관리(ITSM)

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

## ITIL

[IT 정보 라이브러리](#)를 참조하세요.

## ITSM

[IT 서비스 관리](#)를 참조하세요.

## L

### 레이블 기반 액세스 제어(LBAC)

사용자와 데이터 자체에 각각 보안 레이블 값이 명시적으로 할당되는 필수 액세스 제어(MAC) 구현입니다. 사용자 보안 레이블과 데이터 보안 레이블 간의 교차 부분에 따라 사용자가 볼 수 있는 행과 열이 결정됩니다.

### 랜딩 존

랜딩 존은 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.

### 대형 언어 모델(LLM)

방대한 양의 데이터에 대해 사전 훈련된 딥 러닝 [AI](#) 모델입니다. LLM는 질문 답변, 문서 요약, 텍스트를 다른 언어로 번역, 문장 완성과 같은 여러 작업을 수행할 수 있습니다. 자세한 내용은 [LLMs란 무엇입니까?](#)를 참조하세요.

### 대규모 마이그레이션

300대 이상의 서버 마이그레이션입니다.

### LBAC

[레이블 기반 액세스 제어](#)를 참조하세요.

### 최소 권한

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

### 리프트 앤드 시프트

[7 Rs](#)를 참조하세요.

## 리틀 엔디안 시스템

가장 덜 중요한 바이트를 먼저 저장하는 시스템입니다. [Endianness](#)도 참조하세요.

## LLM

[큰 언어 모델을](#) 참조하세요.

## 하위 환경

[환경을](#) 참조하세요.

## 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

[브랜치를](#) 참조하세요.

### 맬웨어

컴퓨터 보안 또는 프라이버시를 손상시키도록 설계된 소프트웨어입니다. 맬웨어는 컴퓨터 시스템을 중단하거나 민감한 정보를 유출하거나 무단 액세스를 가져올 수 있습니다. 맬웨어의 예로는 바이러스, 웜, 랜섬웨어, 트로이 목마, 스파이웨어, 키로거 등이 있습니다.

### 관리형 서비스

AWS 서비스는 인프라 계층, 운영 체제 및 플랫폼을 AWS 운영하고 엔드포인트에 액세스하여 데이터를 저장하고 검색합니다. Amazon Simple Storage Service(Amazon S3) 및 Amazon DynamoDB는 관리형 서비스의 예입니다. 이를 추상화된 서비스라고도 합니다.

### 제조 실행 시스템(MES)

원재료를 생산 현장의 완제품으로 변환하는 생산 프로세스를 추적, 모니터링, 문서화 및 제어하기 위한 소프트웨어 시스템입니다.

## MAP

[마이그레이션 가속화 프로그램을](#) 참조하세요.

## 메커니즘

도구를 생성하고 도구 채택을 유도한 다음 결과를 검사하여 조정하는 전체 프로세스입니다. 메커니즘은 작동하면서 자체를 강화하고 개선하는 주기입니다. 자세한 내용은 AWS Well-Architected Framework의 [메커니즘 구축](#)을 참조하세요.

## 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 시스템](#)을 참조하세요.

## 메시지 대기열 원격 측정 전송(MQTT)

리소스가 제한된 IoT 디바이스에 대한 [게시/구독](#) 패턴을 기반으로 하는 경량 machine-to-machine(M2M) 통신 프로토콜입니다.

## 마이크로서비스

잘 정의된 APIs를 통해 통신하고 일반적으로 소규모 독립 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 [AWS 서버리스 서비스를 사용하여 마이크로서비스 통합을 참조하세요](#).

## 마이크로서비스 아키텍처

각 애플리케이션 프로세스를 마이크로서비스로 실행하는 독립 구성 요소를 사용하여 애플리케이션을 구축하는 접근 방식입니다. 이러한 마이크로서비스는 경량 APIs를 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 [에서 마이크로서비스 구현을 참조하세요 AWS](#).

## 마이그레이션 가속화 프로그램(MAP)

조직이 클라우드로 전환하기 위한 강력한 운영 기반을 구축하고 초기 마이그레이션 비용을 상쇄하는 데 도움이 되는 컨설팅 지원, 훈련 및 서비스를 제공하는 AWS 프로그램입니다. MAP에는 체계적인 방식으로 레거시 마이그레이션을 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 스프린트에서 작업하는 운영, 비즈니스 분석가 및 소유자, 마이그레이션 엔지니어, 개발자 및 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 및 AWS 계정이 있습니다.

### 마이그레이션 패턴

사용되는 마이그레이션 전략, 마이그레이션 대상, 마이그레이션 애플리케이션 또는 서비스를 자세히 설명하는 반복 가능한 마이그레이션 작업입니다. 예: AWS Application Migration Service를 사용하여 Amazon EC2로 마이그레이션을 다시 호스팅합니다.

### 마이그레이션 포트폴리오 평가(MPA)

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

### 마이그레이션 준비 평가(MRA)

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

## 마이그레이션 전략

워크로드를 로 마이그레이션하는 데 사용되는 접근 방식입니다 AWS 클라우드. 자세한 내용은 이 용어집의 [7 Rs](#) 항목을 참조하고 [대규모 마이그레이션을 가속화하기 위해 조직 동원을 참조하세요.](#)

## ML

[기계 학습](#)을 참조하세요.

## 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 애플리케이션 현대화 전략을 참조하세요 AWS 클라우드.](#)

## 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 참조하세요 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 참조하세요.

## MQTT

[메시지 대기열 원격 측정 전송](#)을 참조하세요.

## 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 워크로드에 대한 기존 인프라를 업데이트하고 수정하는 모델입니다. 일관성, 신뢰성 및 예측 가능성을 높이기 위해 AWS Well-Architected Framework는 [변경할 수 없는 인프라](#)를 모범 사례로 사용할 것을 권장합니다.

## O

### OAC

[오리진 액세스 제어를](#) 참조하세요.

### OAI

[오리진 액세스 자격 증명을](#) 참조하세요.

### OCM

[조직 변경 관리를](#) 참조하세요.

### 오프라인 마이그레이션

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

## OI

[작업 통합](#)을 참조하세요.

### OLA

[운영 수준 계약을](#) 참조하세요.

### 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

[Open Process Communications - Unified Architecture](#)를 참조하세요.

### Open Process Communications - 통합 아키텍처(OPC-UA)

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

## 운영 수준 계약(OLA)

서비스 수준 계약(SLA)을 지원하기 위해 기능 IT 그룹이 서로에게 제공할 것을 명확히 하는 계약입니다.

## 운영 준비 검토(ORR)

인시던트 및 가능한 장애의 범위를 이해, 평가, 예방 또는 줄이는 데 도움이 되는 질문 및 관련 모범 사례 체크리스트입니다. 자세한 내용은 AWS Well-Architected Framework의 [운영 준비 검토\(ORR\)](#)를 참조하세요.

## 운영 기술(OT)

물리적 환경과 협력하여 산업 운영, 장비 및 인프라를 제어하는 하드웨어 및 소프트웨어 시스템입니다. 제조업에서 OT와 정보 기술(IT) 시스템의 통합은 [Industry 4.0](#) 혁신의 주요 초점입니다.

## 운영 통합(OI)

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

## 조직 트레일

에서 생성한 추적은 조직 AWS 계정 내 모든에 대한 모든 이벤트를 AWS CloudTrail 기록합니다 AWS Organizations. 이 트레일은 조직에 속한 각 AWS 계정에 생성되고 각 계정의 활동을 추적합니다. 자세한 내용은 CloudTrail 설명서의 [조직에 대한 추적 생성](#)을 참조하세요.

## 조직 변경 관리(OCM)

사람, 문화 및 리더십 관점에서 중대하고 파괴적인 비즈니스 혁신을 관리하기 위한 프레임워크입니다. OCM는 변화 채택을 가속화하고, 전환 문제를 해결하고, 문화적 및 조직적 변화를 주도하여 조직이 새로운 시스템 및 전략을 준비하고 전환할 수 있도록 지원합니다. AWS 마이그레이션 전략에서는 클라우드 채택 프로젝트에 필요한 변경 속도 때문에이 프레임워크를 인력 가속화라고 합니다. 자세한 내용은 [OCM 가이드](#)를 참조하세요.

## 오리진 액세스 제어(OAC)

In CloudFront, Amazon Simple Storage Service(Amazon S3) 콘텐츠를 보호하기 위한 액세스를 제한하는 향상된 옵션입니다. OAC는 AWS KMS (SSE-KMS)를 사용한 모든 AWS 리전서버 측 암호화, 동적 PUT 및 S3 버킷에 대한 DELETE 요청에서 모든 S3 버킷을 지원합니다.

## 오리진 액세스 자격 증명(OAI)

In CloudFront, Amazon S3 콘텐츠를 보호하기 위해 액세스를 제한하는 옵션입니다. OAI를 사용하면 CloudFront 는 Amazon S3가 인증할 수 있는 보안 주체를 생성합니다. 인증된 보안 주체는 특정

CloudFront 배포를 통해서만 S3 버킷의 콘텐츠에 액세스할 수 있습니다. 보다 세분화되고 향상된 액세스 제어를 제공하는 [OAC](#)도 참조하세요.

## ORR

[운영 준비 상태 검토](#)를 참조하세요.

## OT

[운영 기술을](#) 참조하세요.

## 아웃바운드(송신) VPC

AWS 다중 계정 아키텍처에서 애플리케이션 내에서 시작된 네트워크 연결을 처리하는 VPC입니다. [AWS 보안 참조 아키텍처](#)는 애플리케이션과 더 넓은 인터넷 간의 양방향 인터페이스를 보호하기 위해 인바운드, 아웃바운드 및 검사 VPCs로 네트워크 계정을 설정하는 것이 좋습니다.

## P

### 권한 경계

사용자 또는 역할이 가질 수 있는 최대 권한을 설정하기 위해 IAM 보안 주체에 연결된 IAM 관리 정책입니다. 자세한 내용은 IAM 설명서의 [권한 경계](#)를 참조하세요.

### 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 및 연락처 정보가 있습니다.

## PII

[개인 식별 정보를](#) 참조하세요.

### 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

[프로그래밍 가능한 로직 컨트롤러](#)를 참조하세요.

## PLM

[제품 수명 주기 관리](#)를 참조하세요.

## 정책

권한을 정의하거나([자격 증명 기반 정책](#) 참조), 액세스 조건을 지정하거나([리소스 기반 정책](#) 참조), 조직의 모든 계정에 대한 최대 권한을 정의할 수 있는 객체입니다 AWS Organizations ([서비스 제어 정책](#) 참조).

## 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

WHERE 절에서 false 일반적으로 위치한 true 또는를 반환하는 쿼리 조건입니다.

## 조건부 푸시다운

전송 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 엔터티입니다. 이 엔터티는 일반적으로 AWS 계정, IAM 역할 또는 사용자의 루트 사용자입니다. 자세한 내용은 IAM 설명서의 역할의 보안 주체 용어 및 개념을 참조하세요. [https://docs.aws.amazon.com/IAM/latest/UserGuide/id\\_roles.html#id\\_roles\\_terms-and-concepts](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html#id_roles_terms-and-concepts)

## 프라이버시 중심 설계

전체 개발 프로세스를 통해 프라이버시를 고려하는 시스템 엔지니어링 접근 방식입니다.

## 프라이빗 호스팅 영역

Amazon Route 53이 하나 이상의 DNS 내에서 도메인 및 하위 도메인에 대한 VPCs 쿼리에 응답하는 방법에 대한 정보를 포함하는 컨테이너입니다. 자세한 내용은 Route 53 설명서의 [프라이빗 호스팅 영역 작업](#)을 참조하십시오.

### 사전 예방적 제어

규정 미준수 리소스의 배포를 방지하도록 설계된 [보안 제어](#)입니다. 이러한 제어는 리소스가 프로비저닝되기 전에 스캔 리소스를 스캔합니다. 리소스가 컨트롤을 준수하지 않으면 프로비저닝되지 않습니다. 자세한 내용은 AWS Control Tower 설명서의 [제어 참조 가이드](#)를 참조하고의 보안 [제어 구현의 사전](#) 예방적 제어를 참조하세요. AWS

### 제품 수명 주기 관리(PLM)

설계, 개발 및 출시부터 성장 및 성숙도, 거부 및 제거에 이르기까지 전체 수명 주기 동안 제품의 데이터 및 프로세스를 관리합니다.

### 프로덕션 환경

[환경](#)을 참조하세요.

### 프로그래밍 가능한 로직 컨트롤러(PLC)

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

### 프롬프트 체인

한 [LLM](#) 프롬프트의 출력을 다음 프롬프트의 입력으로 사용하여 더 나은 응답을 생성합니다. 이 기법은 복잡한 작업을 하위 작업으로 세분화하거나 예비 응답을 반복적으로 개선하거나 확장하는 데 사용됩니다. 이는 모델 응답의 정확성과 관련성을 개선하고 보다 세분화되고 개인화된 결과를 가능하게 합니다.

### 가명화

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

### publish/subscribe (pub/sub)

마이크로서비스 간의 비동기 통신을 가능하게 하여 확장성과 응답성을 개선하는 패턴입니다. 예를 들어 마이크로서비스 기반 [MES](#)에서 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 지침과 같은 일련의 단계입니다.

### 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 행렬

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

### RAG

[증강된 생성 검색](#)을 참조하세요.

### 랜섬웨어

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

### RASCI 행렬

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

### RCAC

[행 및 열 액세스 제어](#)를 참조하세요.

### 읽기 전용 복제본

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

### 재설계

[7 Rs](#)를 참조하세요.

## 복구 시점 목표(RPO)

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

## 복구 시간 목표(RTO)

서비스 중단과 서비스 복원 사이의 허용 가능한 지연 시간입니다.

## 리팩터

[7 Rs](#)를 참조하세요.

## 리전

지리적 영역의 AWS 리소스 모음입니다. 각 AWS 리전은 내결함성, 안정성 및 복원력을 제공하기 위해 서로 격리되고 독립적입니다. 자세한 내용은 [계정에서 사용할 수 있는 항목 지정을 참조 AWS 리전 하세요](#).

## 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 Rs](#)를 참조하세요.

## release

배포 프로세스에서 변경 사항을 프로덕션 환경으로 승격시키는 행위입니다.

## 재배치

[7 Rs](#)를 참조하세요.

## 리플랫폼

[7 Rs](#)를 참조하세요.

## 재구매

[7 Rs](#)를 참조하세요.

## 복원력

중단에 저항하거나 복구할 수 있는 애플리케이션의 기능입니다. 에서 복원력을 계획할 때 [고가용성](#) 및 [재해 복구](#)는 일반적인 고려 사항입니다 AWS 클라우드. 자세한 내용은 [AWS 클라우드 복원력을 참조하세요](#).

## 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.

## 책임, 책임, 상담, 정보(RACI) 매트릭스

마이그레이션 활동 및 클라우드 운영에 참여하는 모든 당사자의 역할과 책임을 정의하는 매트릭스입니다. 매트릭스 이름은 매트릭스에 정의된 책임 유형에서 파생됩니다. 실무 담당자 (R), 의사 결정권자 (A), 업무 수행 조연자 (C), 결과 통보 대상자 (I). 지원자는 (S) 선택사항입니다. 지원을 포함하는 경우 매트릭스를 RASCI 매트릭스라고 하고, 이를 제외하는 경우 이를 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 Rs](#)를 참조하세요.

## 사용 중지

[7 Rs](#)를 참조하세요.

## 검색 증강 생성(RAG)

[LLM](#)가 응답을 생성하기 전에 훈련 데이터 소스 외부에 있는 신뢰할 수 있는 데이터 소스를 참조하는 [생성형 AI](#) 기술입니다. 예를 들어 Word RAG 모델은 조직의 지식 기반 또는 사용자 지정 데이터에 대한 의미 검색을 수행할 수 있습니다. 자세한 내용은 [RAG란 무엇입니까?](#)를 참조하세요.

## 교체

공격자가 보안 인증 정보에 액세스하는 것을 더 어렵게 만들기 위해 [보안 암호](#)를 주기적으로 업데이트하는 프로세스입니다.

## 행 및 열 액세스 제어(RCAC)

액세스 규칙을 정의한 기본적이고 유연한 SQL 표현식 사용. RCAC는 행 권한과 열 마스크로 구성됩니다.

## RPO

[복구 시점 목표](#)를 참조하세요.



## RTO

[복구 시간 목표를](#) 참조하세요.

## 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 자격 증명 공급자(IdPs)가 사용하는 개방형 표준입니다. 이 기능을 사용하면 페더레이션 Single Sign-On(SSO)을 사용할 수 있으므로 사용자는 조직의 모든 사용자를 위해 IAM에서 사용자를 생성할 필요 없이 AWS Management Console 에 로그인하거나 AWS API 작업을 호출할 수 있습니다. SAML 2.0 기반 페더레이션에 대한 자세한 내용은 [SAML 설명서의 Word 2.0 기반 페더레이션 정보를](#) 참조하세요. IAM

### SCADA

[감독 제어 및 데이터 수집](#)을 참조하세요.

### SCP

[서비스 제어 정책을](#) 참조하세요.

### secret

에는 암호 또는 사용자 자격 증명과 같이 암호화된 형식으로 저장하는 AWS Secrets Manager 기밀 또는 제한된 정보가 있습니다. 보안 암호 값과 메타데이터로 구성됩니다. 보안 암호 값은 바이너리, 단일 문자열 또는 여러 문자열일 수 있습니다. 자세한 내용은 [Secrets Manager 설명서의 Secrets Manager 보안 암호의 내용을](#) 참조하세요.

### 설계별 보안

전체 개발 프로세스를 통해 보안을 고려하는 시스템 엔지니어링 접근 방식입니다.

### 보안 제어

위험 행위자가 보안 취약성을 악용하는 능력을 방지, 탐지 또는 감소시키는 기술적 또는 관리적 가드레일입니다. 보안 제어에는 [예방](#), [탐지](#), [대응](#) 및 [사전](#) 예방이라는 네 가지 주요 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) 및 보안 이벤트 관리(SEM) 시스템을 결합하는 도구 및 서비스입니다. SIEM 시스템은 서버, 네트워크, 디바이스 및 기타 소스에서 데이터를 수집, 모니터링 및 분석하여 위협 및 보안 위반을 탐지하고 알림을 생성합니다.

### 보안 응답 자동화

보안 이벤트에 자동으로 응답하거나 이를 해결하도록 설계된 사전 정의되고 프로그래밍된 작업입니다. 이러한 자동화는 보안 모범 사례를 구현하는 데 도움이 되는 [탐지적](#) 또는 [대응적](#) AWS 보안 제어 역할을 합니다. 자동 응답 작업의 예로는 VPC 보안 그룹 수정, Amazon EC2 인스턴스 패치 적용 또는 보안 인증 정보 교체 등이 있습니다.

### 서버 측 암호화

대상에서 데이터를 AWS 서비스 수신하는에 의한 데이터 암호화.

### 서비스 제어 정책(SCP)

AWS Organizations에 속한 조직의 모든 계정에 대한 권한을 중앙 집중식으로 제어하는 정책입니다. SCPs는 가드레일을 정의하거나 관리자가 사용자 또는 역할에 위임할 수 있는 작업에 대한 제한을 설정합니다. SCPs를 허용 목록 또는 거부 목록으로 사용하여 허용되거나 금지된 서비스 또는 작업을 지정할 수 있습니다. 자세한 내용은 AWS Organizations 설명서의 [서비스 제어 정책을](#) 참조하세요.

### 서비스 엔드포인트

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

### 서비스 수준 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 측면 측정입니다.

## 서비스 수준 목표(SLO)

서비스 [수준 지표](#)로 측정된 서비스 상태를 나타내는 대상 지표입니다.

## 공동 책임 모델

클라우드 보안 및 규정 준수에 AWS 대해 공유하는 책임을 설명하는 모델. AWS 는 클라우드의 보안을 책임지고, 는 클라우드의 보안을 책임집니다. 자세한 내용은 [공동 책임 모델](#)을 참조하십시오.

## SIEM

[보안 정보 및 이벤트 관리 시스템](#)을 참조하세요.

## 단일 장애 지점(SPOF)

시스템을 중단시킬 수 있는 애플리케이션의 중요한 단일 구성 요소의 장애입니다.

## SLA

[서비스 수준 계약을 참조하세요.](#)

## SLI

[서비스 수준 표시기를 참조하세요.](#)

## SLO

[서비스 수준 목표를 참조하세요.](#)

## 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 [에서 애플리케이션 현대화에 대한 단계별 접근 방식을 참조하세요 AWS 클라우드.](#)

## SPOF

[단일 장애 지점을 참조하세요.](#)

## 스타 스키마

하나의 큰 팩트 테이블을 사용하여 트랜잭션 또는 측정된 데이터를 저장하고 하나 이상의 작은 차원 테이블을 사용하여 데이터 속성을 저장하는 데이터베이스 조직 구조입니다. 이 구조는 [데이터 웨어하우스](#) 또는 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 ASP API Gateway를 사용하여 레거시 Microsoft Word.NET\(ASMX\) 웹 서비스 증분 현대화를 참조하세요](#).

## 서브넷

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

## 감독 제어 및 데이터 수집(SCADA)

제조에서 하드웨어와 소프트웨어를 사용하여 물리적 자산과 생산 작업을 모니터링하는 시스템입니다.

## 대칭 암호화

동일한 키를 사용하여 데이터를 암호화하고 복호화하는 암호화 알고리즘입니다.

## 합성 테스트

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

## 시스템 프롬프트

Word의 동작을 지시하기 위해 [LLM](#)에 컨텍스트, 지침 또는 지침을 제공하는 기법입니다. 시스템 프롬프트는 컨텍스트를 설정하고 사용자와의 상호 작용 규칙을 설정하는 데 도움이 됩니다.

# T

## tags

AWS 리소스를 구성하기 위한 메타데이터 역할을 하는 키-값 페어입니다. 태그를 사용하면 리소스를 손쉽게 관리, 식별, 정리, 검색 및 필터링할 수 있습니다. 자세한 내용은 [AWS 리소스에 태그 지정을 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](#) 참조하세요.

## 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

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

## 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

사용자를 대신하여 AWS Organizations 및 해당 계정에서 조직에서 작업을 수행하도록 지정한 서비스에 권한 부여. 신뢰할 수 있는 서비스는 필요할 때 각 계정에 서비스 연결 역할을 생성하여 관리 작업을 수행합니다. 자세한 내용은 설명서의 [다른 AWS 서비스와 AWS Organizations 함께 사용](#)을 참조하세요 AWS Organizations .

## 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

두 개의 피자로 피드할 수 있는 small DevOps 팀입니다. 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

[환경](#)을 참조하세요.

## V

### 정리

스토리지를 회수하고 성능을 향상시키기 위해 증분 업데이트 후 정리 작업을 수행하는 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 있는 두 VPCs 간의 연결입니다. 자세한 내용은 Amazon [VPC 설명서의 Word 피어링이란 무엇입니까](#) VPC를 참조하세요.

### 취약성

시스템 보안을 손상시키는 소프트웨어 또는 하드웨어 결함입니다.

# W

## 웜 캐시

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

## 웜 데이터

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

## 창 함수

현재 레코드와 어떤 식으로든 관련된 행 그룹에 대해 계산을 수행하는 SQL 함수입니다. 창 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.

## WORM

[쓰기를 한 번 보고, 많이 읽습니다.](#)

## WQF

[AWS 워크로드 검증 프레임워크를 참조하세요.](#)

## 한 번 쓰기, 많이 읽기(WORM)

데이터를 한 번에 쓰고 데이터가 삭제되거나 수정되지 않도록 하는 스토리지 모델입니다. 권한 있는 사용자는 필요한 만큼 데이터를 읽을 수 있지만 변경할 수는 없습니다. 이 데이터 스토리지 인프라는 [변경할 수 없는](#) 것으로 간주됩니다.

## Z

### 제로데이 익스플로잇

[제로데이 취약성](#)을 활용하는 공격, 일반적으로 맬웨어입니다.

### 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.

### 제로샷 프롬프트

[LLM](#)에 작업 수행에 대한 지침을 제공하지만, 작업 가이드에 도움이 될 수 있는 예제(샷)는 제공하지 않습니다. LLM는 사전 훈련된 지식을 사용하여 작업을 처리해야 합니다. 제로샷 프롬프트의 효과는 작업의 복잡성과 프롬프트의 품질에 따라 달라집니다. 또한 [몇 번의 샷 프롬프트를 참조하세요](#).

### 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.