



에서 에이전트 AI 운영 AWS

# AWS 권장 가이드



# AWS 권장 가이드: 에서 에이전트 AI 운영 AWS

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon의 상표 및 트레이드 드레스는 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
중점 영역 .....	1
대상 독자 .....	2
목표 .....	2
이 콘텐츠 시리즈 정보 .....	2
에이전트 AI의 기반 .....	3
중점 영역 .....	4
의도 및 범위 .....	5
전략 .....	5
비즈니스 가치 .....	6
구성성 및 협업 .....	7
전략 .....	7
비즈니스 가치 .....	9
다중 테넌시 및 제어 .....	10
전략 .....	10
비즈니스 가치 .....	11
신뢰할 수 있는 자율성 .....	11
전략 .....	12
비즈니스 가치 .....	13
수명 주기 관리 .....	13
전략 .....	14
비즈니스 가치 .....	14
비즈니스 조정 .....	15
전략 .....	15
소프트웨어 제공 .....	17
의도 영역 .....	17
SDLC 발전 .....	18
팀 준비 .....	19
규모 조정 준비 .....	21
팀 및 소유권 모델 .....	21
변경 관리 .....	22
상호 운용성 및 협업 .....	23
거버넌스 .....	23
운영 사고방식 .....	24

규모 조정 .....	24
결론 .....	26
리소스 .....	28
AWS 서비스 .....	28
기타 AWS 리소스 .....	29
문서 기록 .....	31
용어집 .....	32
# .....	32
A .....	33
B .....	36
C .....	37
D .....	41
E .....	44
F .....	46
G .....	48
H .....	49
I .....	51
L .....	53
M .....	54
O .....	58
P .....	60
Q .....	63
R .....	63
S .....	66
T .....	70
U .....	71
V .....	72
W .....	72
Z .....	73
.....	lxxv

# 에서 에이전트 AI 운영 AWS

Aaron Sempf, Brad Ryan, Bhargs Srivathsan, Akhil Bhaskar, Amazon Web Services

2025년 8월([문서 기록](#))

에이전트 AI는 기능이 아니라 새로운 운영 패러다임입니다. 훈련된 아키텍처, 신뢰 프레임워크 및 비즈니스 정렬 배포 모델에 투자하는 조직은 차세대 적응형 지능형 엔터프라이즈를 주도할 것입니다.

에이전트 AI는 자율 소프트웨어 에이전트와 생성형 AI의 수렴을 나타냅니다. 에이전트의 의사 결정 및 목표 지향적 행동을 대규모 언어 모델(LLMs) 이러한 에이전트는 동적 엔터프라이즈 환경에서 추론, 행동, 적응 및 협업할 수 있습니다. 이러한 잠재력을 운영하려면 기업이 모델 배포에서 에이전트 인프라로 사고방식을 전환해야 합니다.

이 가이드는 에이전트 AI를 격리된 실험에서 엔터프라이즈 규모의 가치 창출 인프라로 변환하는 조직 전략을 제공합니다. 거버넌스, 확장성 및 비즈니스 조정을 통해 워크플로에 지능형 에이전트를 포함할 수 있습니다.

## 주요 중점 영역 및 권장 사항

이 가이드는 에이전트 AI를 운영할 때 다음과 같은 기본 영역에 중점을 둡니다. 각 중점 영역에 대해 조직 및 비즈니스 권장 사항이 제공됩니다.

- [중점 영역 1: 에이전트 의도 및 범위 명확화](#) - 에이전트를 비즈니스 우선순위 및 인지 병목 현상에 맞게 조정합니다. 에이전트를 도구뿐만 아니라 디지털 팀원으로 취급합니다.
- [중점 영역 2: 구성 가능성 및 협업을 위한 설계](#) - 중재자 에이전트를 통해 모듈식 아키텍처, 의미 체계 프로토콜 및 동적 위임으로 다중 에이전트 시스템을 수용합니다.
- [중점 영역 3: 다중 테넌시 및 제어를 위한 설계자](#) - 공유 에이전트 서비스, 중앙 집중식 거버넌스 및 역할 기반 액세스를 사용하여 확장 가능한 테넌트 인식 인프라를 구축합니다.
- [중점 영역 4: 자격 증명, 가드레일 및 관찰성을 통한 신뢰 구축](#) - 추적 가능성, 런타임 제어 및 설명 가능성을 적용하여 이해관계자 신뢰를 얻습니다.
- [중점 영역 5: 수명 주기 관리](#) - 에이전트 AI 성능 및 효율성을 지원하기 위해 지속적 통합 및 지속적 배포(CI/CD) 파이프라인, 프롬프트 버전 관리, 원격 측정 및 지속적 재훈련을 설정합니다.
- [중점 영역 6: 에이전트 모델을 비즈니스 모델에 맞게 조정](#) - 사용량 기반 모델, 내부 ROI 지표 및 상용 제품을 통해 에이전트 기능을 수익화합니다.

이 가이드의 권장 사항을 사용하여 대규모 에이전트 AI를 위한 비즈니스를 준비할 수 있습니다. DevOps for Agent(AgentOps) 팀 구축, 상호 운용 가능한 시스템, 채택 규모를 조정하는 변경 관리 전략을 포함하여 조직이 에이전트 AI를 중심으로 재구성하는 방법을 간략하게 설명합니다. 의사 결정 우선 사고와 AWS Well-Architected 프레임워크와의 조정을 강조합니다.

## 대상 독자

이 가이드는 에이전트 시스템을 설계 및 확장하고, AI를 핵심 비즈니스 워크플로에 내장하고, 프로덕션 환경에서 LLMs 및 디지털 혁신 전략가를 대상으로 합니다. 이 가이드의 개념과 권장 사항을 이해하려면 최신 클라우드 네이티브 아키텍처 및 분산 시스템, 대규모 언어 모델, 파운데이션 모델 기능, AI 거버넌스, DevOps 및 플랫폼 엔지니어링의 원칙을 숙지해야 합니다.

## 목표

이 가이드의 권장 사항을 구현하면 조직은 다음과 같은 비즈니스 성과를 달성할 수 있습니다.

- 인간 병목 현상과 인지 부하를 줄이는 자율적이고 목표 지향적인 에이전트를 통해 의사 결정 및 워크플로 실행을 가속화합니다.
- 재사용 가능한 멀티테넌트 에이전트 플랫폼을 통해 여러 사업부에 지능형 기능을 확장 가능하고 비용 효율적으로 배포합니다.
- AI 시스템의 복원력, 신뢰 및 거버넌스가 향상되어 규제, 미션 크리티컬 또는 고객 대면 환경에서 자신 있게 채택할 수 있습니다.

## 이 콘텐츠 시리즈 정보

이 가이드는 에이전트 AI on에 대한 시리즈의 일부입니다 AWS. 자세한 내용과 이 시리즈의 다른 가이드를 보려면 AWS 권장 가이드 웹 사이트의 [에이전트 AI](#)를 참조하세요.

## 에이전트 AI의 전략적 기반

에이전트 시스템은 새 것이 아닙니다. 로봇 프로세스 자동화(RPA) 및 의사 결정 엔진을 비롯한 소프트웨어 에이전트는 수십 년 동안 존재해 왔습니다. 하지만 반복적이고 변형이 적은 작업을 실행하기 위해 사전 정의된 규칙과 심볼 로직을 따르도록 설계된 간단하고 결정적이었습니다. 생성형 AI가 증가함에 따라 게임이 변경되었습니다. 대규모 언어 모델(LLMs) 이제 복잡한 입력을 해석하고, 동적으로 응답을 생성하고, 지식을 빠르게 합성할 수 있습니다. 이제 부서지거나 하드 코딩된 로직 없이 기관을 확장할 수 있습니다. 이제 에이전트는 추론하고, 결정을 내리고, 도구를 호출하고, 컨텍스트에 적응하고, 워크플로 전반에서 다른 에이전트와 조정할 수 있습니다. 목표를 향해 자율적으로 운영하고, 메모리를 유지하고, 결과를 반영할 수 있습니다.

그러나 원시 기능으로는 충분하지 않습니다. 통합이 없는 인텔리전스는 영향을 미치지 않고 새로운 기능을 제공합니다. 강력한 LLMs에서 가치를 실현하려면 기업은 격리된 실험을 넘어 엔지니어링된 에코 시스템으로 전환해야 합니다. 에이전트는 모든 엔터프라이즈 시스템과 동일한 분야에서 운영되는 프로덕션급 서비스로 취급되어야 합니다. 여기에는 거버넌스, 관찰성, 보안 자격 증명 모델 및 수명 주기 관리가 포함됩니다. 또한 추측적 잠재력이 아닌 실제 비즈니스 성과로 이어져야 합니다. 이러한 시스템은 의사 결정 및 내결합성을 위해 명확한 경계로 설계되어야 합니다. 자동 복구 메커니즘, 실시간 성능 모니터링 및 확장 가능한 리소스 관리를 통합하는 것이 중요합니다. 이를 통해 엔터프라이즈 워크플로 전반에서 일관된 서비스 수준을 유지하면서 에이전트 상호 작용의 동적이고 비결정적인 특성을 처리할 수 있습니다.

기본 수준에서 기업은 인텔리전스가 운영 구조에 어떻게 포함되는지 다시 생각해야 합니다. 에이전트는 핵심 시스템과 통합하고, 엔터프라이즈 정책을 준수하며, 측정 가능한 가치를 제공하도록 설계되어야 합니다. 여러 부서, 도메인 및 사용자 컨텍스트에서 대규모로 운영해야 합니다. 에이전트 AI 운영은 궁극적으로 사용에 관한 것입니다. 이는 격리된 작업을 수행하는 AI 배포와 비즈니스 모델을 발전시키는 에이전트 배포의 차이입니다.

에이전트 AI는 조직 전체에서 인텔리전스를 확장하기 위해 시스템, 프로세스 및 인력에 접근하는 방식을 근본적으로 전환해야 하는 새로운 운영 철학을 나타냅니다. 에이전트는 인적 역량을 강화하는 전략적 자산이 됩니다. 조직은 에이전트 AI를 운영에 통합하여 비즈니스 가치를 창출하고, 인적 역량을 강화하며, 복잡한 워크플로를 최적화하는 인사이트를 얻을 수 있습니다.

## 에이전트 AI를 위한 전략적 중점 영역

초기 프로토타입에서 프로덕션 등급 및 가치 창출 시스템으로 전환하기 위해 팀은 아키텍처, 프로세스 및 제품 사고를 혼합하는 일관된 전략이 필요합니다.

많은 조직이 여전히 도구 우선 또는 모델 중심 사고방식을 사용하여 AI에 접근합니다. 생성형 AI는 실험을 증폭시켰지만 비즈니스 전략 또는 측정 가능한 결과에 대한 명확한 조정이 없는 경우가 많습니다. 정의된 전략적 역할이 없으면 에이전트는 확장 가능한 가치를 제공하는 대신 리소스를 비우는 새로운 실험이 될 위험이 있습니다. 에이전트 AI의 전략적 역할을 설정하려면 조직이 비즈니스 우선 순위부터 시작해야 합니다. 자율성이 완화를 제공할 수 있는 인지 과부하, 결정 병목 현상 또는 조각화된 워크플로 영역을 식별합니다. 도메인별 문제 설명을 사용하여 에이전트 책임을 구성합니다. 에이전트를 도구가 아니라 추론, 위임 및 적응할 수 있는 디지털 팀원으로 취급합니다.

의사 결정 과학은 데이터 과학, 분석 및 행동 모델링을 결합하여 의사 결정을 개선하는 분야입니다. 설계를 비즈니스 성과에 맞게 조정하려면 에이전트 아키텍처 프로세스 초기에 통합해야 합니다. 의사 결정 패턴을 식별하고, 장단점을 시뮬레이션하고, 가치 영향을 정량화함으로써 의사 결정 과학은 에이전트 자율성이 가장 높은 가치를 제공할 수 있는 위치를 정확히 파악하는 데 도움이 될 수 있습니다. 의사 결정 과학은 의사 결정을 가속화하고 오류를 줄이며 실시간 적응을 가능하게 할 수 있습니다. 이 데이터 기반 기반은 측정 가능한 인사이트에서 에이전트 설계를 기반으로 하며 규칙 엔진, 분석 플랫폼 및 예측 모델과 같은 기존 엔터프라이즈 기술과 더 긴밀하게 통합할 수 있습니다.

에이전트의 전략적 역할을 설정하는 데 도움이 되도록 이 섹션에서는 에이전트 AI 운영의 백bones을 형성하는 기본 중점 영역을 소개합니다. 각 에이전트를 구상하고 설계하는 방법을 담당하는 기술 리더, 아키텍트 또는 제품 소유자의 관점에서 수행할 핵심 작업에 매핑됩니다. 이러한 중점 영역은 순차적 단계가 아닙니다. 각 시스템 수명 주기 전반에 걸쳐 복원력, 확장성 및 수익화가 가능한 에이전트 에코 시스템을 구축하기 위해 재검토할 가치가 있습니다.

이 섹션에는 다음과 같은 중점 영역이 포함되어 있습니다.

- [중점 영역 1: 에이전트 의도 및 범위 명확화](#)
- [중점 영역 2: 구성 가능성 및 협업을 위한 설계](#)
- [중점 영역 3: 다중 테넌시 및 제어를 위한 설계자](#)
- [중점 영역 4: 자격 증명, 가드레일 및 관찰성을 통한 신뢰 구축](#)
- [중점 영역 5: 수명 주기 관리](#)
- [중점 영역 6: 에이전트 모델을 비즈니스 모델에 맞게 조정](#)

## 중점 영역 1: 에이전트 의도 및 범위 명확화

수행할 작업: "각 에이전트가 멋진 데모뿐만 아니라 명확한 경계로 실제 문제를 해결할 수 있도록 도와주세요."

에이전트 AI는 단지 기능 구축에만 국한되지 않습니다. 올바른 결과를 위해 올바른 방식으로 올바른 문제를 해결하는 것입니다. 이는 에이전트 AI 솔루션의 의도를 완전히 명확하게 파악하는 것부터 시작됩니다.

### 전략

조직은 모델이 수행할 수 있는 작업(예: APIs 호출, 질문 답변 또는 요약 생성)으로 시작하여 이를 중심으로 사용 사례를 개선합니다. 이로 인해 기술적으로 중요하지만 운영상 쓸모 없는 범위 크리프, 잘못된 통합 및 에이전트가 발생합니다. 대신 다음과 같은 특정 질문을 통해 에이전트의 역할을 정의하는 것으로 시작합니다.

- 에이전트가 담당하는 구체적인 결과는 무엇입니까?
- 누가 대신 행동하나요?
- 누가 혜택을 받나요?
- 에이전트의 자율성은 어디에서 시작되고 종료되나요?
- 실패하면 어떻게 되나요?

범위가 잘 지정된 에이전트에는 명확한 작업, 정의된 책임 및 측정 가능한 성공 기준이 있습니다. 에이전트를 어시스턴트 또는 챗봇으로 생각하지 마세요. 대신 제목을 지정합니다. 고객 성공 에이전트, 제품 반환 핸들러 또는 규정 준수 모니터라고 생각하십시오.

이해관계자 또는 고객을 참여시킬 때는 에이전트 AI 시스템의 확장성과 적응성을 강조합니다. 이러한 에이전트는 비즈니스와 함께 발전하여 학습과 피드백을 통해 지속적으로 개선됩니다. 저항을 줄이고 채택을 가속화하려면 작업자 공감을 염두에 두고 에이전트 도구를 설계하는 방법을 강조합니다. 이는 신뢰를 구축하는 투명성, 제어 및 선택적 재정의 메커니즘을 제공합니다. 에이전트는 사람을 대체하는 대신 인적 역량과 의사 결정을 강화하여 직원이 루프를 유지하고 고부가가치 작업에 집중할 수 있도록 지원합니다.

성공적인 구현의 핵심은 에이전트 AI를 구체적이고 영향력이 큰 비즈니스 성과에 맞추는 것입니다. 팀과 파트너가 눈에 보이는 문제를 해결하는 집중적인 파일럿 프로젝트로 시작하도록 장려합니다. 빠른 성공은 측정 가능한 투자 수익(ROI)을 창출하고, 내부 동의를 구축하며, 더 광범위한 채택을 위한 추진력을 창출합니다.

채택 및 성숙도를 안내하기 위해 조직은 진화 모델을 따라 에이전트 설계를 구성할 수 있습니다. 에이전트 자율성, 복잡성 및 비즈니스 영향이 점진적으로 증가합니다. 다음은 이 모델의 단계입니다.

- 오픈서버 에이전트는 노이즈의 인사이트를 표시합니다. 디지털 채널 전반의 브랜드 지각을 추적하는 시장 감성 에이전트를 예로 들 수 있습니다.
- 보조 에이전트는 사람의 의사 결정을 지원합니다. 영업 팀을 위해 경쟁자 데이터와 시장 조건을 합성하는 거래 자문 에이전트를 예로 들 수 있습니다.
- 자율 에이전트는 정의된 경계 내에서 독립적으로 작동합니다. 예를 들어, 수요에 따라 클라우드 인프라를 동적으로 조정하는 리소스 할당 에이전트가 있습니다.
- 오케스트레이터 에이전트는 다중 에이전트 워크플로를 조정합니다. 재고, 물류 및 예측 에이전트 간의 상호 작용을 관리하는 공급망 최적화 에이전트를 예로 들 수 있습니다.
- 혁신 에이전트는 새로운 전략적 가능성을 창출합니다. 시장 추세를 분석하고 새로운 수익 흐름을 추천하는 비즈니스 모델 혁신 에이전트를 예로 들 수 있습니다.

에이전트를 이러한 전략적 성과 및 성숙도 수준을 중심으로 프레이밍하면 집중력이 향상되고 채택이 가속화되며 이해관계자의 신뢰도가 높아집니다.

[Amazon Quick](#) AWS 서비스와 같은 이 중점 영역에서 정렬을 지원하기 위해서는 에이전트 기반 결과와 연결된 핵심 성과 지표(KPIs)를 시각화할 수 있습니다. [Amazon CloudWatch](#)를 사용하여 에이전트 동작, 성능 지표 및 시스템 상태를 거의 실시간으로 모니터링할 수 있습니다. 운영 피드백을 사용하여 에이전트 상호 작용 및 리소스 사용을 조정합니다. 초기 실험 및 개선 단계에서 에이전트 활동 및 통합 패턴을 파악할 [AWS CloudTrail](#) 수 있습니다.

## 의도 및 범위 정의의 비즈니스 가치

에이전트 AI의 채택은 조직이 디지털 트랜스포메이션과 운영 우수성에 접근하는 방식의 핵심 전환을 나타냅니다. 이는 자동화에만 국한되지 않습니다. 이는 의사 결정과 가치 실현을 가속화하는 지능형 자율성을 활성화하는 것입니다.

주요 비즈니스 동인은 다음과 같습니다.

- 경쟁 우위 - 얼리 어답터는 더 빠른 인사이트, 더 나은 서비스 및 적응형 운영을 통해 전략적 이점을 얻습니다.
- 고객 경험 개선 - 에이전트는 만족도와 충성도를 높이는 실시간 맞춤형 상시 지원을 제공합니다.
- 운영 효율성 - 에이전트 AI는 복잡하고 반복적인 결정 작업을 자동화하여 인간의 인지 부하를 크게 줄입니다. 이를 통해 직원은 가치가 높은 활동에 집중할 수 있으며 비용을 절감할 수 있습니다.

산업 전반의 실제 사용 사례는 다음과 같습니다.

- 금융 서비스 - AI 에이전트는 맞춤형 금융 조언을 제공하고 사기를 탐지할 수 있습니다.
- 의료 - 분류 및 처리 계획 에이전트는 임상 처리량을 개선할 수 있습니다.
- 소매 - 에이전트는 지능형 쇼핑 도우미 역할을 하거나 실시간으로 인벤토리를 최적화할 수 있습니다.
- 제조 - 에이전트는 예측 유지 관리를 수행하거나 공급망을 조정할 수 있습니다.

## 중점 영역 2: 구성 가능성 및 협업을 위한 설계

수행할 작업: "내와 같은 에이전트를 빌드해 보겠습니다 - 필요에 따라 구성하고 오케스트레이션할 수 있도록 모듈식이며 테스트할 수 있습니다."

많은 AI 노력은 모놀리식 모델 중심 파일럿으로 시작됩니다. 유용하지만 도메인 간에 확장하거나 복잡한 문제에 적응하기가 어렵습니다. 이러한 에이전트가 상호 작동하도록 설계된 경우 컴파운드 값을 지정합니다. 기술에서 구성성은 모듈식 구성 요소를 결합하여 변화에 적응할 수 있는 유연하고 확장 가능한 솔루션을 만드는 것입니다. 구성성이 없으면 인텔리전스가 특정 워크플로 내에서 잠깁니다. 또한 에이전트 협업은 기존 자동화 팀이 처리하지 못할 수 있는 오케스트레이션, 상태 관리 및 프로토콜 협상 복잡성을 도입합니다.

### 전략

다중 에이전트 패러다임을 수용합니다. 조직 부서와 같은 모델 에이전트: 모듈식, 전문화 및 상호 운용 가능. 명확한 인터페이스, 공유 컨텍스트 형식, [모델 컨텍스트 프로토콜\(MCP\) 또는 Agent2Agent\(A2A\)와 같은 표준 통신 프로토콜을](#) 정의합니다. [Agent2Agent](#) 스웸, 그래프 또는 계층적 조정과 같은 다중 에이전트 오케스트레이션 패턴을 채택합니다. 이러한 패턴은 에이전트가 작업 구조 및 신뢰 수준에 따라 병렬, 순차 또는 합의 기반 워크플로에서 기능을 검색하고 서로 동적으로 서비스를 요청하는 데 도움이 됩니다.

확장 가능하고 통제된 공동 작업을 촉진하려면 arbiter 에이전트를 사용합니다. 이러한 종류의 에이전트는 알려진 기능 및 대체 전략을 기반으로 작업 위임을 용이하게 하는 중립적인 기관입니다. 중앙 집중식 컨트롤러는 아니지만 arbiter 에이전트는 신뢰 및 규정 준수에 중요한 역할을 합니다. 민감한 작업이나 규제되는 작업이 자격 증명 및 정책 요구 사항을 충족하는 에이전트에게만 라우팅되도록 합니다. 정책 바운드 워크플로의 게이트키퍼 역할을 합니다. 격리를 적용하고 설명 가능한 위임을 활성화합니다. 중요한 것은 파티터 에이전트가 병목 현상이 아니라 수평적인 peer-to-peer 방식으로 작동하는 자체 조정 에이전트와 공존한다는 것입니다. 이러한 에이전트는 하위 작업을 위임하고, 컨텍스트를 공유하고, 종속성을 직접 해결합니다.

이 하이브리드 모델은 결정적 할당(파이터 에이전트를 통해)과 새로운 공동 작업을 모두 지원합니다. 구조와 유연성을 혼합합니다. 이 아키텍처 내에서 에이전트는 다음과 같은 특수 역할로 분류될 수 있습니다.

- 정책 시행자, 리소스 할당자 및 위험 평가자와 같은 결정 에이전트
- 컨텍스트 집계자, 패턴 인식기, 이상 탐지기와 같은 지식 에이전트
- 작업 실행기, 품질 컨트롤러 및 통합 관리자와 같은 실행 에이전트

효과적으로 조정하려면 다중 에이전트 시스템이 상태 관리, 장애 복구 및 충돌 해결을 위한 강력한 상호 작용 프로토콜을 지원해야 합니다. 이렇게 하면 에이전트가 독립적으로 운영되더라도 안정성과 책임을 높일 수 있습니다.

로드 기반 에이전트 인스턴스화, 컨텍스트 인식 리소스 할당, 자동화된 기능 검색 및 등록과 같은 조정 에 대한 명확한 규칙을 설정합니다. 이러한 조치는 수요 또는 복잡성에 대응하여 시스템이 동적으로 성장하는 데 도움이 됩니다.

분산 메시징 기관 내에서 ready-to-use 모듈로 에이전트를 설계합니다. 예를 들어 사일로화된 서비스가 아닌 A2A 또는 MCP와 함께 [Amazon EventBridge](#)를 사용할 수 있습니다. 버전 관리, CI/CD 파이프라인 및 에이전트 템플릿을 채택하여 시스템 안정성을 지원하는 동시에 내부 채택 및 수명 주기 진화를 가속화합니다. 코드 재사용 및 표준화를 장려하여 통합 마찰을 줄이고 탄력적인 에코시스템을 촉진합니다.

공동 작업은 물리적 승수입니다. 다중 에이전트 환경에서 규모 조정, 전문화 및 복원력을 실현합니다. 이러한 동적 협업을 지원하기 위해 조직은 에이전트 조정을 위한 경량 컨트롤 플레인을 설계해야 합니다. 이 컨트롤 플레인에는 다음이 포함됩니다.

- 각 에이전트가 수행할 수 있는 작업을 정의하고 피어 검색을 위해 버전이 지정된 메타데이터를 지원하는 기능 레지스트리
- 중재자 또는 감독자 에이전트를 사용하여 컨텍스트, 가용성 및 정책에 따라 작업을 라우팅하는 작업 중재 로직
- 실시간 의사 결정 컨텍스트 및 안전한 핸드오프를 지원하는 수명 주기 및 상태 추적

컨트롤 플레인은 다중 에이전트 시스템이 권한을 중앙 집중화하거나 작업을 늦출 필요 없이 확장 가능하고 정책이 일치하며 내결합성을 유지하도록 합니다.

그러나 다중 에이전트 환경에서는 운영 문제도 발생합니다. 에이전트 상호 작용 전반에 걸쳐 컨텍스트를 유지하고, 공유 상태를 관리하고, 작업을 조정하면 복잡성과 비용이 발생할 수 있습니다. 에이전트

간 통신 중에 토큰LLMs을 사용하는 경우 비용이 증가할 수 있습니다. 이러한 비용은 대규모 지능형 자율성의 복합적인 비즈니스 이점과 비교해야 합니다.

이러한 문제를 해결하려면 다음과 같은 주요 문제를 추상화하는 에이전트 플랫폼을 고려하세요.

- 표준화된 통신 프로토콜 및 의미 체계 형식
- 내장 오케스트레이션 로직 및 동적 라우팅
- 에이전트 간 공유 컨텍스트 및 메모리 관리
- 장애 발생 시 폴백 처리 및 정상적인 성능 저하

다중 에이전트 전략을 채택하는 팀의 경우 가장 좋은 방법은 소규모로 시작하고 규모에 맞게 설계하는 것입니다. 실제 문제를 해결하는 대상 단일 에이전트 솔루션으로 시작합니다. 그런 다음 이러한 에이전트를 공동 목표 및 시스템 전체 컨텍스트에 따라 각 에이전트가 검색, 조정 및 위임할 수 있는 협력 시스템으로 점진적으로 구성합니다.

중요한 것은 강력한 오류 처리와 정상적인 성능 저하가 기본 설계 원칙이어야 한다는 것입니다. 다중 에이전트 시스템은 에이전트를 사용할 수 없거나 에이전트가 실패할 때 부분 워크플로를 계속하거나 백업 로직을 시작할 수 있어야 합니다. 이렇게 하면 엄격한 결합 없이 신뢰성을 높일 수 있습니다.

AWS 서비스 는이 아키텍처를 대규모로 지원하는 강력한 기능을 제공합니다. [Amazon EventBridge](#) 및 [EventBridge 파이프](#)는 다중 에이전트 메시징을 위한 구조화된 이벤트 기반 백본을 제공합니다. 모듈식 동작을 관리하기 위해 [AWS AppConfig](#)를 사용하면 에이전트 인스턴스 간에 안전하고 동적인 구성을 전환할 수 있습니다. 공유 컨텍스트 및 메모리 관리를 지원하려면 [Amazon DynamoDB](#)를 사용하여 에이전트 간에 간단한 테넌트 인식 상태 지속성 및 빠른 컨텍스트 검색을 수행합니다. [Amazon Simple Storage Service\(Amazon S3\)](#)를 사용하여 구조화된 흐름포트 기록, 공유 아티팩트 또는 에이전트 생성 출력을 저장할 수 있습니다. 상태 저장 조정이 필요한 보다 복잡한 워크플로의 경우는 체크포인트 및 오류 복구 로직을 사용하여 장기 실행 프로세스를 오케스트레이션할 [AWS Step Functions](#) 수 있습니다. 이러한 서비스를 함께 사용하면 엔터프라이즈 요구 사항에 따라 확장되는 구성 가능하고 복원력이 뛰어나며 의미론적으로 연결된 다중 에이전트 시스템을 생성할 수 있습니다.

## 다중 에이전트 시스템의 비즈니스 가치

많은 조직이 단일 에이전트 솔루션으로 AI 여정을 시작하지만 확장 가능한 다중 에이전트 시스템을 통해 에이전트 AI의 잠재력을 최대한 활용할 수 있습니다. 이러한 시스템은 복잡하고 분산된 문제를 해결하고 비즈니스 요구 사항에 따라 진화하는 강력하고 유연한 AI 에코시스템을 만드는 데 중요합니다.

다중 에이전트 시스템의 핵심 비즈니스 이점은 다음과 같습니다.

- 확장성 - 작업 및 워크로드를 특수 에이전트에 분산하여 용량과 성능을 높일 수 있습니다.

- 유연성 - 종단을 최소화하면서 에이전트를 추가, 교체 또는 수정할 수 있으므로 동적 환경에서 민첩성을 높일 수 있습니다.
- 복원력 - 중복 역할과 지능형 장애 조치로 인해 개별 에이전트가 실패하더라도 시스템 안정성이 유지됩니다.
- 전문화 - 특별히 구축된 에이전트는 더 높은 효율성과 정밀도로 작업을 수행합니다.
- 비용 효율성 - 재사용 가능한 에이전트 구성 요소는 개발을 가속화하고 새로운 기능 배포 비용을 줄입니다.

다중 에이전트 시스템에는 더 많은 사전 계획이 필요하지만 장기적인 민첩성, 속도 및 혁신 용량을 제공합니다. 유연한 에이전트 협업 아키텍처에 투자하는 기업은 새로운 AI 기능을 신속하게 배포하고, 변화하는 수요에 적응하고, 점점 더 에이전트 중심의 경쟁 환경을 조성할 수 있는 위치에 있습니다.

### 중점 영역 3: 다중 테넌시 및 제어를 위한 설계자

수행할 작업: "제어, 책임 또는 가시성을 잃지 않고 여러 고객에 걸쳐 에이전트 사용량을 조정할 수 있도록 도와주세요."

초기 프로토타입은 가치를 독립적으로 증명하는 데 적합하지만 대부분의 기업은 여러 고객, 부서 또는 워크플로를 동시에 지원해야 합니다. 즉, 각 에이전트는 명확하게 정의된 정책, 데이터 및 자격 증명 경계 내에서 운영되어야 합니다. 다중 테넌시가 없으면 운영이 취약해지고 비용이 많이 들며 거버넌스는 패치워크가 됩니다.

#### 전략

서비스형 소프트웨어(SaaS) 아키텍처의 원칙을 따릅니다. 예를 들어 테넌트 격리, 정책 적용 및 리소스 제어를 위한 설계입니다. 테넌트 인식 메모리, 구성 및 ID를 사용하여 에이전트 및 오케스트레이션 플랫폼을 설계합니다. 경계를 적용하려면 태그 지정, 역할 기반 액세스 제어(RBAC), ID 및 액세스 관리 범위를 사용합니다.

테넌트 컨텍스트별로 에이전트 원격 측정을 집계하는 통합 관찰성 계층을 채택합니다. 중앙 집중식 정책 엔진 및 구성 기반 기능 토글을 구현하여 동적 동작 규칙을 적용합니다.

에이전트를 서비스로 배포합니다. 내부 팀 또는 고객이 확장 가능하고 관리되는 APIs는 이러한 패턴에 대한 강력한 기반을 AWS 제공합니다. [Amazon Cognito](#)를 사용하여 사용자 및 테넌트 자격 증명을 관리하고, [AWS Organizations 서비스 제어 정책\(SCPs\)](#)을 사용하여 교차 계정 거버넌스를 관리하고, [AWS Resource Access Manager \(AWS RAM\)](#)를 사용하여 기능을 안전하게 공유할 수 있습니다. 또한 [AWS AppConfig](#)는 테넌트 또는 환경별로 에이전트 동작을 동적으로 관리할 수 있습니다. 이러한 서비스는 공유 인프라를 지원하면서 경계와 정책을 적용하는 데 도움이 됩니다.

이러한 정적 배포에서 동적 프로비저닝으로의 전환은 에이전트 AI를 전사적 플랫폼으로 전환합니다.

## 다중 테넌트 에이전트 플랫폼의 비즈니스 가치

다중 테넌트는 단순한 아키텍처 편의가 아니라 비즈니스 액셀러레이터입니다. 지능형 에이전트가 부서 및 팀 간에 확산됨에 따라 조직은 인프라를 복제하거나 거버넌스를 세분화하지 않고 성장을 지원해야 합니다.

멀티 테넌트 시스템의 주요 비즈니스 이점은 다음과 같습니다.

- 확장성 - 다중 테넌트 에이전트 플랫폼을 사용하면 내부 팀, 사업부 또는 클라이언트가 맞춤형 환경 없이 AI 기능을 더 빠르게 온보딩할 수 있습니다.
- 비용 효율성 - 공유 인프라는 중복 배포를 최소화하고 운영 비용을 통합하며 환경 간 유지 관리를 간소화합니다.
- 거버넌스 및 위험 감소 - 중앙 집중식 정책 제어, 자격 증명 모델 및 관찰성을 통해 에이전트는 모든 테넌트에서 보다 안전하고 규정을 준수하는 방식으로 운영할 수 있습니다.
- 서비스 재사용성 - 재사용을 촉진하고 중복을 줄이기 위해 테넌트 인식 에이전트를 보강, 규정 준수 또는 요약과 같은 내부 서비스로 제공할 수 있습니다.

멀티 테넌트 시스템의 예제 사용 사례는 다음과 같습니다.

- 자회사에 배포된 규정 준수 에이전트는 테넌트별 구성을 통해 로직을 현지 규정에 맞게 조정합니다. 따라서 각 리전에 대해 별도의 에이전트를 구축할 필요가 없습니다.
- 내부 워크플로 자동화 에이전트는 데이터 경계와 권한이 서로 다른 여러 부서에 서비스를 제공합니다. 작업 이행을 가속화하는 동안 격리를 유지합니다.

에이전트를 multi-tenant-aware 서비스로 설계하면 조직은 사일로화된 AI 이니셔티브의 오버헤드를 피할 수 있습니다. 대신 통합 인텔리전스 플랫폼을 조성합니다. 이 아키텍처는 확장 가능한 롤아웃, 운영 일관성 및 더 나은 ROI를 지원합니다. 또한 엔터프라이즈 전체에서 AI 채택을 더 쉽게 확장할 수 있습니다.

## 중점 영역 4: 자격 증명, 가드레일 및 관찰성을 통한 신뢰 구축

수행할 작업: “에이전트가 안전하고 예측 가능하게 행동할 것이라고 확신합니다. 특히 아무도 지켜보지 않을 때 더욱 그렇습니다.”

자율 에이전트는 기존 제어 모델에 도전합니다. 독립적으로 추론하고 행동할 수 있는 능력은 제대로 관리되지 않으면 위험을 초래합니다. 명확한 소유권, 감사 가능성 또는 정책 제약이 없으면 의도한 동작에서 드리프트될 수 있습니다. 조직의 신뢰를 구축하려면 기술적 신뢰성 이상의 것이 필요합니다. 설명 가능성, 책임 및 일관성이 필요합니다.

## 전략

신뢰할 수 있는 자율성의 백본으로 자격 증명 우선 제어 시스템을 구축합니다. 각 에이전트는 검증 가능한 자격 증명, 범위가 지정된 권한 및 추적 가능한 실행 기록으로 운영되어야 합니다. 에이전트는 테넌트 바인딩, 컨텍스트 액세스 상속, 가드레일 및 정책 엔진을 통한 런타임 적용을 포함하는 [제로 트러스트 프레임워크](#)에 포함되어야 합니다. 이를 통해 조직 규칙 및 위험 태세를 기반으로 에이전트 작업을 감사, 반전 또는 제한할 수 있습니다.

지능형 가드레일을 통해 런타임에 신뢰 적용을 포함합니다. 여기에는 행동 패턴 또는 워크로드 조건에 따른 속도 제어 및 제한, Auto Scaling과 함께 적용되는 리소스 경계, 위험을 평가하기 위한 결정 점수가 포함됩니다. 임계값을 초과할 때 human-in-the-loop 워크플로를 참여시키는 트리거를 빌드합니다.

또한 모든 에이전트는 투명하고 설명 가능해야 합니다. 로깅, 추적 및 추론 요약을 통해 구조화된 원격 측정을 포함하여 결정 로직을 공개합니다. 의사 결정 추적 및 영향 추적을 지원합니다. 이를 통해 에이전트 작업을 주요 지표 또는 결과에 다시 연결할 수 있습니다. 예상 동작 또는 정책과의 편차를 모니터링하는 드리프트 감지 메커니즘을 구현합니다.

에이전트 동작과 시스템 패턴을 지속적으로 관찰하는 반사 에이전트를 소개합니다. 실시간으로 이상 또는 불일치에 플래그를 지정해야 합니다. 이러한 에이전트는 기능의 재검증, 적응 또는 폐기를 시작할 수 있는 거버넌스 피드백 루프에 기여합니다.

에이전트 정책을 검토하고, 기능 변경을 승인하고, 인시던트 대응 프로토콜을 감독하는 거버넌스 보드를 구축합니다. 신뢰는 획득, 측정 및 지속적으로 강화되어야 합니다.

AWS 는이 신뢰 프레임워크를 구현하기 위한 강력한 기반을 제공합니다.

- [AWS Identity and Access Management \(IAM\)](#) 역할 기반 실행 및 권한 경계 적용
- [Amazon CloudWatch](#) 및는 완전한 가시성과 추적성을 [AWS X-Ray](#) 지원합니다.
- [Amazon GuardDuty](#) 및는 보안 이상 또는 정책 드리프트를 [AWS Config](#) 감지합니다.

이러한 서비스를 함께 사용하면 ID 적용, 런타임 안전 및 대규모 신뢰 기반 거버넌스가 가능합니다. 이를 통해 자율 시스템을 강력하고 신뢰할 수 있게 만들 수 있습니다.

## 신뢰할 수 있는 자율성의 비즈니스 가치

에이전트가 자율화됨에 따라 신뢰는 엔터프라이즈 채택, 거버넌스 및 운영 성능의 중요한 동인이 됩니다. 자격 증명, 관찰성 및 가드레일의 기반을 설정하면 조직이 거버넌스 또는 제어를 희생하지 않고 에이전트 AI를 민감한 도메인으로 확장할 수 있습니다.

주요 비즈니스 동인은 다음과 같습니다.

- 거버넌스 보증 - 강력한 자격 증명 모델, 감사 추적 및 권한 경계는 규정 준수 위험을 줄이고 규제 조정을 지원합니다.
- 운영 연속성 - 런타임 가드레일과 이상 탐지는 의도하지 않은 동작을 방지하고 엣지 케이스 장애로부터 자체 복구를 지원하는 데 도움이 됩니다.
- 이해관계자 신뢰도 - 의사 결정 설명 가능성 및 원격 측정은 내부 이해관계자, 위험 관리자 및 외부 감사자와 신뢰를 구축합니다.
- 인시던트 복원력 - 임베디드 관찰성은 문제가 발생할 때 근본 원인 분석 및 응답 시간을 가속화합니다.

사용 사례 예시:

- 금융 서비스에서 사기 탐지 에이전트는 추론을 공개하고, 추적 가능한 자격 증명으로 모든 작업을 기록하고, 범위가 좁은 IAM 역할로 운영해야 합니다.
- 의료 분야에서 자율 분류 에이전트는 런타임 안전 검사를 시행하고, 임계값이 충족되면 인적 검토로 에스컬레이션하고, 임상 감독을 위한 전체 로그를 제공해야 합니다.

신뢰 메커니즘을 에이전트 수명 주기에 포함하면 조직은 시스템이 책임과 함께 자율적으로 운영되도록 허용할 수 있습니다. 이 토대는 위험을 줄이고 에이전트가 투명성과 무결성으로 비즈니스를 대신하여 행동할 수 있도록 지원합니다.

궁극적으로 신뢰할 수 있는 자율성은 사용자와 경영진 모두에게 핵심 운영 전반에 걸쳐 지능형 에이전트를 확장할 수 있는 자신감을 제공하여 채택을 가속화합니다.

## 중점 영역 5: 수명 주기 관리

수행할 작업: “팀이 카오스나 히어로틱 없이 시간이 지남에 따라 에이전트를 개선할 수 있는지 확인하세요.”

코드만 형성되는 기존 애플리케이션과 달리 에이전트 동작은 프롬프트, 메모리, 도구 및 훈련 컨텍스트로도 형성됩니다. 이러한 요소는 시간이 지남에 따라 드리프트됩니다. 드리프트는 신뢰성을 약화시키고, 비용을 부풀리고, 디버깅을 거의 불가능하게 만듭니다. 수명 주기 제어가 없으면 에이전트는 가치 제공을 중단하고 위험 누적을 시작합니다.

## 전략

에이전트용 DevOps(AgentOps)를 연습으로 설정합니다. 에이전트에 맞게 조정된 CI/CD 파이프라인을 통합합니다. 이러한 파이프라인을 사용하여 프롬프트 출력을 테스트하고, 도구 통합을 검증하고, 비용 성능 동작을 프로파일링합니다. 프롬프트, 정책 및 모델 상호 작용의 버전 기록을 유지 관리합니다.

관찰성 데이터의 피드백 루프를 사용하여 재훈련, 프롬프트 튜닝 또는 에이전트 사용 중지를 시작합니다. 개선 레지스터와 같은 시스템 전체의 반영 메커니즘을 통합하여 학습을 체계화합니다.

결정 정확도, 지연 시간, 비용 및 신뢰성을 보여주는 성능 원격 측정 대시보드를 구축합니다. 팀은 인프라를 사용하여 AWS 수명 주기 관리를 간소화하고 가속화하기 위해 에이전트 툴킷을 사용할 수 있습니다. 한 가지 예로 [Strands Agents SDK](#)가 있습니다. 이 SDK는 프롬프트 버전 관리, 도구 등록 및 [AWS CodePipeline](#), [AWS Cloud Development Kit \(AWS CDK\)](#),와 AWS 서비스같은 CI/CD 통합을 위한 구조화된 도구를 제공합니다. [AWS Lambda](#). 또한 에이전트 아티팩트 및 훈련 데이터를 저장하려면 [Amazon S3](#) 및 [Amazon Elastic File System\(Amazon EFS\)](#)을 사용합니다. [AWS Step Functions](#)를 사용하여 복잡한 재훈련 또는 검증 워크플로를 자동화합니다. 에이전트가 LLM 오케스트레이션 외에 사용자 지정 모델 튜닝 또는 미세 조정 워크플로가 필요한 경우 [Amazon SageMaker AI](#)를 사용할 수 있습니다. 수명 주기 원칙은 에이전트를 실험에서 내구성이 뛰어나고 진화하는 자산으로 변환합니다.

시간이 지남에 따라 수명 주기 시스템은 혁신의 백본을 형성합니다. 이를 통해 민첩성을 갖춘 기능을 다시 구성, 재훈련 및 재배포할 수 있습니다. 이렇게 하면 에이전트 계층이 살아 있는 시스템으로 변환되어 피드백과 기회 모두에 대응하여 진화할 수 있습니다.

## 수명 주기 관리의 비즈니스 가치

효과적인 수명 주기 관리의 에이전트 성능 및 비용 효율성의 주요 동인입니다. 이를 통해 지능형 에이전트는 진화함에 따라 정확하고 안정적이며 가치에 부합하는 결과를 계속 제공할 수 있습니다. 에이전트는 기본적으로 가치를 유지하지 않습니다. 변화하는 비즈니스 요구 사항, 워크플로 및 데이터 환경에 맞춰 진화해야 합니다. 체계적인 AgentOps 팀은 에이전트가 정확하고 효율적이며 시간이 지남에 따라 엔터프라이즈 목표에 부합할 수 있도록 지원합니다.

주요 비즈니스 동인은 다음과 같습니다.

- 성능 일관성 - 지속적인 테스트, 프롬프트 검증 및 재학습은 에이전트가 변화하는 조건 및 데이터 세트에서 의사 결정 품질을 유지하는 데 도움이 됩니다.

- 비용 최적화 - 원격 측정 기반 프로파일링은 비효율적인 도구, 높은 토큰 프롬프트 또는 불필요한 실행을 식별합니다. 그런 다음을 조정하여 운영 비용을 줄일 수 있습니다.
- 더 빠른 반복 - CI/CD를 사용한 수명 주기 자동화는 개발 주기를 가속화하여 팀이 자신 있게 에이전트를 실험, 배포 및 개선하는 데 도움이 됩니다.
- 위험 감소 - 프롬프트 버전 관리, 롤백 지원 및 구조화된 평가 메커니즘은 회귀를 방지하고 안전하고 신뢰할 수 있는 변경 관리를 지원하는 데 도움이 됩니다.

사용 사례의 예로 다음과 같은 것들이 있습니다.

- 고객 지원 에이전트의 지연 시간, 모델 비용 및 사용자 피드백을 모니터링합니다. 관찰성은 비용 급증을 보여주며, 이는 임베디드 프롬프트 및 폴백 모델 로직을 다시 조정하라는 메시지를 표시합니다.
- 계약 요약 에이전트는 법률 팀의 피드백을 기반으로 업데이트됩니다. 버전이 지정된 프롬프트는 프로덕션 릴리스 전에 샌드박스 환경에서 테스트되어 안전 및 품질을 지원합니다.

구조화된 수명 주기 관리를 통해 조직은 사후 대응 유지 관리에서 선제적이고 지속적인 개선으로 전환합니다. 에이전트는 비즈니스 목표에 따라 측정, 개선 및 재검증되는 적응형 디지털 자산이 됩니다. 이 방법은 에이전트 에코시스템을 성능, 비용 인식 및 복원력이 뛰어난 시스템으로 변환하여 변화에 보조를 맞추면서 내구성 있는 가치를 제공합니다.

## 중점 영역 6: 에이전트 모델을 비즈니스 모델에 맞게 조정

수행할 작업: “지속적인 투자를 정당화할 수 있도록 영향을 보여줘.”

기술적으로 역량이 있는 에이전트도 비즈니스 성과와 관련이 없는 경우 부채가 됩니다. 에이전트는 효율성, 수익화 또는 전략적 차별화를 제공해야 합니다. 그러나 대부분의 기업은 에이전트가 요금, 패키징 또는 사용 모델에 어떻게 맞는 지 정의하는 데 어려움을 겪고 있습니다. 비즈니스 가치에 대한 명확한 조정이 없으면 규모 조정을 정당화하거나 투자를 유지하기가 어렵습니다.

### 전략

제품 관리 관행을 채택합니다. 에이전트를 측정 가능한 ROI가 있는 수익 창출 가능한 서비스로 취급합니다. 결정, 세션 또는 결과를 기반으로 요금 전략을 정의합니다. 그런 다음 에이전트 기능을 고객 세그먼트 또는 내부 사업부에 맞는 계층형 상품으로 패키징합니다.

지속 가능성을 촉진하기 위해 조직은 에이전트 배포를 통해 직접 가치와 성장 승수를 모두 캡처해야 합니다. 다음 ROI 지표를 사용하여 즉각적인 값을 측정하는 것이 좋습니다.

- 결정당 비용 - 에이전트 처리 비용을 동등한 사람과 비교하여 벤치마킹합니다.
- 시간 압축 - 더 빠른 판매 또는 승인과 같은 가속화된 주기의 가치를 정량화합니다.
- 오류 감소 - 향상된 정확도, 일관성 및 규정 준수로 비용 절감을 측정합니다.

에이전트는 이러한 즉각적인 이점 외에도 다음과 같은 장기 성장 기회를 활용할 수 있습니다.

- 기능 스택킹 - 에이전트 서비스를 결합하여 도메인별 수직 솔루션을 생성합니다.
- 네트워크 효과 - 조정이 유용한 다중 에이전트 에코시스템을 통해 가치를 높입니다.
- 시장 확장 - 외부에서 사용할 수 있는 에이전트 지원 서비스를 통해 새로운 수익 스트림을 생성합니다.

비즈니스 지표(예: 비용 절감, 변환 리프트 또는 time-to-resolution)에서 피드백 루프를 생성하여 지속적인 에이전트 진화를 주도합니다. 사용량 원격 측정 및 사용자 만족도 점수를 분석하여 가치 조정 및 로드맵 우선순위를 구체화합니다. 에이전트 기능을 비즈니스 모델에 직접 연결하면 조직은 기술적 성과 뿐만 아니라 지속 가능하고 복합적인 가치를 창출할 수 있습니다.

다음은 강력한 추적 및 수익화 프레임워크를 제공하여 이러한 조정을 AWS 서비스 지원합니다.

- [AWS Cost Explorer](#) 및 [Amazon CloudWatch](#)는 에이전트당 비용 및 운영 효율성에 대한 인사이트를 제공합니다.
- [Amazon API Gateway](#)를 사용하면 에이전트 엔드포인트에 대해 측정된 액세스, 속도 제한 및 계층화된 요금을 사용할 수 있습니다.
- [AWS Marketplace](#)는 에이전트 및 에이전트 솔루션을 상용 제품으로 게시하기 위한 채널을 제공합니다.

이러한 서비스를 통해 에이전트 기능을 엔터프라이즈 성장 및 수익화 전략에 맞는 확장 가능한 가치 기반 디지털 상품으로 변환할 수 있습니다.

## 에이전트 AI를 위한 소프트웨어 제공 개선

최신 소프트웨어 제공은 사용자가 배송하는 시스템을 제어한다는 단순한 가정을 바탕으로 이루어집니다. 요구 사항을 정의하고, 로직을 작성하고, 예상 결과와 비교하여 테스트하고, 예측 가능한 서비스를 배포합니다. 애자일 및 DevOps 접근 방식도 여전히 각 스프린트가 결정적이고 검증 가능하며 주로 사람의 감독 내에서 무언가를 제공한다는 원칙에 의존합니다.

에이전트 AI는 해당 기반을 업데이트합니다. 에이전트 시스템은 스크립트를 따르는 대신 해석, 추론 및 적응합니다. 이들의 동작은 작성하는 코드, 작업하는 컨텍스트, 제공된 입력, 액세스할 수 있는 도구, 할당된 목표에 따라 달라집니다. 즉, 주문을 따르지 않고 결과를 추구합니다.

이렇게 하면 제어와 정렬에 대한 전달이 줄어듭니다. 지침을 제공하는 대신 동작 방식을 만들어야 합니다. 즉, 기존 소프트웨어 개발 수명 주기(SDLC)는 로직 기반 인적 제어 시스템을 위해 설계되었기 때문에 더 이상 적합하지 않습니다.

이 섹션은 다음 주제를 포함합니다:

- [에이전트 AI의 의도 영역](#)
- [에이전트 AI의 전송 수명 주기 개선](#)
- [에이전트 AI를 위한 팀 준비](#)

### 에이전트 AI의 의도 영역

정의, 구축, 테스트 및 릴리스와 같은 엄격한 단계 대신 자율성, 불확실성 및 출현을 수용하는 모델이 필요합니다. 대신 의도 영역을 사용합니다. 의도 영역은 제약 내에서 에이전트가 자율적으로 작동할 수 있는 경계 공간을 정의합니다. 목표는 모든 작업의 마이크로 관리에서 에이전트가 안전하게 행동하고, 배우고, 협업할 수 있는 환경 설계로 전환하는 것입니다. 대상(원하는 성과), 이유(의도) 및 가드레일(제한, 정책 및 신뢰 경계)을 지정합니다. 이러한 경계와 이 정보를 고려하여 에이전트는 방법을 파악합니다.

조립 라인 대신 환경을 영공으로 생각하세요. 입력할 수 있는 사람, 수행할 수 있는 작업 및 이동할 수 있는 위치를 제어합니다. 하지만 안에 들어가면 필요에 따라 자유롭게 탐색할 수 있습니다. 이것이 에이전트 시스템이 혼란 없이 확장되는 방식입니다.

이는 단지 철학적 전환이 아니라 실용적인 전환입니다. 에이전트 기반 시스템의 비결정적 출력은 단위 테스트를 통해 완전히 테스트할 수 없습니다. 정적 바이너리처럼 버전이 지정될 수 없습니다. 에이전트는 시간이 지남에 따라 변화하고, 새 데이터에 적응하며, 예측할 수 없는 방식으로 다른 시스템과 상호

작용합니다. 기존 모델을 사용하여 제공하려고 하면 확장할 수 없는 아키텍처가 취약해집니다. 최악의 경우 실제로 관리할 수 없는 시스템에 대한 잘못된 신뢰도로 이어집니다.

팀이 의도 기반 제공을 수용하면 두 가지 이점이 있습니다.

- 가장 중요한 위치 제어 - 출력 대신 경계를 정의합니다.
- 위임을 통한 확장성 - 에이전트가 사람이 하드코딩할 수 없는 복잡성을 처리할 수 있습니다.

이는 격리된 프로토타입에서 반복적이고 안정적으로 가치를 제공할 수 있는 실제 프로덕션급 에이전트 시스템으로 이동하는 방법입니다.

## 에이전트 AI의 전송 수명 주기 개선

지능형 적응형 동작을 지원하려면 SDLC를 결정적 제어에서 적응형 의도로 재구성해야 합니다. 다음은 에이전트 AI용 기존 SDLC를 발전시키는 데 필요한 변경 사항입니다.

- 계획은 의도 설계가 됩니다. 팀은 목표, 제약 조건 및 예상 에이전트 동작을 정의합니다. 정책 및 성공 기준은 로직이 아닌 정렬로 구성됩니다.
- 아키텍처가 비계가 됩니다. 팀은 모든 의사 결정 경로를 스크립팅하는 대신 역할, 인터페이스, 가드 레일, 대체 메커니즘 및 관찰성을 정의하는 데 중점을 둡니다.
- 테스트는 동작 평가가 됩니다. 팀은 특정 출력을 어설션하는 대신 에이전트가 허용 범위 내에 있는지 확인하고 다양한 입력에서 의도를 이행합니다.
- 배포는 지속적인 오케스트레이션이 됩니다. 에이전트 시스템은 실시간 튜닝을 지원하는 런타임 제어, 실시간 모니터링 및 피드백 채널과 함께 배포됩니다.
- 반복은 피드백과 적응이 됩니다. 팀은 기존 코드 변경 패치 주기 대신 에이전트가 진화하는 방식, 성공하는 위치 또는 드리프트 시점을 관찰합니다. 필요에 따라 팀은 업데이트된 제약 조건, 재훈련, 제어 메커니즘 추가 또는 수정에 개입합니다.

반복, 실험 및 빠른 피드백에 초점을 맞춘 기존 사례는 중간에 있습니다. 에이전트 시스템으로의 전환은 애자일 원칙을 거부하는 것이 아닙니다. 실제로 이는 자연스러운 진화입니다. 애자일 사고는 엄격한 계획보다 적응성, 피드백 및 작업 솔루션을 강조합니다. 이는 실시간으로 컨텍스트를 학습, 적응 및 대응하는 에이전트 시스템의 특성에 완벽하게 부합합니다. 이미 짧은 주기를 실행하고, 가정을 빠르게 검증하고, 지속적인 전달을 통해 불확실성을 관리하는 경우 이러한 전환을 주도할 준비가 잘 되어 있습니다.

하지만 주요 차이점이 있습니다. 기존 애자일 접근 방식은 전달되는 사물이 결정적이라고 가정합니다. 일단 구축되면 사물이 일관되고 예측 가능하게 작동하여 동일한 입력에 대해 반복 가능한 결과를 얻을

수 있다고 가정합니다. 이러한 반복성을 통해 디버깅, 테스트 및 반복 작업을 자신 있게 수행할 수 있습니다. 에이전트 시스템은 해당 모델을 중단합니다. 확률적이고 상황에 민감하며 독립적으로 진화할 수 있습니다. 즉, 스토리 완성을 기반으로 한 속도 추적, 엄격한 수락 기준 또는 결정적 스프린트 계획과 같은 일부 애자일 관행은 유용성이 떨어집니다.

기존 SDLC의 다음 측면은 에이전트 AI에 적용됩니다.

- 반복 개발 및 제공
- 고객 피드백을 기본 신호로 사용
- 부서 간 협업
- 지속적인 통합 및 배포

기존 SDLC의 다음 측면은 에이전트 AI를 위해 진화해야 합니다.

- 의도에 맞게 완료를 재정의합니다. 에이전트의 동작이 정의된 제약 내에서 의도한 목표를 충족하는지 여부에 중점을 둡니다.
- 수락 기준에서 행동 가드레일로 전환합니다.
- 지속적 학습 및 신뢰를 지원하는 관찰성, 설명성 및 피드백 메커니즘을 포함하는 런타임 준비 상태를 포함하도록 완료의 정의를 확장합니다.
- 선결제 계획보다 실시간 피드백 루프 및 동작 추적 우선 순위 지정

좋은 소식은 SDLC 플레이북을 배포할 필요가 없다는 것입니다. 코드 관리에서 행동 형성으로 발전시키기만 하면 됩니다. 에이전트 시스템에서 성공은 소프트웨어가 실행되는지 여부뿐만 아니라 어떻게 작동하는지에 관한 것입니다.

## 에이전트 AI를 위한 팀 준비

소프트웨어 엔지니어링은 사라지지 않습니다. 진화하고 있습니다. 작업은 함수 작성에서 지능형 동작을 위한 프레임워크 및 제어 메커니즘 형성으로 전환됩니다. 에이전트 AI의 세계에서 구축은 더 이상 어려운 부분이 아니며, 급증 관리는 더 이상 어려운 부분입니다. 대부분의 엔지니어링 팀에서 진화는 기술적 도약이 아닌 사고방식 변화처럼 느껴집니다. "시스템은 어떻게 하나요?"라고 묻는 대신 질문은 "무엇을 추구할 수 있도록 지원했는가? 그리고 어떻게 하면 진행 중인지 알 수 있을까?"가 됩니다.

엔지니어링 팀의 경우 에이전트 AI로 발전하려면 다음과 같은 변경 사항이 필요합니다.

- 문화적 변화 - 팀은 완전히 제어하지 못하는 시스템의 불확실성과 자율성에 익숙해져야 합니다.

- 새로운 역할 - 의도 디자이너, 행동 테스터 및 관찰성 엔지니어가 제공의 핵심이 됩니다.
- 공유 언어 - 팀은 한때 사양 및 테스트 사례가 필요했던 것처럼 목표, 가드레일 및 성공 신호에 대한 명확하고 공유된 이해가 필요합니다.

생성형 AI가 성숙해지면 고객, 제품 및 운영과 상호 작용하는 에이전트 시스템이 늘어납니다. 성공한 조직은 최상의 모델을 가진 조직이 아닙니다. 에이전트를 신뢰도, 제어 및 속도로 실제 워크플로에 통합할 수 있습니다. 즉, 제공 모델과 엔지니어링 팀이 함께 발전해야 합니다. 의도 영역은 이를 위한 추상화를 제공합니다. 이를 통해 책임을 포기하지 않고 자율성을 운영할 수 있습니다. 또한 하드 코딩할 수 없는 시스템을 관리하는 데 도움이 되는 팀 간 공유 프레임워크를 제공합니다.

에이전트 AI를 위한 팀 준비에 대한 자세한 내용은 이 가이드 [의 대규모 에이전트 AI를 위한 비즈니스 준비](#) 섹션을 참조하세요.

## 대규모 에이전트 AI를 위한 비즈니스 준비

이 가이드에 설명된 [중점 영역](#)이 수렴됨에 따라 에이전트 AI는 격리된 함수에서 기능 플랫폼으로 이해될 수 있는 통합 인텔리전스 계층으로 전환합니다. 이 플랫폼은 단순히 작업을 실행하는 것이 아닙니다. 도메인 간에 발전, 적응 및 조정됩니다. 에이전트는 혁신을 가속화하고, 인지 부하를 줄이고, 전사적으로 측정 가능한 결과를 창출하는 모듈식의 재사용 가능하고 검색 가능한 서비스가 됩니다. 이 플랫폼 보기는 운영 모델 전체에 포함된 확장 가능한 인텔리전스의 단계를 설정합니다.

에이전트 AI를 운영하려면 지능형 에이전트를 배포하는 것 이상의 것이 필요합니다. 이를 위해서는 비즈니스가 팀을 구성하고, 프로세스를 설계하고, 기술을 관리하는 방식에 근본적인 혁신이 필요합니다. 클라우드 또는 DevOps 재정의된 운영 모델로 전환하는 것처럼 에이전트 AI는 의사 결정 자동화, 지속적인 학습 및 자율 조정의 새로운 시대를 도입합니다. 성공은이 새로운 운영 철학을 중심으로 시스템, 사람 및 프로세스를 조정하는 데 달려 있습니다.

이 섹션은 다음 주제를 포함합니다:

- [팀 및 소유권 모델 조정](#)
- [변경 및 조직 준비 관리](#)
- [상호 운용성 및 협업을 위한 설계](#)
- [에이전트 패브릭으로 거버넌스 구축](#)
- [의사 결정 우선 운영 사고방식 채택](#)
- [목적 및 의도를 사용한 조정](#)

### 팀 및 소유권 모델 조정

성숙도를 향한 첫 번째 단계는 부서 간 정렬입니다. 기업은 분산 시스템 아키텍트, 소프트웨어 엔지니어, 제품 소유자, 규정 준수 책임자, 플랫폼 아키텍트와 같은 AI/ML 실무자와 도메인 전문가를 포함하는 AgentOps 팀을 구성해야 합니다. 이러한 팀은 설계 및 배포부터 재훈련 및 모니터링에 이르기까지 에이전트의 전체 수명 주기를 공동으로 소유합니다.

에이전트 프로비저닝 및 릴리스는 코드형 인프라 및 자동 배포에 [AWS Cloud Development Kit \(AWS CDK\)](#) [AWS CodePipeline](#) 및를 사용하는 것과 같은 클라우드 네이티브 관행을 따라야 합니다. 이 구조는 공동 책임을 촉진하고 반복을 가속화합니다. DevOps가 개발 및 운영을 통합하는 것처럼 AgentOps는 인텔리전스를 거버넌스 및 실행과 연결합니다.

이러한 팀의 효율성을 높이려면 공유 언어도 필요합니다. 비즈니스 이해관계자는 [에이전트가 무엇인지](#), 에이전트가 [어떻게 운영되는지](#), 에이전트가 [어떤 결과를 가져오는지](#) 이해해야 합니다. 훈련 및 내

부 활성화가 필수적입니다. 에이전트를 설명하고이 멘탈 모델을 일상적인 대화에 임베딩함으로써 조직은 더 광범위한 참여와 보다 일관된 혁신을 실현할 수 있습니다.

를 사용하여 에이전트의 개발 및 통합을 가속화하기 위해 AWS 서비스팀은 에이전트 스캐폴딩, 구성 및 패키징을 위한 CLI 기반 도구를 제공하는 [Strands Agents SDK](#)와 같은 프레임워크를 채택할 수 있습니다. Strands Agents는 [Amazon Bedrock](#), [AWS Lambda Amazon EventBridge](#), AWS CDK 및 등의 인프라에서 AWS 원활하게 작동하도록 설계되었습니다 AWS CodePipeline. 프로덕션 등급 표준을 유지하면서 신속한 프로토타입 생성 및 배포가 가능합니다.

하지만 구조와 도구만으로는 충분하지 않습니다. 에이전트 AI 규모를 조정하려면 도입이 조직 전체에 뿌리를 내리도록 하기 위해 신중한 문화적, 교육적 및 리더십 준비가 필요합니다.

## 변경 및 조직 준비 관리

에이전트 AI를 성공적으로 확장하려면 인프라 또는 지능형 에이전트를 배포하는 것 이상의 것이 필요합니다. 조직 변화에 대한 구조화된 접근 방식이 필요합니다. 여기에는 문화적 준비 상태, 기술 개발, 지표 기반 피드백 루프, 채택이 의도적이고 지속 가능한지 확인하기 위한 경영진 조정이 포함됩니다.

### 문화적 진화 촉진

- 에이전트를 대체가 아닌 팀원으로 배치하여 저항을 줄이고 신뢰를 구축합니다.
- 에이전트 기능 및 제한 사항에 대해 투명하게 전달하여 현실적인 기대치를 설정합니다.
- 에이전트가 결정을 더 높은 기관에 에스컬레이션하거나 프로세스의 일부를 인간 공동 작업자에게 위임해야 하는 경우를 위한 명확한 핸드오프 프로토콜을 설정합니다.

### 기술 개발 프레임워크 설정

- 엔지니어, 제품 관리자, 도메인 책임자 및 규정 준수 책임자에 맞게 역할 기반 교육을 제공합니다.
- 우수성 센터를 만들어 모범 사례, 도구 패턴 및 재사용 가능한 자산을 공유합니다.
- 멘토십 프로그램을 통해 AI 전문가를 도메인 전문가와 페어링하여 지식 격차를 해소합니다.

### 지표 및 피드백 루프 정의

- 전략적 가치에 맞게 기술 및 비즈니스 KPIs를 앵커링하여 영향을 평가합니다. 가치의 예로는 결정 지연 시간, 해결 정확도, 비용 절감 등이 있습니다.
- 사용자 피드백을 체계적이고 지속적으로 캡처하여 마찰점과 채택 문제를 표시합니다.

- 정기적인 회고를 수행하여 에이전트 성능, 사용 추세 및 개선 기회를 평가합니다.

## 위에서 리더십 조정

- 에이전트 이니셔티브를 전략적 성과 및 ROI에 연결하여 경영진의 후원을 받습니다.
- 기술 및 비즈니스 리더십을 모두 포함하는 부서 간 거버넌스 위원회를 구성합니다.
- 모든 조직 수준에서 명확성과 참여를 위해 커뮤니케이션 전략을 조정합니다.

이러한 체계적인 변경 관리 접근 방식은 기술 구현이 조직의 성숙도에 부합하도록 합니다. 신뢰, 채택 및 장기적인 비즈니스 가치를 위한 기반을 마련합니다.

## 상호 운용성 및 협업을 위한 설계

격리된 에이전트 배포는 로컬 성공을 제공합니다. 그러나 에이전트가 서로 동적으로 검색, 호출 및 협업할 수 있으면 엔터프라이즈 가치가 나타납니다. 즉, 에이전트 등록, 인증 및 기능 교환에 대한 표준을 정의합니다. 이 아키텍처는 모놀리스에서 마이크로서비스로의 전환을 미러링합니다. 마이크로서비스는 복잡한 문제를 함께 해결하는 구성 가능하고 재사용 가능하며 느슨하게 결합된 단위입니다.

[A2A](#) 및 [MCP](#)와 같은 새로운 프로토콜은 기본입니다. 는 에이전트, 도구 및 메모리 시스템에서 의미론적 상호 운용성을 지원합니다. A2A는 피어 수준 상호 작용을 지원하므로 에이전트가 작업 소유권을 협상하고 컨텍스트를 공유하며 워크플로를 조정할 수 있습니다. MCP는 에이전트와 환경 간에 컨텍스트 데이터를 교환하기 위한 공유 스키마를 제공하여 이를 보완합니다. 함수 호출, APIs 액세스 및 상태 유지 관리 방법을 표준화합니다. 이러한 프로토콜은 함께 에이전트 에코시스템 전반의 확장성, 일관성 및 장기 유지 관리를 촉진합니다.

거버넌스는 여전히 중요합니다. arbiter 에이전트와 같은 제어 계층은 중앙 집중식 병목 현상 없이 정책 인식 위임을 활성화합니다. 이러한 에이전트는 신뢰 브로커 역할을 합니다. 다른 에이전트가 자체적으로 조직할 수 있도록 경계를 적용합니다. 에이전트 협업은 조직이 민첩성과 신뢰로 에이전트 AI 에코시스템을 확장하는 데 도움이 됩니다.

## 에이전트 패브릭으로 거버넌스 구축

자율성이 높을수록 위험이 커집니다. 거버넌스는 첫날부터 에이전트 아키텍처에 포함되어야 합니다. 여기에는 에이전트가 수행할 수 있는 작업의 범위를 지정하는 정책 경계 정의, 에이전트가 누구를 대신하여 행동하는지 결정하는 자격 증명 모델 적용, 설명 가능성 및 추적 가능성 구현이 포함됩니다. 관찰성 시스템은 에이전트 워크플로 전반에 걸쳐 중앙 집중식 로깅 및 분산 추적 [AWS X-Ray](#)을 제공하는

[Amazon CloudWatch](#) 및와 같은 서비스를 사용하여 에이전트 동작에 대한 원격 측정을 캡처해야 합니다. 반사 에이전트는 이러한 원격 측정 피드를 기반으로 성능을 지속적으로 감사하고 평가할 수 있습니다.

또한 에이전트 에코시스템이 성숙함에 따라 거버넌스도 진화해야 합니다. 에이전트의 역량과 자율성이 향상됨에 따라 감독 메커니즘의 적응력이 향상되어야 합니다. 정책 업데이트, 기능 게이팅 및 런타임 동작 제약 조건은 동적이어야 하며 대규모로 적용할 수 있어야 합니다. 신뢰는 볼트 온 기능이 아닙니다. 아키텍처, 동작 및 프로세스를 통해 지속적으로 강화됩니다. [AWS Identity and Access Management \(IAM\)](#) 및 [AWS AppConfig](#)는 에이전트 간에 보안 자격 증명, 런타임 권한 경계 및 환경별 동작 토글을 적용하는 데 중요한 역할을 합니다.

## 의사 결정 우선 운영 사고방식 채택

기존 자동화는 사전 정의된 스크립트 또는 워크플로를 더 빠르고 안정적으로 실행하는 프로세스 효율성에 중점을 둡니다. 반면 에이전트 AI는 의사 결정 우선 자동화를 도입합니다. 에이전트는 컨텍스트를 평가하고, 옵션을 측정하며, 실시간으로 동작을 조정합니다. 실행 우선 사고 방식에서 의사 결정 우선 사고 방식으로 전환하려면 성공 지표와 결과에 대한 새로운 사고 방식이 필요합니다. 에이전트성 AI의 성공은 작업 완료로만 성공을 측정하는 대신 결정이 의도, 정책 및 진화하는 조건에 얼마나 잘 부합하는지로 측정됩니다.

조직은 작업 완료 또는 주기 시간만 측정하는 대신 결정 품질, time-to-action 및 변화에 대한 대응성을 평가해야 합니다. KPIs에는 다음과 같은 지표가 포함되어야 합니다.

- 결정 품질 - 에이전트가 특정 사용자 또는 시나리오에 대한 응답을 얼마나 잘 개인화했습니까? 비즈니스 목표 및 사용자 컨텍스트에 맞는 미묘한 결정을 내렸나요?
- Time-to-action 에이전트가 상황을 얼마나 빠르고 지능적으로 평가하고 대응했습니까? 적응력이 뛰어나고 인간처럼 느껴질 만큼 지연 시간이 짧았나요?
- 인지 오프로드 - 에이전트가 인간을 대신하여 처리할 수 있었던 수동 분석, 분류 또는 일상적인 의사결정의 양 노력을 줄였나요, 아니면 그냥 이동했나요?

의사 결정 우선 사고방식을 수용하는 기업은 복원력이 뛰어나고 적응력이 뛰어나며 새로운 수준의 복잡성으로 운영할 수 있습니다.

## 목적 및 의도를 사용한 조정

에이전트 AI를 성공적으로 확장하는 것은 더 많은 도구를 실험하는 것이 아닙니다. 내구성이 뛰어난 엔터프라이즈 인텔리전스 계층을 구축하는 것입니다. 이를 위해서는 플랫폼 인프라, 운영 문화, 거버넌스

프레임워크 및 전략적 조정에 대한 투자가 필요합니다. 기업은 의도적인 접근 방식을 채택해야 합니다. 에이전트는 실험이 아니라 디지털 운영 모델의 핵심 구성 요소로 취급해야 합니다.

[AWS Well-Architected Framework](#)에 맞게 조정하면 시스템이 신뢰성, 보안, 성능 효율성 및 비용 최적화에 대한 엔터프라이즈 표준을 충족할 수 있습니다. [Strands Agents SDK](#)와 같은 도구는 구조화된 프롬프트, 도구 등록 및 CI/CD 준비 상태를 제공하여이 여정을 가속화할 수 있습니다. 이를 통해 팀은 익숙한 AWS 워크플로를 사용하여 실험에서 확장 가능한 제공으로 전환할 수 있습니다.

에이전트 AI는 도구가 아니라 인텔리전스가 운영에 포함되는 방식의 변화입니다. 그에 따라 준비하는 조직은 점점 더 복잡한 세계에서 더 많은 것을 자동화하고, 더 스마트하게 운영하고, 더 빠르게 적응하고, 지속적인 이점을 창출할 수 있습니다.

## 에이전트 AI를 운영하기 위한 결론

에이전트 AI는 기술적 변화 이상을 나타냅니다. 엔터프라이즈를 위한 새로운 운영 체제의 출현을 나타냅니다. 이러한 혁신을 수용하는 조직은 좁은 자동화 사용 사례를 넘어 운영의 기초로 인텔리전스를 구축합니다. 이 전환은 의사 결정 방법, 시스템 적응 방법, 대규모 결과 실현 방법을 재설계하는 것입니다.

복잡성, 실시간 수요 및 정보 오버로드가 증가하는 시대에 스크립트 자동화의 기존 모델은 한계에 도달했습니다. 이제 성공은 인지하고, 추론하고, 행동하고, 진화하는 시스템을 만들기 위해 인텔리전스를 워크플로에 직접 임베드하는 기능에 달려 있습니다. 에이전트 AI는 자율성을 목적과 일치시키고, 의사 결정을 거버넌스와 일치시키고, 적응성을 책임과 일치시킬 수 있습니다.

이 전환을 위해서는 실행 우선 사고에서 의사 결정 우선 사고로 전환해야 합니다. 에이전트 시스템은 단순히 지침을 따르지 않습니다. 목표를 해석하고, 장단점을 평가하고, 정의된 제약 내에서 결과를 추구합니다. 이 맥락에서 성공은 작업 완료뿐만 아니라 측정됩니다. 또한 실시간으로 내린 결정의 품질, 민첩성 및 설명 가능성으로 측정됩니다. 조직은 불확실성에서 지능적으로 운영하는 에이전트를 지원하기 위해 지표, 인센티브 및 시스템 설계를 재고해야 합니다.

에이전트 AI를 운영하는 것은 plug-and-play 업그레이드가 아닙니다. 이는 아키텍처 및 문화 혁신입니다. 이를 위해서는 수명 주기 관리, 신뢰 적용, 상호 운용성 및 비즈니스 모델에 대한 조정 전반에 걸쳐 훈련된 관행이 필요합니다. 또한 의도 영역 셰이핑, 런타임 가드레일 임베딩, 에이전트 동작을 전략적 결과에 지속적으로 조정하는 등 전달 모델의 진화를 요구합니다. 팀은 에이전트 성능 및 안전을 위해 공유 언어, 공유 소유권 및 공유 책임을 채택해야 합니다.

엔터프라이즈 준비는 이 새로운 환경에서 누가 적응하는지 확인할 수 있습니다. 조직은 확장하고 장기적인 가치를 창출하는 내부 지원, AgentOps 기능 및 거버넌스 프레임워크에 투자해야 합니다. 성공하는 사람은 더 스마트한 시스템을 구축할 수 있으며 적응력이 뛰어나고 복원력이 뛰어나며 통찰력 기반 비즈니스를 구축할 수도 있습니다.

이 가이드는 토대를 마련합니다. 전략을 실행에 연결하고 조직이 확장 가능한 지능형 에이전트 플랫폼을 구축할 수 있도록 준비합니다. 의 에이전트 AI에 대한 광범위한 콘텐츠 시리즈는 보완적인 지침을 AWS 제공합니다. 이 시리즈의 다른 가이드를 보려면 AWS 권장 가이드 웹 사이트의 [에이전트 AI](#)를 참조하세요. 이 콘텐츠 시리즈는 원칙과 의도를 바탕으로 자율성을 운영하기 위한 로드맵을 제공합니다.

시작하려면 에이전트가 속도, 정확도 또는 응답성에서 측정 가능한 개선을 제공할 수 있는 영향력이 큰 의사 결정 공간을 식별합니다. 그런 다음 계층, 거버넌스 및 피드백 루프가 있는 집중된 파일럿 에이전트를 배포합니다. 이를 사용하여 가치 가설을 검증하고, 내부 추진력을 생성하고, 접근 방식에 대한 신뢰를 구축합니다. 학습을 통한 모멘텀 구성.

에이전트 AI는 대상이 아니라 비즈니스와 함께 발전하는 기능 계층입니다. 이는 인프라로서 인텔리전스에 대한 장기 전환을 나타냅니다. 이 공간을 주도하는 조직은 더 많은 것을 자동화하고, 더 빠르게 대응하고, 더 잘 적응하고, 엔터프라이즈 규모로 복잡성을 탐색할 수 있는 운영 모델을 구축할 수 있습니다.

# 에이전트 AI를 운영하기 위한 리소스

## AWS 서비스

다음 AWS 서비스 및 기능은에서 에이전트 AI 시스템을 구축하고 운영하는 데 도움이 될 수 있습니다.  
AWS 클라우드

- [Amazon API Gateway](#)는 에이전트 기능을 확장 가능한 것으로 표시하고 사용량 기반 요금을 제공합니다.
- [AWS AppConfig](#)는 테넌트 또는 환경 전반의 에이전트에 대한 런타임 구성 관리 및 기능 전환을 제공합니다.
- [Amazon Bedrock](#)은 에이전트가 추론, 생성 및 프롬프트 실행에 사용할 수 있는 파운데이션 모델 서비스입니다.
- [AWS Cloud Development Kit \(AWS CDK\)](#)는 에이전트 스택을 배포하고 관리하는 데 사용할 수 있는 코드형 인프라 서비스입니다.
- [AWS CloudTrail](#)는 에이전트 활동, 감사 추적 및 통합 동작을 추적할 수 있도록 이벤트 기록을 기록합니다.
- [Amazon CloudWatch](#)는 에이전트 성능 및 다중 에이전트 협업 동작을 모니터링하기 위한 로그, 지표 및 경보를 관리할 수 있습니다.
- [AWS CodePipeline](#)는 에이전트 코드를 테스트, 검증 및 배포하는 데 사용할 수 있는 CI/CD 자동화를 제공합니다.
- [Amazon Cognito](#)는 다중 에이전트 시스템에서 사용자 및 테넌트 인증을 관리하는 데 사용할 수 있는 자격 증명 서비스입니다.
- [AWS Config](#)는 에이전트 정책 및 환경 구성에 대한 규정 준수 및 드리프트 감지를 제공합니다.
- [AWS Cost Explorer](#)는 에이전트 수준 사용량을 추적하고 비용을 조정하여 ROI를 극대화할 수 있습니다.
- [Amazon DynamoDB](#)는 에이전트 메모리, 개선 로그 및 컨텍스트 상태에 사용할 수 있는 스토리지 서비스입니다.
- [Amazon Elastic File System\(Amazon EFS\)](#)은 워크플로 전반의 에이전트 협업 또는 중간 처리에 사용할 수 있는 공유 파일 시스템입니다.
- [Amazon EventBridge](#)는 에이전트 패브릭에서 작업을 라우팅하고 통신을 오케스트레이션하는 데 사용할 수 있는 코어 이벤트 버스입니다.

- [Amazon EventBridge 파이프](#)는 에이전트와 서비스를 연결하기 위한 이벤트 수집 및 라우팅을 간소화할 수 있습니다.
- [Amazon GuardDuty](#)는 안전한 에이전트 실행을 지원할 수 있는 위협 탐지 및 이상 모니터링을 제공합니다.
- [AWS Identity and Access Management \(IAM\)](#)를 사용하면 에이전트 실행 및 데이터 액세스에 대한 세분화된 권한을 정의할 수 있습니다.
- [AWS Lambda](#)는 에이전트 로직을 실행하고 드론을 스웸할 수 있는 상태 비저장 컴퓨팅 서비스입니다.
- [AWS Marketplace](#)는 에이전트 기능을 상용 제품으로 제공하는 데 사용할 수 있는 외부 배포 플랫폼입니다.
- [AWS Organizations](#)는 다중 테넌트 에이전트 인프라를 관리하는 데 도움이 되는 교차 계정 거버넌스 및 정책 적용 서비스입니다.
- [AWS Organizations 서비스 제어 정책](#)은 계정 또는 조직 단위 수준에서 권한을 제어하기 위한 가드레일 역할을 합니다.
- [Amazon Quick](#)은 데이터를 분석하고, 시각화를 생성하고, 워크플로를 자동화하고, 조직 전체의 다른 사용자와 협업하는 데 도움이 되는 생성형 AI 기반 비즈니스 인텔리전스(BI) 플랫폼입니다.
- [AWS Resource Access Manager \(AWS RAM\)](#)를 사용하면 계정과 에이전트 서비스 간에 기능을 공유할 수 있습니다.
- [Amazon SageMaker AI](#)는 기본 모델을 넘어 모델 훈련, 미세 조정 및 추론에 사용할 수 있는 서비스입니다.
- [Amazon Simple Storage Service\(Amazon S3\)](#)는 프롬프트 라이브러리, 모델 아티팩트 및 에이전트 생성 데이터를 위한 객체 스토리지를 제공합니다.
- [AWS Step Functions](#)는 다중 에이전트 흐름을 조정하고 파이프라인을 재훈련하는 데 도움이 되는 워크플로 엔진입니다.
- [AWS X-Ray](#)는 에이전트 결정 흐름 및 서비스 종속성을 추적하는 데 사용할 수 있는 분산 추적을 제공합니다.

## 기타 AWS 리소스

- [에서 에이전트 AI의 기초 AWS](#)
- [의 에이전트 AI 패턴 및 워크플로 AWS](#)
- [의 에이전트 AI 프레임워크, 프로토콜 및 도구 AWS](#)
- [에서 에이전트 AI를 위한 서버리스 아키텍처 구축 AWS](#)

- [에서 에이전트 AI를 위한 멀티 테넌트 아키텍처 구축 AWS](#)

## 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="#">최초 게시</a>	—	2025년 8월 12일

# AWS 권장 가이드 용어집

다음은 AWS 권장 가이드에서 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: 온프레미스 Oracle 데이터베이스를 AWS 클라우드의 Amazon Relational Database Service(Amazon RDS) for Oracle로 마이그레이션합니다.
- 재구매(드롭 앤드 쇼) - 일반적으로 기존 라이선스에서 SaaS 모델로 전환하여 다른 제품으로 전환합니다. 예: 고객 관계 관리(CRM) 시스템을 Salesforce.com으로 마이그레이션합니다.
- 리호스팅(리프트 앤드 시프트) - 애플리케이션을 변경하지 않고 클라우드로 이동하여 클라우드 기능을 활용합니다. 예: 온프레미스 Oracle 데이터베이스를 AWS 클라우드클라우드의 EC2 인스턴스에 있는 Oracle로 마이그레이션합니다.
- 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 로 마이그레이션합니다 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

## A2A(Agent-to-Agent)

작업 위임 및 상태 전송 agent-to-agent 공동 작업을 위한 상태 저장 프로토콜입니다.

## ABAC

[속성 기반 액세스 제어](#)를 참조하세요.

## 추상화된 서비스

[관리형 서비스](#)를 참조하세요.

## ACID

[원자성, 일관성, 격리성, 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. 더 유연하지만 [액티브 패시브 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.

## 에이전트

목표를 달성하기 위한 도구를 사용하여 자율적으로 추론, 계획 및 조치를 취할 수 있는 AI 시스템입니다.

## 에이전트 운영

대규모 프로덕션 환경에서 AI 에이전트를 구축, 테스트, 배포 및 실행하기 위한 운영 사례입니다.

## 집계 함수

행 그룹에서 작동하고 그룹에 대한 단일 반환 값을 계산하는 SQL 함수입니다. 집계 함수의 예로 SUM 및 MAX가 있습니다.

## AI

[인공 지능](#)을 참조하세요.

### AIOps

[인공 지능 운영](#)을 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 사용하도록 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 탐색 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 [용 ABAC 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 격리 AWS 리전 되고 동일한 리전의 다른 가용 영역에 저렴하고 지연 시간이 짧은 네트워크 연결을 제공하는 내의 고유한 위치입니다.

## AWS 클라우드 채택 프레임워크(AWS CAF)

조직이 클라우드로 성공적으로 전환하기 위한 효율적이고 효과적인 계획을 개발하는 AWS 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, 교육 및 커뮤니케이션에 대한 지침을 제공합니다. 자세한 내용은 [AWS CAF 웹사이트](#)와 [AWS CAF 백서](#)를 참조하세요.

## AWS 워크로드 검증 프레임워크(AWS WQF)

데이터베이스 마이그레이션 워크로드를 평가하고, 마이그레이션 전략을 권장하고, 작업 견적을 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

### 악성 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 직접 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. [엔디안](#)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 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

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

### bot

인터넷을 통해 자동화된 태스크를 실행하고 인적 활동이나 상호 작용을 시뮬레이션하는 소프트웨어 애플리케이션입니다. 인터넷에서 정보를 인덱싱하는 웹 크롤러와 같이 유용하거나 이로운 봇도 있습니다. 악성 봇이라고 하는 다른 일부 봇은 개인 또는 조직을 방해하거나 해를 입히기 위한 봇입니다.

## 봇넷

[맬웨어](#)에 감염되고 봇 허더 또는 봇 운영자와 같은 단일 당사자가 제어하는 [봇](#) 네트워크입니다. 봇넷은 봇의 규모와 봇의 영향 범위를 확대하는 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 [About branches](#)(GitHub 설명서)를 참조하십시오.

## 긴급 액세스 권한

예외적인 상황에서 승인된 프로세스를 통해 사용자가 일반적으로 액세스할 권한이 없는데 액세스할 수 있는 빠른 방법입니다. 자세한 내용은 AWS Well-Architected 지침의 [Implement break-glass procedures](#) 지표를 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 Cloud Adoption Framework](#)를 참조하세요.

## 카나리 배포

최종 사용자에게 제공하는 느린 증분 릴리스 버전입니다. 확신이 들면 새 버전을 배포하고 현재 버전을 완전히 교체합니다.

## CCoE

[클라우드 혁신 센터](#)를 참조하세요.

## CDC

[데이터 캡처 변경](#)을 참조하세요.

## 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 중단 이벤트를 도입합니다. [AWS Fault Injection Service \(AWS FIS\)](#)를 사용하여 AWS 워크로드에 스트레스를 주고 응답을 평가하는 실험을 수행할 수 있습니다.

## CI/CD

[지속적 통합 및 지속적 전송](#)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.

## 시민 개발자

전문 기술 없이 노코드/로우코드 플랫폼을 사용하여 AI 애플리케이션을 생성하는 비즈니스 사용자입니다.

## 클라이언트측 암호화

대상이 데이터를 AWS 서비스 수신하기 전에 로컬에서 데이터를 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 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 연결되어 있습니다.

## 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 AWS 클라우드로 마이그레이션할 때 일반적으로 거치는 4단계는 다음과 같습니다.

- 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

이러한 단계는 Stephen Orban이 블로그 게시물 [The Journey Toward Cloud-First and the Stages of Adoption](#) on the AWS 클라우드 Enterprise Strategy 블로그에서 정의했습니다. 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 리포지토리로 GitHub 또는 Bitbucket Cloud가 포함됩니다. 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 Amazon SageMaker AI는 CV에 대한 이미지 처리 알고리즘을 제공합니다.

## 구성 드리프트

워크로드의 경우 구성이 예상되는 상태에서 변경됩니다. 이로 인해 워크로드가 규정을 준수하지 않을 수 있으며, 이는 일반적으로 점진적이고 의도되지 않은 작업입니다.

## 구성 관리 데이터베이스(CMDB)

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

## 규정 준수 팩

규정 준수 및 보안 검사를 사용자 지정하기 위해 조합할 수 있는 AWS Config 규칙 및 문제 해결 작업의 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/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 비전](#)을 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 탈중앙화된 데이터 소유권을 제공하는 아키텍처 프레임워크입니다.

### 데이터 최소화

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

### 데이터 경계

신뢰할 수 있는 자격 증명만 예상 네트워크에서 신뢰할 수 있는 리소스에 액세스하도록 하는 데 도움이 되는 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 보통 많은 양의 기록 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

[데이터 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.

## 심층 방어

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 컨트롤을 AWS 추가하여 리소스를 보호할 수 있습니다. 예를 들어, 심층 방어 접근 방식은 다단계 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 서비스를 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 AWS에서 보안 제어 구현의 [탐지 제어](#)를 참조하세요.

## 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.

## 차원 테이블

[스타 스키마](#)에서 팩트 테이블의 정량적 데이터에 대한 데이터 속성을 포함하는 더 작은 테이블을 말합니다. 차원 테이블 속성은 일반적으로 텍스트 필드나 텍스트처럼 동작하는 개별 숫자입니다. 이러한 속성은 보통 쿼리 제약, 필터링 및 결과 세트 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)

[재해](#)로 인한 가동 중지 시간 및 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 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 참조하세요.

## 드리프트 감지

기준이 되는 구성과의 편차 추적을 말합니다. 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 가치 흐름 매핑](#)을 참조하세요.

## E

### EDA

[탐색 데이터 분석](#)을 참조하세요.

### EDI

[전자 데이터 교환](#)을 참조하세요.

## 엣지 컴퓨팅

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

## 전자 데이터 교환(EDI)

조직 간 비즈니스 문서의 자동화된 교환을 나타냅니다. 자세한 내용은 [전자 데이터 교환\(EDI\)이란 무엇인가요?](#)를 참조하세요.

## 암호화

사람이 읽을 수 있는 일반 텍스트 데이터를 사이퍼텍스트로 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(VPC) 설명서의 [엔드포인트 서비스 생성](#)을 참조하십시오.

## 엔터프라이즈 리소스 계획(ERP)

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

## ERP

[엔터프라이즈 리소스 계획](#)을 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

[스타 스키마](#)의 중앙 테이블입니다. 비즈니스 운영에 대한 정량적 데이터를 저장합니다. 일반적으로 팩트 테이블은 측정값이 있는 열 및 차원 테이블에 대한 외래 키가 있는 열과 같이 두 가지 열 유형을 포함합니다.

## 빠른 실패

개발 수명 주기를 줄이기 위해 빈번한 증분 테스트를 사용하는 철학입니다. 애자일 접근 방식의 핵심입니다.

## 장애 격리 경계

에서 장애의 영향을 제한하고 워크로드의 복원력을 개선하는 데 도움이 되는 가용 영역, AWS 리전 컨트롤 플레인 또는 데이터 플레인과 같은 AWS 클라우드경계입니다. 자세한 내용은 [AWS 장애 격리 경계](#)를 참조하세요.

## 기능 브랜치

[브랜치](#)를 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 SHAP(Shapley Additive Descriptions) 및 통합 그래디언트와 같은 다양한 기법을 통해 계산할 수 있는 수치 점수로 표현됩니다. 자세한 내용은 [기계 학습 모델 해석 가능성을 참조하세요 AWS](#).

## 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.

## 퓨샷 프롬프팅

유사한 태스크를 수행하도록 요청하기 전에 [LLM](#)에 태스크와 원하는 출력을 보여주는 몇 가지 예제를 제공합니다. 이 기법은 모델이 프롬프트에 포함된 예제(샷)에서 학습하는 컨텍스트 내 학습을 적용합니다. 퓨샷 프롬프팅은 특정 형식 지정, 추론 또는 분야별 지식이 필요한 태스크에 효과적일 수 있습니다. [제로샷 프롬프팅](#)도 참조하세요.

## FGAC

[세분화된 액세스 제어](#)를 참조하세요.

## 세분화된 액세스 제어(FGAC)

여러 조건을 사용하여 액세스 요청을 허용하거나 거부합니다.

## 플래시컷 마이그레이션

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

### FM

[파운데이션 모델](#)을 참조하세요.

#### 파운데이션 모델(FM)

일반화되고 레이블이 지정되지 않은 데이터의 대규모 데이터세트에서 훈련된 대규모 딥 러닝 신경망입니다. FM은 언어 이해, 텍스트 및 이미지 생성, 자연어 대화와 같은 다양한 일반 태스크를 수행할 수 있습니다. 자세한 내용은 [파운데이션 모델이란?](#)을 참조하세요.

#### FM 게이트웨이

[파운데이션 모델에](#) 대한 액세스를 제어하고 정규화하는 중앙 집중식 중개자입니다. LLM 게이트웨이이라고도 합니다.

## G

### 생성형 AI

대량의 데이터에서 훈련되었으며 간단한 텍스트 프롬프트를 사용하여 이미지, 비디오, 텍스트, 오디오와 같은 새 콘텐츠와 아티팩트를 생성할 수 있는 [AI](#) 모델의 하위 세트입니다. 자세한 내용은 [생성형 AI란 무엇인가요?](#)를 참조하세요.

#### 지리적 차단

[지리적 제한](#)을 참조하세요.

#### 지리적 제한(지리적 차단)

Amazon CloudFront에서 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 설명서의 [콘텐츠의 지리적 배포 제한](#)을 참조하십시오.

#### Gitflow 워크플로

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

## 골든 이미지

시스템 또는 소프트웨어의 새 인스턴스를 배포하기 위한 템플릿으로 사용되는 해당 시스템 또는 소프트웨어의 스냅샷입니다. 예를 들어 제조 분야에서는 골든 이미지를 사용하여 여러 디바이스에서 소프트웨어를 프로비저닝할 수 있으며 이를 통해 딥이스 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(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, 이는 AWS Config Amazon GuardDuty AWS Security Hub CSPM, , AWS Trusted Advisor Amazon Inspector 및 사용자 지정 AWS Lambda 검사를 사용하여 구현됩니다.

## 가드레일(AI)

책임감 있고 안전한 AI 동작을 보장하기 위해 [에이전트](#) 입력 및 출력을 필터링, 검증 및 제약하는 안전 메커니즘입니다.

# H

## HA

[고가용성](#)을 참조하세요.

## 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 [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 데이터세트에서 보류되는 레이블이 지정된 기록 데이터의 일부입니다. 홀드아웃 데이터를 사용하여 모델 예측을 홀드아웃 데이터와 비교해 모델 성능을 평가할 수 있습니다.

## human-in-the-loop(HitL)

중요한 결정 시점에서 인적 검토 및 승인을 위해 [에이전트](#) 실행이 일시 중지되는 워크플로 패턴입니다.

## 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. 핫픽스는 긴급하기 때문에 일반적인 DevOps 릴리스 워크플로 외부에서 실행됩니다.

## 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

[코드형 인프라](#)를 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.

## 변경 불가능한 인프라

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

## 인바운드(수신) VPC

AWS 다중 계정 아키텍처에서 애플리케이션 외부에서 네트워크 연결을 수락, 검사 및 라우팅하는 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.

## 증분 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 통해 제조 프로세스의 현대화를 나타내기 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\(IoT\) 디지털 트랜스포메이션 전략 구축](#)을 참조하십시오.

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

### IoT

[사물 인터넷](#)을 참조하세요.

### IT 정보 라이브러리(ITIL)

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

### IT 서비스 관리(ITSM)

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 라이브러리](#)를 참조하세요.

## ITSM

[IT 서비스 관리](#)를 참조하세요.

## 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

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.

### 대규모 언어 모델(LLM)

방대한 양의 데이터에서 사전 훈련된 딥 러닝 [AI](#) 모델입니다. LLM은 질문에 대한 답변, 문서 요약, 텍스트를 다른 언어로 번역, 문장 완성과 같은 여러 태스크를 수행할 수 있습니다. 자세한 내용은 [대규모 언어 모델\(LLM\)이란 무엇인가요?](#)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 참조하십시오.

### 리프트 앤드 시프트

[7R](#)을 참조하세요.

## 리틀 엔디안 시스템

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

## LLM

[대규모 언어 모델](#)을 참조하세요.

## 하위 환경

[환경](#)을 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

[브랜치](#)를 참조하세요.

### 맬웨어

컴퓨터 보안 또는 프라이버시를 위협하도록 설계된 소프트웨어입니다. 맬웨어는 컴퓨터 시스템을 방해하거나 민감한 정보를 유출하거나 무단 액세스 권한을 확보할 수 있습니다. 맬웨어의 예로 바이러스, 웜, 랜섬웨어, 트로이 목마, 스파이웨어, 키로거 등이 있습니다.

### 관리형 서비스

AWS 서비스는 인프라 계층, 운영 체제 및 플랫폼을 AWS 운영하며 사용자는 엔드포인트에 액세스하여 데이터를 저장하고 검색합니다. 관리형 서비스의 예로 Amazon Simple Storage Service(Amazon S3) 및 Amazon DynamoDB가 있습니다. 이를 추상화된 서비스라고도 합니다.

### 제조 실행 시스템(MES)

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

## MAP

[Migration Acceleration Program](#)을 참조하세요.

## MCP

[모델 컨텍스트 프로토콜](#)을 참조하세요.

### Model Context Protocol(MCP)

[에이전트 간??? 통신](#)을 위한 상태 비저장 프로토콜입니다.

### MCP 서버

[모델 컨텍스트 프로토콜](#)을 통해 하나 이상의 [도구를](#) 노출하는 서비스입니다.

### 메커니즘

도구를 생성하고 도구 채택을 유도한 다음 조정을 위해 결과를 검사하는 전체 프로세스입니다. 메커니즘은 작동 시 자체적으로 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 큐 원격 분석 전송(MQTT)

리소스 제약이 있는 [IoT](#) 디바이스에 대한 [게시 및 구독](#) 패턴을 기반으로 하는 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 [AWS 서버리스 서비스를 사용하여 마이크로서비스 통합](#)을 참조하세요.

### 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 마이크로서비스를 업데이트, 배포 및 조정할 수 있습니다. 자세한 내용은 [에서 마이크로서비스 구현을 참조하세요 AWS](#).

## Migration Acceleration Program(MAP)

조직이 클라우드로 전환하기 위한 강력한 운영 기반을 구축하고 초기 마이그레이션 비용을 상쇄하는 데 도움이 되는 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 스프린트에서 일하는 운영, 비즈니스 분석가 및 소유자, 마이그레이션 엔지니어, 개발자, 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 리호스팅합니다.

## Migration Portfolio Assessment(MPA)

AWS 클라우드로 마이그레이션하는 비즈니스 사례를 검증하기 위한 정보를 제공하는 온라인 도구입니다. 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 AWS 컨설턴트와 APN 파트너 컨설턴트가 무료로 사용할 수 있습니다.

## 마이그레이션 준비 상태 평가(MRA)

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

### 마이그레이션 전략

워크로드를 AWS 클라우드로 마이그레이션하는 데 사용되는 접근 방식입니다. 자세한 내용은 [이 용어집의 7R 항목과 조직을 동원하여 대규모 마이그레이션 가속화](#)를 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 [AWS 클라우드에서 애플리케이션을 현대화하기 위한 전략](#)을 참조하세요.

### 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 [AWS 클라우드에서 애플리케이션의 현대화 준비 상태 평가](#)를 참조하세요.

### 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

[Migration Portfolio Assessment](#)를 참조하세요.

### MQTT

[메시지 큐 원격 분석 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 [변경 불가능한 인프라](#)를 모범 사례로 사용할 것을 권장합니다.

## O

### OAC

[오리진 액세스 제어](#)를 참조하세요.

### OAI

[오리진 액세스 ID](#)를 참조하세요.

### OCM

[조직 변경 관리](#)를 참조하세요.

## 오프라인 마이그레이션

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

## OI

[운영 통합](#)을 참조하세요.

### OLA

[운영 수준 계약](#)을 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\(OPC-UA\)](#)를 참조하세요.

### Open Process Communications - Unified Architecture(OPC-UA)

산업 자동화를 위한 Machine-to-Machine(M2M) 통신 프로토콜입니다. OPC-UA는 데이터 암호화, 인증 및 권한 부여 체계에 관한 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 상태 검토(ORR)

인시던트 및 잠재적 장애의 범위를 이해, 평가 또는 예방하거나 줄이는 데 도움이 되는 질문 체크리스트 및 관련 모범 사례입니다. 자세한 내용은 AWS Well-Architected Framework의 [운영 준비 상태 검토\(ORR\)](#)를 참조하세요.

### 운영 기술(OT)

물리적 환경에서 작동하여 산업 운영, 장비 및 인프라를 제어하는 하드웨어 및 소프트웨어 시스템입니다. 제조 분야에서 OT 및 정보 기술(IT) 시스템의 통합은 [Industry 4.0](#) 트랜스포메이션의 주요 중점 사항입니다.

### 운영 통합(OI)

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

### 조직 트레일

조직 AWS 계정 내 모든에 대한 모든 이벤트를 로깅 AWS CloudTrail 하는에서 생성된 추적입니다 AWS Organizations. 이 트레일은 조직에 속한 각 AWS 계정에 생성되고 각 계정의 활동을 추적합니다. 자세한 내용은 CloudTrail 설명서의 [Creating a trail for an organization](#)을 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. AWS 마이그레이션 전략에서는 클라우드 채택 프로젝트에 필요한 변경 속도 때문에이 프레임워크를 인력 가속화라고 합니다. 자세한 내용은 [사용 가이드](#)를 참조하십시오.

## 오리진 액세스 제어(OAC)

CloudFront에서 Amazon Simple Storage Service(S3) 콘텐츠를 보호하기 위해 액세스를 제한하는 고급 옵션입니다. OAC는 AWS KMS (SSE-KMS)를 사용한 모든 서버 측 암호화 AWS 리전와 S3 버킷에 대한 동적 PUT 및 DELETE 요청에서 모든 S3 버킷을 지원합니다.

## 오리진 액세스 ID(OAI)

CloudFront에서 Amazon S3 콘텐츠를 보호하기 위해 액세스를 제한하는 옵션입니다. OAI를 사용하면 CloudFront는 Amazon S3가 인증할 수 있는 보안 주체를 생성합니다. 인증된 보안 주체는 특정 CloudFront 배포를 통해서만 S3 버킷의 콘텐츠에 액세스할 수 있습니다. 더 세분화되고 향상된 액세스 제어를 제공하는 [OAC](#)도 참조하십시오.

## ORR

[운영 준비 상태 검토](#)를 참조하세요.

## OT

[운영 기술](#)을 참조하세요.

## 아웃바운드(송신) VPC

AWS 다중 계정 아키텍처에서 애플리케이션 내에서 시작된 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 정보](#)를 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 가능 로직 컨트롤러](#)를 참조하세요.

## PLM

[제품 수명 주기 관리](#)를 참조하세요.

## 정책

권한 정의([ID 기반 정책](#) 참조), 액세스 조건 지정([리소스 기반 정책](#) 참조), AWS Organizations 내 조직의 모든 계정에 대한 최대 권한 정의([서비스 제어 정책](#) 참조)와 같은 작업을 수행할 수 있는 객체입니다.

## 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 저장소를 사용하면 마이크로서비스를 더 쉽게 구현하고 성능과 확장성을 높일 수 있습니다.

## 포트폴리오 평가

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

## 조건자

보통 WHERE 절에 있는 true 또는 false를 반환하는 쿼리 조건입니다.

## 푸시다운 조건자

전송 전에 쿼리의 데이터를 필터링하는 데이터베이스 쿼리 최적화 기법입니다. 이렇게 하면 관계형 데이터베이스에서 검색하고 처리해야 하는 데이터의 양이 줄고 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 IAM 엔티티입니다. 이 엔티티는 일반적으로 , AWS 계정 IAM 역할 또는 사용자의 루트 사용자입니다. 자세한 내용은 IAM 설명서의 [역할 용어 및 개념](#)의 보안 주체를 참조하십시오.

## 개인 정보 보호 중심 설계

전체 개발 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 작업](#)을 참조하십시오.

## 선제적 제어

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

## 제품 수명 주기 관리(PLM)

설계, 개발 및 출시부터 성장 및 성숙도를 거쳐 거부 및 제거에 이르기까지 전체 수명 주기 동안 제품의 데이터 및 프로세스 관리를 나타냅니다.

## 프로덕션 환경

[환경](#)을 참조하세요.

## 프로그래밍 가능 로직 컨트롤러(PLC)

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

## 프롬프트 체이닝

한 [LLM](#) 프롬프트의 출력을 다음 프롬프트의 입력으로 사용하여 더 나은 응답을 생성합니다. 이 기법은 복잡한 태스크를 하위 태스크로 나누거나 예비 응답을 반복적으로 세부 조정하거나 확장하는데 사용됩니다. 이를 통해 모델 응답의 정확성과 관련성을 개선하고 보다 세분화되고 개인화된 결과를 얻을 수 있습니다.

## 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(pub/sub)

여러 마이크로서비스에서 비동기 통신을 지원하여 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 명령어와 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 매트릭스

[Responsible, Accountable, Consulted, Informed\(RACI\)](#)를 참조하세요.

### RAG

[검색 증강 생성](#)을 참조하세요.

### 랜섬웨어

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

### RASCI 매트릭스

[Responsible, Accountable, Consulted, Informed\(RACI\)](#)를 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.

### 리아키텍팅

[7R](#)을 참조하세요.

### 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

[7R](#)을 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 AWS 리전 지정](#)을 참조하세요.

### 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

[7R](#)을 참조하세요.

### 릴리스

배포 프로세스에서 변경 사항을 프로덕션 환경으로 승격시키는 행위입니다.

## 재배치

[7R](#)을 참조하세요.

## 리플랫폼

[7R](#)을 참조하세요.

## 재구매

[7R](#)을 참조하세요.

## 복원력

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

## 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 결정권자 (A), 업무 수행 조언자 (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 AWS에서 보안 제어 구현의 [대응 제어](#)를 참조하세요.

## retain

[7R](#)을 참조하세요.

## 사용 중지

[7R](#)을 참조하세요.

## 검색 증강 세대(RAG)

응답을 생성하기 전에 [LLM](#)이 훈련 데이터 소스 외부에 있는 신뢰할 수 있는 데이터 소스를 참조하는 [생성형 AI](#) 기술입니다. 예를 들어 RAG 모델은 조직의 지식 기반 또는 사용자 지정 데이터에 대

한 시맨틱 검색을 수행할 수 있습니다. 자세한 내용은 [검색 증강 생성\(RAG\)이란 무엇인가요?](#)를 참조하세요.

## 교체

공격자가 자격 증명에 액세스하는 것을 더욱 어렵게 만들기 위해 [보안 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

[목표 복구 시점\(RPO\)](#)을 참조하세요.

## RTO

[목표 복구 시간\(RTO\)](#)을 참조하세요.

## 런북

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

# S

## SAML 2.0

많은 ID 제공업체(idP)에서 사용하는 개방형 표준입니다. 이 기능을 사용하면 연동 SSO(Single Sign-On)를 AWS Management Console 사용할 수 있으므로 사용자는 조직의 모든 사용자에게 대해 IAM에서 사용자를 생성하지 않고도 로그인하거나 AWS 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 획득](#)을 참조하세요.

## SCP

[서비스 제어 정책](#)을 참조하세요.

## 보안 암호

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 AWS Secrets Manager 설명서의 [Secrets Manager 보안 암호란 무엇인가요?](#)를 참조하세요.

## 보안 중심 설계

전체 개발 프로세스에서 보안을 고려하는 시스템 엔지니어링에서의 접근 방식입니다.

## 보안 제어

위협 행위자가 보안 취약성을 악용하는 능력을 방지, 탐지 또는 감소시키는 기술적 또는 관리적 가드레일입니다. 보안 제어는 [예방](#), [감지](#), [대응](#), [선제적](#)과 같은 기본적인 네 가지 보안 제어 유형으로 구분됩니다.

## 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 응답 자동화

보안 이벤트에 자동으로 응답하거나 이를 해결하도록 설계된 사전 정의되고 프로그래밍된 작업입니다. 이러한 자동화는 보안 모범 사례를 구현하는 데 도움이 되는 [탐지](#) 또는 [대응](#) AWS 보안 제어 역할을 합니다. 자동화된 응답 작업의 예로 VPC 보안 그룹 수정, Amazon EC2 인스턴스 패치 적용 또는 자격 증명 교체 등이 있습니다.

## 서버 측 암호화

데이터를 AWS 서비스 수신하는가 대상에서 데이터를 암호화합니다.

## 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 작

업을 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 수준에 관한 계약(SLA)

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

## 서비스 수준 지표(SLI)

오류 발생률, 가용성 또는 처리량과 같은 서비스의 성능 측면에 대한 측정값입니다.

## 서비스 수준 목표(SLO)

[서비스 수준 지표](#)로 측정되는 서비스의 상태를 나타내는 목표 지표입니다.

## 공동 책임 모델

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

## 새도우 AI

조직 내 관리형 채널 외부에서 구축되거나 사용되는 승인되지 않은 [AI](#) 애플리케이션입니다.

## SIEM

[보안 정보 및 이벤트 관리 시스템](#)을 참조하세요.

## 단일 장애점(SPOF)

애플리케이션을 중단시킬 수 있는 애플리케이션의 중요한 단일 구성 요소에서 발생하는 장애입니다.

## SLA

[서비스 수준 계약](#)을 참조하세요.

## SLI

[서비스 수준 지표](#)를 참조하세요.

## SLO

[서비스 수준 목표](#)를 참조하세요.

### 분할 앤 시드 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 [AWS 클라우드에서 애플리케이션을 현대화하기 위한 단계별 접근 방식](#)을 참조하세요.

## SPOF

[단일 장애점](#)을 참조하세요.

### 스타 스키마

하나의 큰 팩트 테이블을 사용하여 트랜잭션 또는 측정된 데이터를 저장하고 하나 이상의 더 작은 차원 테이블을 사용하여 데이터 속성을 저장하는 데이터베이스 조직 구조입니다. 이 구조는 [데이터 웨어하우스](#)에서 또는 비즈니스 인텔리전스 목적으로 사용하도록 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 방법](#)을 참조하십시오.

### 서브넷

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

### 감독 제어 및 데이터 획득(SCADA)

제조 분야에서 하드웨어와 소프트웨어를 사용하여 물리적 자산과 프로덕션 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 진행되는 시스템 테스트입니다. [Amazon CloudWatch Synthetics](#)를 사용하여 이러한 테스트를 생성할 수 있습니다.

## 시스템 프롬프트

[LLM](#)에 컨텍스트, 명령 또는 지침을 제공하여 동작을 지시하는 기법입니다. 시스템 프롬프트는 컨텍스트를 설정하고 사용자와의 상호 작용을 위한 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, 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

[환경](#)을 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.

### tool

[에이전트](#)가 외부 시스템에서 작업을 수행하기 위해 호출할 수 있는 함수 또는 API입니다.

## Transit Gateway

VPC와 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 지정하는 서비스에 대한 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

피자 두 판이면 충분한 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.

### 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 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.

### 웜 데이터

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

### 창 함수

현재 레코드와 어떤 식으로든 관련된 행 그룹에서 계산을 수행하는 SQL 함수입니다. 창 함수는 이동 평균을 계산하거나 현재 행의 상대적 위치를 기반으로 행 값에 액세스하는 등의 태스크를 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

[Write Once, Read Many\(WORM\)](#)를 참조하세요.

## WQF

[AWS Workload Qualification Framework](#)를 참조하세요.

## Write Once Read Many(WORM)

데이터를 한 번 쓰고 데이터가 삭제되거나 수정되지 않도록 하는 스토리지 모델입니다. 권한 있는 사용자는 필요한 만큼 여러 번 데이터를 읽을 수 있지만 데이터를 변경할 수는 없습니다. 이 데이터 스토리지 인프라는 [변경 불가능](#)한 항목으로 간주됩니다.

## Z

### 제로데이 익스플로잇

[제로데이 취약성](#)을 악용하는 공격(일반적으로 맬웨어)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.

### 제로샷 프롬프팅

태스크를 수행하기 위해 [LLM](#)에 명령을 제공하지만 안내에 도움이 되는 예제(샷)는 제공하지 않습니다. LLM은 사전 훈련된 지식을 사용하여 태스크를 처리해야 합니다. 제로샷 프롬프팅의 효과는 태스크의 복잡성과 프롬프트의 품질에 따라 달라집니다. [퓨샷 프롬프팅](#)도 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.