에서 에이전트 AI의 기초 AWS

AWS 권장 가이드

AWS 권장 가이드: 에서 에이전트 AI의 기초 AWS

Table of Contents

에서 에이전트 AI의 기초 AWS

Aaron Sempf, Amazon Web Services

2025년 7월(문서 기록)

점점 더 지능적이고 분산된 자율 시스템 세계에서 환경, 상태에 대한 이유를 인식하고 의도에 따라 행동할 수 있는 엔터티인 에이전트의 개념은 기초가 되었습니다. 에이전트는 단순히 지침을 실행하는 프로그램이 아니라 사용자, 시스템 또는 조직을 대신하여 결정을 내리는 목표 지향적인 컨텍스트 인식 엔터티입니다. 이들의 등장은 소프트웨어를 구축하고 생각하는 방식의 변화를 반영합니다. 즉, 절차 로직 및 사후 대응 자동화에서 자율성과 목적으로 작동하는 시스템으로의 전환입니다.

AI, 분산 시스템 및 소프트웨어 엔지니어링의 교차점에는 에이전트 AI라고 하는 강력한 패러다임이 있습니다. 이 차세대 지능형 시스템은 적응형 동작, 복잡한 조정 및 위임된 의사 결정을 수행할 수 있는 소프트웨어 에이전트로 구성됩니다.

이 가이드에서는 최신 소프트웨어 에이전트를 정의하는 원칙을 소개하고 에이전트 AI에 대한 진화를 간략하게 설명합니다. 이 변화를 설명하기 위해 가이드는 개념적 배경을 제공한 다음 소프트웨어 에이전트의 에이전트 AI로의 진화를 추적합니다.

- 소프트웨어 에이전트 소개는 소프트웨어 에이전트를 정의하고, 이를 기존 소프트웨어 구성 요소와 비교하며, 확립된 프레임워크에서 끌어와 에이전트 행동을 기존 자동화와 차별화하는 필수 특성을 소개합니다.
- 소프트웨어 에이전트의 목적은 소프트웨어 에이전트가 존재하는 이유, 수행하는 역할, 해결하는 문제, 지능형 위임을 활성화하고, 인지 부하를 줄이고, 동적 환경에서 적응형 동작을 지원하는 방법을 검사합니다.
- 소프트웨어 에이전트의 진화는 자율성 및 동시성의 초기 개념부터 다중 에이전트 시스템 및 공식 에이전트 아키텍처의 출현에 이르기까지 소프트웨어 에이전트를 형성한 지적 및 기술적 이정표를 추적하여 생성형 AI와 수렴합니다.
- 에이전트 AI에 대한 소프트웨어 에이전트는 분산 에이전트 모델을 파운데이션 모델, 서버리스 컴퓨팅 및 오케스트레이션 프로토콜과 결합하는 수십 년의 발전의 정점으로 에이전트 AI를 도입합니다. 이 섹션에서는 이러한 수렴을 통해 자율성, 비동기성 및 실제 기관 규모로 작동하는 새로운 세대의 지능형 도구 사용 에이전트를 지원하는 방법을 설명합니다.

수강 대상

이 가이드는 최신 클라우드 솔루션을 위해이 기술을 채택하기 전에 소프트웨어 에이전트의 역사, 주요 개념 및 에이전트 AI로의 진화를 이해하려는 아키텍트, 개발자 및 기술 리더를 위해 설계되었습니다 AWS.

목표

에이전트 아키텍처를 채택하면 조직이 다음과 같은 이점을 얻을 수 있습니다.

- 가치 실현 시간 단축: 지식 작업을 자동화 및 확장하고 수동 작업 및 지연 시간을 줄입니다.
- 고객 참여 개선: 도메인 간에 지능형 어시스턴트를 제공합니다.
- 운영 비용 절감: 이전에 사람의 의견이나 감독이 필요했던 의사 결정 흐름을 자동화합니다.
- 혁신 및 차별화 추진: 실시간으로 적응하고, 배우고, 경쟁하는 지능형 제품을 구축합니다.
- 레거시 워크플로 현대화: 스크립트와 모놀리스를 모듈식 추론 에이전트로 재구성합니다.

이 콘텐츠 시리즈 정보

이 가이드는 AI 기반 소프트웨어 에이전트를 구축하기 위한 아키텍처 청사진과 기술 지침을 제공하는 간행물 세트의 일부입니다 AWS. 시리즈에는 다음이 포함됩니다.

- 에서 에이전트 AI 운영 AWS
- 의 에이전트 AI 기반 AWS(이 가이드)
- 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

이 콘텐츠 시리즈에 대한 자세한 내용은 Agentic AI를 참조하세요.

소프트웨어 에이전트 소개

소프트웨어 에이전트의 개념은 1960년대 자율 엔터티의 기반에서 1990년대 초 공식 탐색으로 크게 발전했습니다. 결정론적 스크립트에서 적응형 지능형 애플리케이션에 이르기까지 디지털 시스템이 점점 더 복잡해짐에 따라 소프트웨어 에이전트는 컴퓨팅 시스템에서 자율적이고 컨텍스트를 인식하며 목표 기반 동작을 활성화하기 위한 필수 구성 요소가 되었습니다. 클라우드 네이티브 및 AI 강화 아키텍처의 맥락에서, 특히 생성형 AI, 대규모 언어 모델(LLMs) 및 Amazon Bedrock과 같은 플랫폼이 등장함에 따라 소프트웨어 에이전트는 새로운 기능 및 규모 렌즈를 통해 재정의되고 있습니다.

이 소개는 Hyacinth S. Nwana(Nwana 1996)의 중요 작업 소프트웨어 에이전트: 개요에서 발췌한 것입니다. 소프트웨어 에이전트를 정의하고, 개념적 루트를 논의하며, 토론을 현대적 프레임워크로 확장하여 최신 소프트웨어 에이전트의 세 가지 중요한 원칙인 자율성, 비동기성 및 기관을 정의합니다. 이러한 원칙은 소프트웨어 에이전트를 다른 유형의 서비스 또는 애플리케이션과 구분하며, 이러한 에이전트가 분산된 실시간 환경에서 목적, 복원력 및 인텔리전스를 바탕으로 작동할 수 있도록 합니다.

이 섹션의 내용

자율성에서 분산 인텔리전스로

소프트웨어 에이전트라는 용어가 주류에 진입하기 전에 초기 컴퓨팅 연구를 통해 독립적으로 행동하고, 입력에 반응하고, 내부 규칙 또는 목표에 따라 결정을 내릴 수 있는 시스템인 자율 디지털 엔터티에 대한 아이디어를 탐색했습니다. 이러한 초기 아이디어는 에이전트 패러다임이 될 수 있는 개념적 토대를 마련했습니다. (과거 타임라인은이 가이드 뒷부분의 소프트웨어 에이전트 진화 섹션을 참조하세요.)

자율성의 초기 개념

인간 운영자와 독립적으로 작동하는 기계 또는 프로그램의 개념은 수십 년 동안 시스템 디자이너에게 흥미를 불러일으켰습니다. 사이버넷, 인공 지능 및 제어 시스템의 초기 작업은 소프트웨어가 자체 규제 동작을 보이고, 변화에 동적으로 대응하고, 지속적인 사람의 감독 없이 운영할 수 있는 방법을 조사했습니다.

이러한 아이디어는 지능형 시스템의 핵심 속성으로 자율성을 도입하고 대응 또는 실행만 하는 대신 결정하고 조치를 취할 수 있는 소프트웨어 출현의 단계를 설정했습니다.

액터 모델 및 비동기 실행

1970년대에 인공 지능용 범용 모듈식 ACTOR 격식(Hewitt et al. 1973) 논문에 도입된 액터 모델은 분산된 메시지 기반 계산을 고려하기 위한 공식 프레임워크를 제공했습니다. 이 모델에서 액터는 비동기식 메시지를 전달하여 독점적으로 통신하고 확장 가능하고 동시적이며 내결함성 시스템을 활성화하는 독립 엔터티입니다.

액터 모델은 현대 에이전트 설계에 계속 영향을 미치는 세 가지 주요 속성을 강조했습니다.

- 상태 및 동작 격리
- 개체 간의 비동기 상호 작용
- 작업의 동적 생성 및 위임

이러한 속성은 분산 시스템의 요구 사항에 부합하며 클라우드 네이티브 환경에서 소프트웨어 에이전트의 운영 특성을 미리 구성했습니다.

분산형 인텔리전스 및 다중 에이전트 시스템

1960년대 이후 컴퓨팅 시스템이 더 상호 연결됨에 따라 연구원은 분산형 인공 지능(DAI)을 탐색했습니다. 이 필드는 여러 자율 개체가 시스템 전체에서 협업하거나 경쟁적으로 작동하는 방식에 중점을 두었습니다. DAI는 각 에이전트가 로컬 목표, 지각 및 추론을 가지고 있지만 더 광범위하고 상호 연결된 환경에서도 운영되는 다중 에이전트 시스템을 개발했습니다.

의사 결정이 분산되고 에이전트 상호 작용으로 인해 새로운 동작이 발생하는 분산 인텔리전스에 대한 이러한 비전은 최신 에이전트 기반 시스템을 구상하고 구축하는 방법의 핵심으로 남아 있습니다.

Nwana의 유형과 소프트웨어 에이전트의 부상

1990년대 중반 소프트웨어 에이전트 개념의 공식화는 지능형 시스템의 진화에서 전환점을 보여주었습니다. 이 공식화에 대한 가장 영향력 있는 기여 중 하나는 Hyacinth S. Nwana의 논문인 Software Agents: An Overview (Nwana 1996)입니다.이 논문은 다양한 차원에서 소프트웨어 에이전트를 분류하고 이해하기 위한 최초의 포괄적인 프레임워크 중 하나를 제공했습니다.

이 백서에서 Nwana는 소프트웨어 에이전트 연구 상태를 설문 조사하고 에이전트가 정의되고 구현되는 방식에서 점점 더 다양해지는 것을 식별합니다. 이 백서는 일반적인 개념 프레임워크의 필요성을 강조하고 주요 기능에 따라 에이전트를 분류하는 유형을 제안합니다. 학계 및 산업의 대표적인 에이전트 시스템을 검토하고, 에이전트를 기존 프로그램 및 객체와 구분하며, 에이전트 기반 컴퓨팅의 과제와 기회를 간략하게 설명합니다.

Nwana는 소프트웨어 에이전트가 모놀리식 개념이 아니라 정교함과 기능의 스펙트럼을 따라 존재한다는 점을 강조합니다. 유형은 이러한 환경을 명확히 하고 향후 설계 및 연구를 안내하는 역할을 합니다.

Nwana는 소프트웨어 에이전트를 특정 환경에서 지속적으로 그리고 자율적으로 작동하는 소프트웨어 엔터티로 정의합니다. 이러한 엔터티는 종종 다른 에이전트와 프로세스에서 서식합니다. 이 정의는 두 가지 주요 특성을 강조합니다.

- 연속성: 에이전트는 지속적인 사람의 개입 없이 시간이 지남에 따라 지속적으로 작동합니다.
- 자율성: 에이전트는 환경에 대한 인식에 따라 독립적으로 결정을 내리고 조치를 취할 수 있습니다.

이 정의는 Nwana의 에이전트 유형과 결합되어 에이전트의 기본 특성으로서 위임된 권한(자율성을 통해)과 적극성을 강조합니다. 직접 명령에만 응답하는 대신 다른 엔터티를 대신하여 독립적으로 행동하고 목표를 추구하는 동작을 시작하는 에이전트의 능력을 강조하여 에이전트와 하위 루틴 또는 서비스를 구분합니다.

Nwana의 에이전트 유형

다양한 유형의 에이전트를 추가로 구분하기 위해 Nwana는 6가지 주요 속성을 기반으로 분류 시스템을 도입합니다.

- 자율성: 에이전트는 사람 또는 다른 사람의 직접적인 개입 없이 작동합니다.
- 소셜 기능: 에이전트는 통신 메커니즘을 사용하여 다른 에이전트 또는 사람과 상호 작용합니다.
- 반응성: 에이전트가 환경을 인식하고 적시에 응답합니다.
- 적극성: 에이전트가 이니셔티브를 통해 목표 지향적인 행동을 보입니다.
- 적응성 및 학습: 에이전트는 경험을 통해 시간이 지남에 따라 성능을 개선합니다.
- 이동성: 에이전트가 다양한 시스템 환경 또는 네트워크를 이동할 수 있습니다.

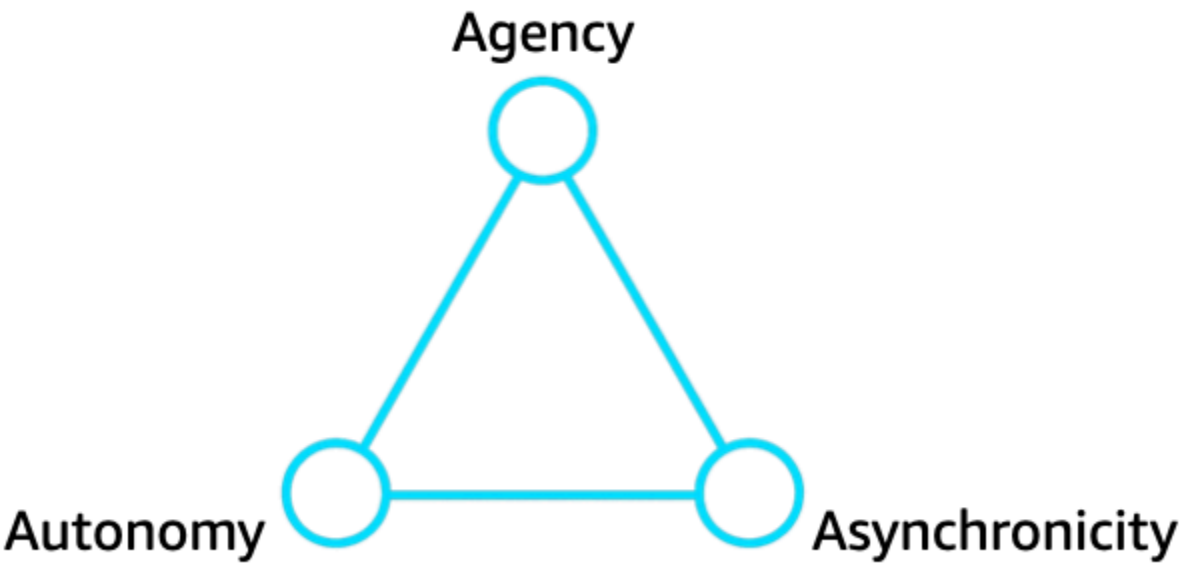
유형에서 최신 에이전트 원칙으로

Nwana의 작업은 분류 체계이자 컴퓨팅 커뮤니티가 소프트웨어에서 진화하는 형태의 기관을 평가할 수 있는 기본 렌즈 역할을 했습니다. 자율성, 능동성, 사용자 또는 시스템을 대신하여 행동하는 개념에 중점을 둔 그는 이제 에이전트적 행동을 고려하는 토대를 마련했습니다.

특히 생성형 AI, 서버리스 인프라 및 다중 에이전트 오케스트레이션 프레임워크가 증가함에 따라 기술과 환경이 변경되었지만 Nwana 작업의 기본 인사이트는 여전히 관련성이 있습니다. 초기 에이전트 이론과 소프트웨어 에이전트의 세 가지 최신 원칙 사이에 중요한 연결 고리를 제공합니다.

최신 소프트웨어 에이전트의 세 가지 원칙

오늘날의 AI 기반 플랫폼, 마이크로서비스 아키텍처 및 이벤트 기반 시스템의 맥락에서 소프트웨어 에이전트는 표준 서비스 또는 자동화 스크립트와 구별되는 세 가지 상호 의존적 원칙, 즉 자율성, 비동기성 및 기관으로 정의할 수 있습니다. 다음 그림과 후속 다이어그램에서 삼각형은 현대 소프트웨어 에이전트의 세 가지 원칙을 나타냅니다.



자율성

최신 에이전트는 독립적으로 작동합니다. 인적 프롬프트 없이 내부 상태 및 환경 컨텍스트를 기반으로 결정을 내립니다. 이를 통해 실시간으로 데이터에 대응하고, 자체 수명 주기를 관리하고, 목표 및 상황 입력에 따라 동작을 조정할 수 있습니다.

자율성은 에이전트 행동의 기반입니다. 이를 통해 에이전트는 지속적인 감독이나 하드코딩된 제어 흐름 없이 작동할 수 있습니다.

비동기성

에이전트는 기본적으로 비동기식입니다. 즉, 호출 차단이나 선형 워크플로에 의존하지 않고 이벤트, 신호 및 자극에 반응합니다. 이 특성을 통해 확장 가능한 비차단 통신, 분산 환경의 응답성, 구성 요소 간의 느슨한 결합이 가능합니다.

에이전트는 비동기성을 통해 실시간 시스템에 참여하고 다른 서비스 또는 에이전트와 유연하고 효율적으로 조정할 수 있습니다.

정의 원칙으로서의 기관

자율성과 비동기성이 필요하지만 이러한 기능만으로는 시스템을 실제 소프트웨어 에이전트로 만들기에 충분하지 않습니다. 중요한 차별화 요소는 다음을 도입하는 기관입니다.

- 목표 지향적 행동: 에이전트는 목표를 추구하고 목표에 대한 진행 상황을 평가합니다.
- 의사 결정: 에이전트는 옵션을 평가하고 규칙, 모델 또는 학습된 정책에 따라 작업을 선택합니다.
- 위임된 의도: 에이전트는 사람, 시스템 또는 조직을 대신하여 행동하며 목적 의식이 내재되어 있습니다.
- 컨텍스트 추론: 에이전트는 환경의 메모리 또는 모델을 통합하여 행동을 지능적으로 안내합니다.

자율적이고 비동기적인 시스템은 여전히 사후 대응 서비스일 수 있습니다. 소프트웨어 에이전트가 되는 이유는 의도와 목적을 가지고 에이전트가 되기 위해 행동하는 능력입니다.

용도가 있는 기관

자율성, 비동기성 및 기관 원칙을 통해 시스템은 분산 환경에서 지능적이고 적응적이며 독립적으로 작동할 수 있습니다. 이러한 원칙은 수십 년간의 개념 및 아키텍처 진화에 기반을 두고 있으며, 이제 오늘날 구축되고 있는 대부분의 최첨단 AI 시스템을 뒷받침합니다.

생성형 AI, 목표 지향 오케스트레이션 및 다중 에이전트 협업의 새로운 시대에서는 소프트웨어 에이전트가 진정으로 에이전트가 되는 이유를 이해하는 것이 중요합니다. 기관을 정의 특성으로 인식하면 자동화를 넘어 목적이 있는 자율 지능 영역으로 전환하는 데 도움이 됩니다.

소프트웨어 에이전트의 목적

최신 시스템이 점점 더 복잡해지고, 분산되고, 지능화됨에 따라 소프트웨어 에이전트의 역할은 자율 운영부터 사용자 지원 기술에 이르기까지 다양한 도메인에서 두드러졌습니다. 하지만 소프트웨어 에이전트의 기본 목적은 무엇인가요? 스크립트, 서비스 또는 정적 모델을 넘어서는 시스템을 설계하는 대신 인식, 추론 및 조치를 수행할 수 있는 엔터티에 작업을 위임하는 이유는 무엇입니까?

이 섹션에서는 자율성, 적응성 및 의도적인 작업에 중점을 두고 동적 환경 내에서 작업을 지능적으로 위임할 수 있도록 하는 소프트웨어 에이전트의 기본 목적을 살펴봅니다. 소프트웨어 에이전트의 개념적 토대를 소개하고, 인지 구조를 추적하며, 해결할 수 있는 고유한 장비를 갖춘 실제 문제를 간략하게 설명합니다.

이 섹션의 내용

액터 모델에서 에이전트 인식으로

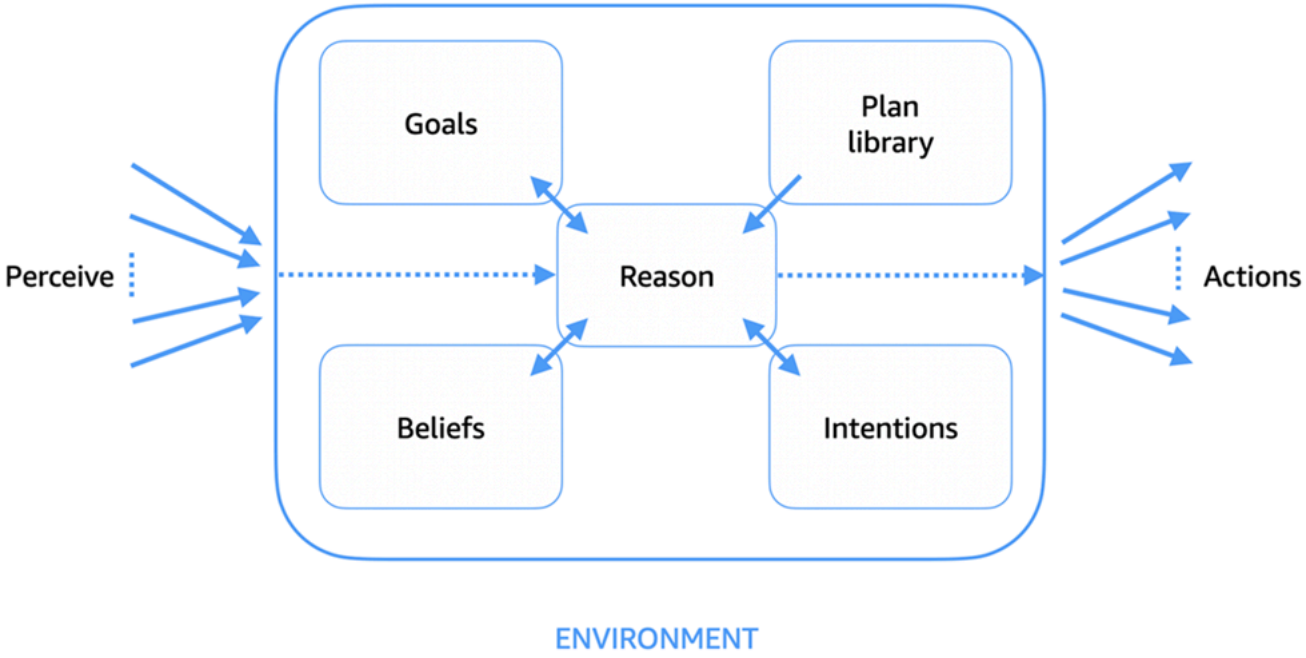
소프트웨어 에이전트의 목적과 구조는 초기 계산 모델, 특히 1970년대에 Carl Hewitt가 도입한 액터 모델에서 나온 아이디어에 기반을 두고 있습니다(Hewitt 외 1973).

액터 모델은 계산을 액터라고 하는 독립적이고 동시에 실행되는 개체의 모음으로 취급합니다. 각 액터는 자체 상태를 캡슐화하고, 비동기 메시지 전달을 통해서만 상호 작용하며, 새 액터를 생성하고 작업을 위임할 수 있습니다.

이 모델은 분산 추론, 대응 및 격리를 위한 개념적 기반을 제공했으며,이 모든 것이 최신 소프트웨어 에이전트의 행동 아키텍처를 뒷받침합니다.

에이전트 함수: 인지, 이유, 행동

모든 소프트웨어 에이전트의 핵심은 인식, 이유, 행동 루프로 설명되는 인지 주기입니다. 이 프로세스는 다음 다이어그램에 설명되어 있습니다. 동적 환경에서 에이전트가 자율적으로 작동하는 방식을 정의합니다.



- 지각: 에이전트는 환경에서 정보(예: 이벤트, 센서 입력 또는 API 신호)를 수집하고 내부 상태 또는 신념을 업데이트합니다.
- 이유: 에이전트는 계획 라이브러리 또는 로직 시스템을 사용하여 현재 신념, 목표 및 컨텍스트 지식을 분석합니다. 이 프로세스에는 목표 우선순위 지정, 충돌 해결 또는 의도 선택이 포함될 수 있습니다.
- 조치: 에이전트는 위임된 목표 달성에 더 가까워지는 조치를 선택하고 실행합니다.

이 아키텍처는 에이전트가 엄격한 프로그래밍을 넘어 작동할 수 있는 기능을 지원하며 유연하고 상황에 민감하며 목표 지향적인 동작을 지원합니다. 소프트웨어 에이전트의 더 광범위한 목적을 안내하는 멘탈 프레임워크를 형성합니다.

자율 협업 및 의도성

소프트웨어 에이전트의 목적은 최신 컴퓨팅에 자율성, 컨텍스트 인식 및 지능형 위임을 제공하는 것입니다. 에이전트는 액터 모델의 원칙을 기반으로 구축되고 인식, 이유, 행동 주기에 구현되므로 사후 대응적일 뿐만 아니라 선제적이고 목적 지향적인 시스템을 활성화합니다.

에이전트는 소프트웨어가 복잡한 환경에서 결정, 적응 및 조치를 취할 수 있도록 지원합니다. 사용자를 대표하고, 목표를 해석하고, 기계 속도로 작업을 구현합니다. 에이전트 AI의 시대로 깊이 들어가면서 소프트웨어 에이전트는 인간의 의도와 지능형 디지털 행동 간의 운영 인터페이스가 되고 있습니다.

의도 위임

기존 소프트웨어 구성 요소와 달리 소프트웨어 에이전트는 사용자, 다른 시스템 또는 상위 수준 서비스 등 다른 것을 대신하여 작업합니다. 위임된 의도를 가지고 있습니다. 즉, 다음과 같습니다.

- 시작 후 독립적으로 작동합니다.
- 위임자의 목표에 맞게 선택합니다.
- 실행 시 불확실성과 장단점을 탐색합니다.

에이전트는 지침과 결과 간의 격차를 해소하므로 사용자는 명시적 지침이 필요하지 않고 더 높은 수준의 추상화에서 의도를 표현할 수 있습니다.

예측할 수 없는 동적 환경에서 운영

소프트웨어 에이전트는 조건이 지속적으로 변경되고 데이터가 실시간으로 도착하며 제어 및 컨텍스트가 분산되는 환경을 위해 설계되었습니다.

정확한 입력 또는 동기식 실행이 필요한 정적 프로그램과 달리 에이전트는 주변 환경에 적응하고 동적으로 응답합니다. 이는 클라우드 네이티브 인프라, 엣지 컴퓨팅, 사물 인터넷(IoT) 네트워크 및 실시간 의사 결정 시스템에서 중요한 기능입니다.

인간의 인지 부하 줄이기

소프트웨어 에이전트의 주요 목적 중 하나는 인간에 대한 인지 및 운영 부담을 줄이는 것입니다. 에이전트는 다음을 수행할 수 있습니다.

- 시스템 및 워크플로를 지속적으로 모니터링합니다.
- 사전 정의되거나 긴급한 상태를 감지하고 이에 대응합니다.
- 반복적이고 대량의 결정을 자동화합니다.
- 지연 시간을 최소화하면서 환경 변화에 대응합니다.

의사 결정이 사용자에서 에이전트로 전환되면 시스템은 더 응답적이고 복원력이 뛰어나며 인간 중심적이 되고 새로운 정보 또는 중단에 실시간으로 적응할 수 있습니다. 이렇게 하면 복잡성이 높거나 규모가 큰 환경에서 반응 처리 속도를 높이고 운영 연속성을 높일 수 있습니다. 그 결과 마이크로 수준의 의사 결정에서 전략적 감독 및 창의적인 문제 해결로 인적 초점이 전환됩니다.

분산 인텔리전스 활성화

소프트웨어 에이전트가 개별적으로 또는 집합적으로 운영할 수 있으므로 환경 또는 조직 간에 조정되는 다중 에이전트 시스템(MAS)을 설계할 수 있습니다. 이러한 시스템은 작업을 지능적으로 분산하고 복합 목표를 향해 협상, 협력 또는 경쟁할 수 있습니다.

예를 들어 글로벌 공급망 시스템에서는 개별 에이전트가 공장, 배송, 창고 및 라스트 마일 배송을 관리합니다. 각 에이전트는 로컬 자율성으로 운영됩니다. 팩토리 에이전트는 리소스 제약 조건에 따라 프로덕션을 최적화하고, 창고 에이전트는 인벤토리 흐름을 실시간으로 조정하며, 배송 에이전트는 트래픽 및 고객 가용성에 따라 배송 경로를 변경합니다.

이러한 에이전트는 동적으로 통신하고 조정하며 중앙 집중식 제어 없이 포트 지연 또는 트럭 고장과 같은 중단에 적응합니다. 시스템의 전체 인텔리전스는 이러한 상호 작용에서 비롯되며 단일 구성 요소의 기능을 넘어서는 복원력이 뛰어나고 최적화된 물류를 가능하게 합니다.

이 모델에서 에이전트는 더 광범위한 인텔리전스 패브릭에서 노드 역할을 합니다. 이는 단일 구성 요소만 처리할 수 없는 문제를 해결할 수 있는 새로운 시스템을 형성합니다.

반응뿐만 아니라 목적을 가지고 행동하기

복잡한 시스템에서는 자동화만으로는 충분하지 않습니다. 소프트웨어 에이전트의 정의 목적은 목적을 가지고 행동하고, 목표를 평가하고, 컨텍스트를 평가하고, 정보에 입각한 선택을 내리는 것입니다. 즉, 소프트웨어 에이전트는 트리거에만 응답하는 대신 목표를 추구합니다. 경험이나 피드백을 기반으로 신념과 의도를 수정할 수 있습니다. 이 컨텍스트에서 신념은 에이전트의 지각(입력 및 센서)을 기반으로 환경에 대한 에이전트의 내부 표현(예: “패키지 X는 창고 A에 있음”)을 나타냅니다. 의도는 에이전트가 목표를 달성하기 위해 선택한 계획을 나타냅니다(예: "전달 경로 B를 사용하고 수신자에게 알림"). 에이전트는 필요에 따라 작업을 에스컬레이션, 연기 또는 조정할 수도 있습니다.

이러한 의도로 인해 소프트웨어 에이전트는 사후 대응 실행기뿐만 아니라 지능형 시스템의 자율 공동 작업자가 됩니다.

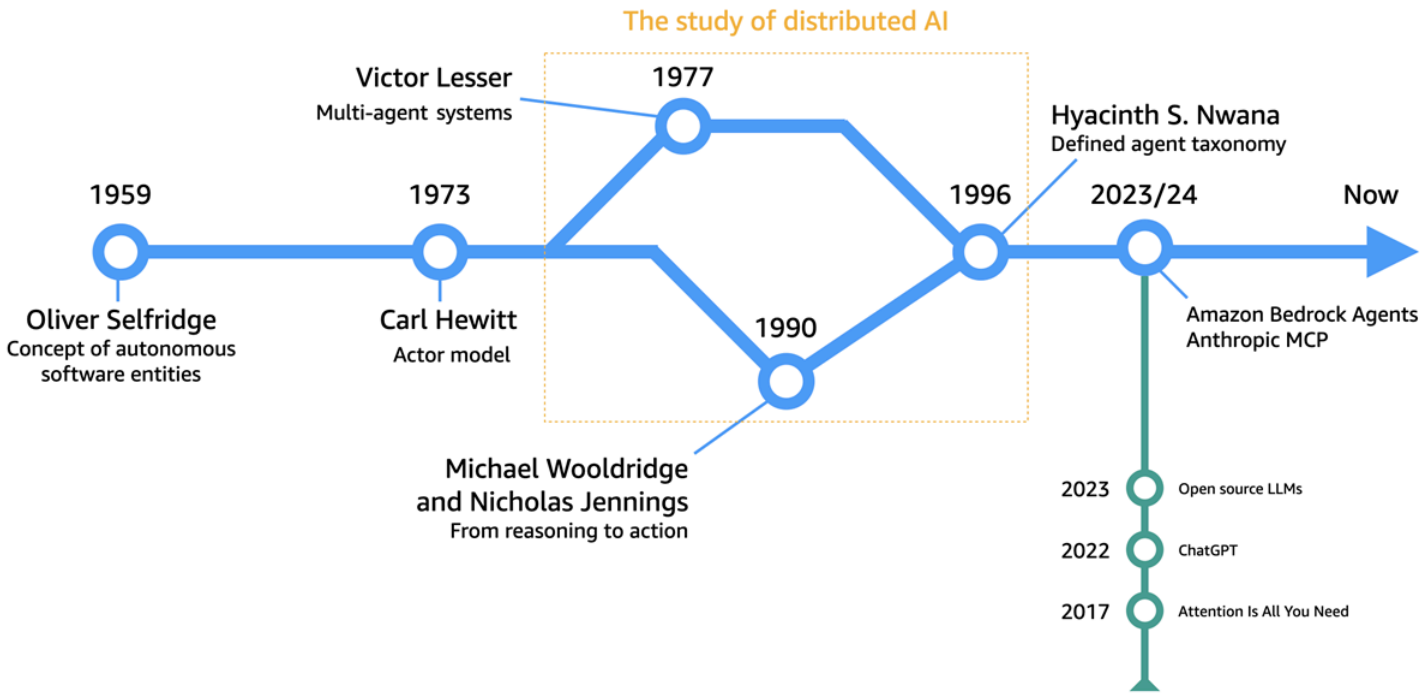
소프트웨어 에이전트의 진화

간단한 자동화 시스템에서 지능형, 자율 및 목표 지향 소프트웨어 에이전트로의 여정은 컴퓨터 과학, 인공 지능 및 분산 시스템의 수십 년 진화를 반영합니다.

그런 다음 기계 학습이 증가하여 패러다임이 수제 규칙에서 통계 패턴 인식으로 바뀌었습니다. 이러한 시스템은 데이터에서 학습할 수 있으며 지각, 분류 및 의사 결정의 발전을 가능하게 했습니다.

대규모 언어 모델(LLMs)은 규모, 아키텍처 및 비지도 학습의 수렴을 나타냅니다. LLMs 작업별 훈련을 거의 또는 전혀 하지 않고 작업을 추론, 생성 및 조정할 수 있습니다. LLMs 확장 가능한 클라우드 네이티브 인프라 및 구성 가능한 아키텍처와 결합하면 이제 엔터프라이즈 규모에서 자율성, 컨텍스트 인식 및 적응성을 갖춘 지능형 소프트웨어 에이전트인 에이전트 AI의 전체 비전을 달성할 수 있습니다.

이 섹션에서는 다음 다이어그램과 같이 기본 이론부터 최신 관행까지 소프트웨어 에이전트의 기록을 살펴봅니다. 분산형 인공 지능(DAI)과 변환기 기반 생성형 AI의 수렴을 강조하고 에이전트 AI의 출현을 형성한 주요 마일스톤을 식별합니다.



이 섹션의 내용

- 소프트웨어 에이전트의 기반
- 필드 성숙: 추론에서 행동으로
- 병렬 타임라인: 대규모 언어 모델의 부상

- 타임라인 수렴: 에이전트 AI의 등장

소프트웨어 에이전트의 기반

1959년 - 올리버 셀프리지: 소프트웨어 자율성의 탄생

소프트웨어 에이전트의 루트는 환경을 인식하고 독립적으로 행동할 수 있는 자율 소프트웨어 엔터티(데몬)의 개념을 도입한 올리버 셀프리지(Selfridge 1959)로 거슬러 올라갑니다. 기계 지각 및 학습에 대한 초기 작업은 독립적이고 지능적인 시스템으로 에이전트의 미래 개념에 대한 철학적 토대를 마련했습니다.

1973년 - Carl Hewitt: 액터 모델

핵심 발전은 에이전트를 독립적인 동시 엔터티로 설명하는 공식 계산 모델인 Carl Hewitt의 액터 모델(Hewitt et al. 1973)과 함께 이루어졌습니다. 이 모델에서 에이전트는 자신의 상태 및 동작을 캡슐화하고, 비동기식 메시지 전달을 사용하여 통신하고, 다른 액터를 동적으로 생성하고, 작업을 위임할 수 있습니다.

액터 모델은 분산된 에이전트 기반 시스템에 대한 이론적 기초와 아키텍처 패러다임을 모두 제공했습니다. 이 모델은 Erlang 프로그래밍 언어 및 Akka 프레임워크와 같은 최신 동시성 구현을 사전 설계했습니다.

필드 성숙: 추론에서 행동으로

1977 - Victor Lesser: 다중 에이전트 시스템

1970년대 후반에 분산형 인공 지능(DAI)이 등장했습니다. 이는 다중 에이전트 시스템(MAS)을 혁신하는 것으로 널리 인정받는 Victor Lesser의 지지를 받았습니다. 그의 작업은 독립 소프트웨어 엔터티가 협력, 조정 및 협상하는 방법에 중점을 두었습니다(리소스 섹션 참조). 이러한 개발로 인해 복잡한 문제를 집합적으로 해결할 수 있었던 시스템은 분산 인텔리전스를 구축하는 데 필수적인 도약이었습니다.

1990년대 - Michael™dridge 및 Nicholas Jennings: 에이전트 스펙트럼

1990년대까지 분산 인텔리전스 분야는 Michael™dridge 및 Nicholas Jennings와 같은 연구자의 기여로 성숙해졌습니다. 이러한 학자들은 비인지 시스템에서 목표 중심의 추론 에이전트(Wooldridge and Jennings 1995)에 이르기까지 반응형부터 고의형까지 스펙트럼을 따라 에이전트를 분류했습니다. 이

들의 작업은 에이전트가 더 이상 추상적인 아이디어가 아니라 로봇 공학부터 엔터프라이즈 소프트웨어에 이르기까지 다양한 실용적인 도메인에 적용되고 있음을 강조했습니다.

또한 이러한 연구자는 중앙 집중식 추론에서 분산 작업으로의 전환에 중점을 두었습니다. 에이전트는 더 이상 단순한 사상가가 아니라 자율성과 목적을 갖춘 실시간 환경에서 운영되는 사람이었습니다.

1996 - Hyacinth S. Nwana: 에이전트 개념 공식화

1996년, Hyacinth S. Nwana는 지금까지 가장 포괄적인 에이전트 분류를 제공한 영향력 있는 소프트웨어 에이전트: 개요를 발표했습니다. 그의 유형에는 자율성, 사회적 능력, 대응성, 능동성, 학습 및 이동성과 같은 속성이 포함되었으며 소프트웨어 에이전트와 기존 소프트웨어 구성 간에 구별되었습니다.

또한 Nwana는 이제 널리 받아들여지는 정의를 제공했습니다. 즉, 소프트웨어 에이전트는 위임 개념에서 파생된 기관 관계의 사용자 또는 기타 프로그램을 위해 작동하는 소프트웨어 기반 컴퓨터 프로그램입니다.

이 공식화는 소프트웨어 에이전트를 이론적 구문에서 실제 애플리케이션으로 전환하는 데 중요한 역할을 했습니다. 이를 통해 통신, 워크플로 자동화, 지능형 어시스턴트와 같은 여러 분야에서 에이전트 기반 시스템이 생성되었습니다.

Nwana의 작업은 초기 분산 AI 연구와 현대 에이전트의 운영 아키텍처의 수렴 지점에 있습니다. 이는 에이전트의 인지 이론과 오늘날 시스템에서의 실제 배포 간의 중요한 연결고리입니다.

병렬 타임라인: 대규모 언어 모델의 부상

에이전트 프레임워크가 발전하는 동안 자연어 처리 및 기계 학습에서 병렬적이고 수렴적인 혁신이 이루어졌습니다.

- 2017 – 변환기: 필요한 모든 것에 주의를 기울이는 백서(Vaswani et al. 2017)는 변환기 아키텍처를 도입하여 기계가 언어를 처리하고 생성하는 방식을 크게 개선했습니다.
- 2022년 - ChatGPT: OpenAI는 ChatGPT라는 GPT-3.5에 대한 채팅 기반 인터페이스를 출시하여 범용 AI 시스템과의 자연스러운 대화형 대화를 가능하게 했습니다.
- 2023 - 오픈 소스 LLMs: Llama, Falcon 및 Mistral의 릴리스는 강력한 모델에 널리 액세스하고 오픈 소스 및 엔터프라이즈 환경에서 에이전트 프레임워크 개발을 가속화했습니다.

이러한 혁신을 통해 언어 모델은 컨텍스트를 구문 분석하고, 작업을 계획하고, 응답을 연결할 수 있는 추론 엔진으로 변모했으며, LLMs 지능형 소프트웨어 에이전트의 주요 조력자가 되었습니다.

타임라인 수렴: 에이전트 AI의 등장

2023-2024 - 엔터프라이즈급 에이전트 플랫폼

분산 소프트웨어 에이전트 아키텍처와 변환기 기반 LLMs의 수렴은 에이전트 AI의 부상에 절정입니다.

- Amazon Bedrock Agents는 Amazon Bedrock의 파운데이션 모델을 사용하여 목표 기반 도구 사용 소프트웨어 에이전트를 구축하는 완전 관리형 방법을 도입했습니다.
- Anthropic의 모델 컨텍스트 프로토콜(MCP)은 대규모 언어 모델이 외부 도구, 환경 및 메모리에 액세스하고 상호 작용할 수 있는 방법을 정의했습니다. 이는 상황별, 지속적 및 자율 동작의 핵심입니다.

이 두 마일스톤은 기관과 인텔리전스의 합성을 나타냅니다. 에이전트는 더 이상 정적 워크플로 또는 엄격한 자동화로 제한되지 않았습니다. 이제 여러 단계에서 추론하고, 도구 및 APIs와 조정하고, 컨텍스트 상태를 유지하고, 시간이 지남에 따라 학습하고 적응할 수 있습니다.

2025년 1월~6월 - 엔터프라이즈 기능 확장

2025년 상반기에 에이전트 AI 환경은 새로운 엔터프라이즈 기능으로 크게 확장되었습니다. 2025년 2월 Anthropic은 시장 최초의 하이브리드 추론 모델인 Claude 3.7 Sonnet을 출시했으며, MCP 사양은 널리 채택되었습니다.

Amazon Q Developer, Cursor 및 WindSurf와 같은 AI 코딩 어시스턴트는 코드 생성, 리포지토리 분석 및 개발 워크플로를 표준화하기 위해 MCP를 통합했습니다. MCP 2025년 3월 릴리스에는 OAuth 2.1 보안 통합, 다양한 데이터 액세스를 위한 확장된 리소스 유형, 스트리밍 가능한 HTTP를 통한 향상된 연결 옵션 등 중요한 엔터프라이즈 지원 기능이 도입되었습니다. 이 기반을 기반으로 AWS 2025년 5월에 MCP 운영 위원회에 가입하여 새로운 agent-to-agent 통신 기능에 기여한다고 발표했습니다. 이렇게 하면 에이전트 AI 상호 운용성을 위한 업계 표준으로서 프로토콜의 입지가 더욱 강화됩니다.

2025년 5월에 Strands Agents 프레임워크를 오픈 소싱하여 에이전트 AI 워크플로를 구축하기 위한 고객 옵션을 AWS 강화했습니다. https://strandsagents.com/latest/ 이 공급자와 무관하고 모델에 구애받지 않는 프레임워크를 통해 개발자는 심층적인 AWS 서비스 통합을 유지하면서 플랫폼 전체에서 파운데이션 모델을 사용할 수 있습니다. AWS 오픈 소스 블로그에서 강조 표시된 것처럼 Strands Agents는 파운데이션 모델을 에이전트 인텔리전스의 핵심에 배치하는 모델 우선 설계 철학을 따릅니다. 이를 통해 고객은 특정 사용 사례에 맞게 정교한 AI 에이전트를 더 쉽게 구축하고 배포할 수 있습니다.

출현 - 에이전트 AI

초기 자율성 아이디어에서 최신 LLM 지원 오케스트레이션에 이르기까지 소프트웨어 에이전트의 진화는 오래되고 계층화되어 왔습니다. 프로그램 인식에 대한 올리버 셀프리지의 비전으로 시작된 것은 협

업, 적응 및 추론이 가능한 지능형, 컨텍스트 인식, 목표 기반 소프트웨어 에이전트로 구성된 강력한 에코시스템으로 성장했습니다.

분산형 인공 지능(DAI)과 변환기 기반 생성형 AI의 수렴은 소프트웨어 에이전트가 더 이상 도구일 뿐만 아니라 지능형 시스템의 자율 액터인 새로운 시대의 시작을 나타냅니다.

에이전트 AI는 소프트웨어 시스템의 차세대 진화를 나타냅니다. 자율적이고 비동기적이며 에이전트적인 지능형 에이전트 클래스를 제공하며, 위임된 의도로 행동하고 동적 분산 환경에서 의도적으로 운영할 수 있습니다. 에이전트 AI는 다음을 통합합니다.

- 다중 에이전트 시스템 및 액터 모델의 아키텍처 계보
- 인지, 이유, 행동의 인지 모델
- LLMs 및 변환기의 생성 능력
- 클라우드 네이티브 및 서버리스 컴퓨팅의 운영 유연성

에이전트 AI에 대한 소프트웨어 에이전트

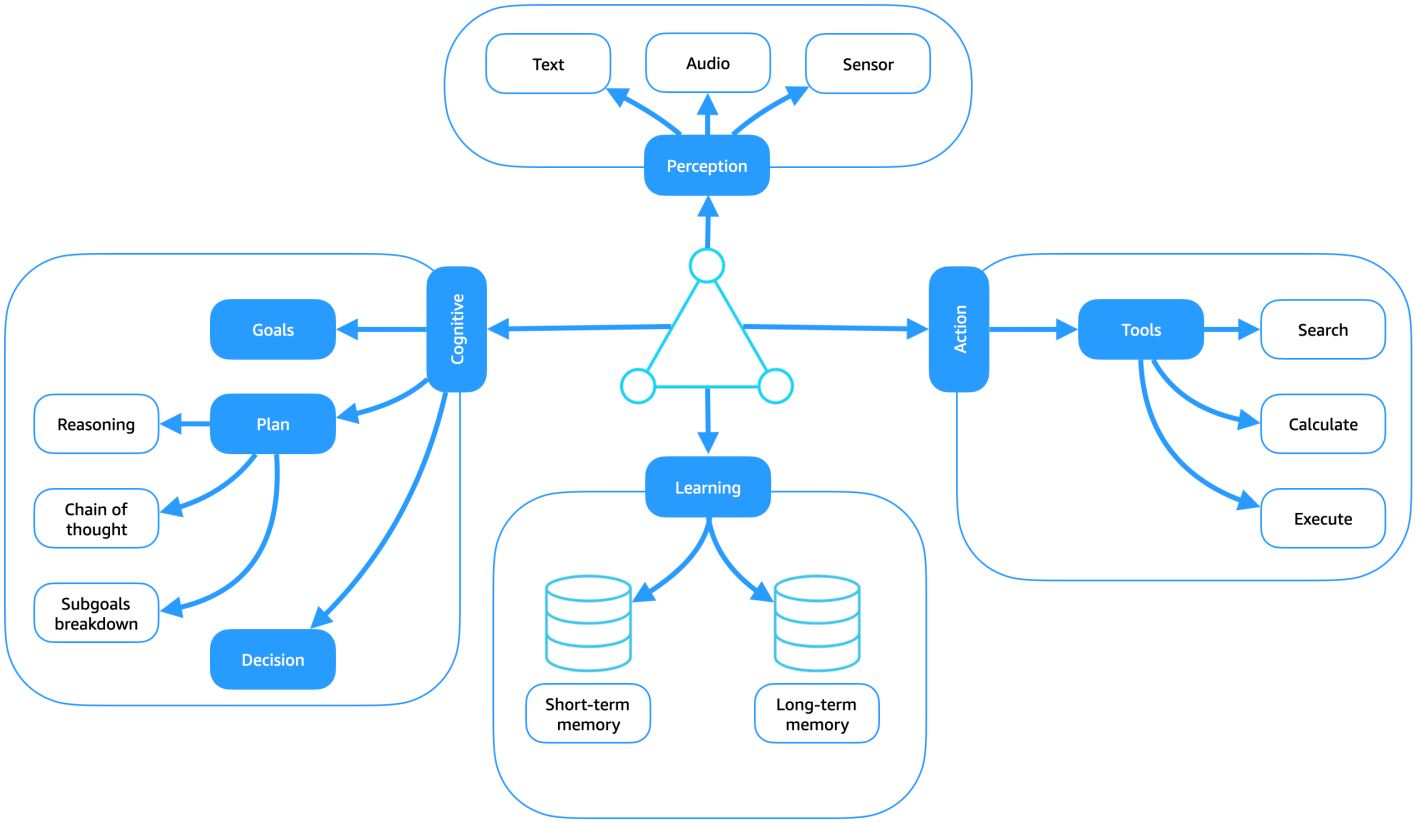
소프트웨어 에이전트는 환경을 인식하고, 목표에 대한 이유를 파악하고, 그에 따라 행동하도록 설계된 자율 디지털 엔터티입니다. 고정 로직을 따르는 기존 소프트웨어 프로그램과 달리 에이전트는 컨텍스트 입력 및 결정 프레임워크를 기반으로 행동을 조정합니다. 따라서 클라우드 네이티브 시스템, 로봇, 지능형 자동화, 현재 생성형 AI 오케스트레이션과 같은 동적 분산 환경에 이상적입니다.

이 섹션에서는 소프트웨어 에이전트의 핵심 구성 요소를 소개하고 인식, 근거, 행동 모델을 기반으로 이러한 구성 요소가 기존 아키텍처 내에서 상호 작용하는 방법을 설명합니다. 생성형 AI, 특히 대규모 언어 모델(LLMs)이 소프트웨어 에이전트의 추론과 계획 방식을 어떻게 변화시켰는지 설명합니다. 이는 규칙 기반 시스템에서 데이터 기반 학습된 에이전트 AI 인텔리전스로 근본적으로 전환된다는 것을 의미합니다.

이 섹션의 내용

소프트웨어 에이전트의 핵심 구성 요소

다음 다이어그램은 대부분의 지능형 에이전트에서 찾을 수 있는 주요 기능 모듈을 보여줍니다. 각 구성 요소는 복잡한 환경에서 에이전트가 자율적으로 운영할 수 있는 능력에 기여합니다.



인식, 이유, 행동 루프의 맥락에서 에이전트의 추론 기능은 인지 및 학습 모듈 모두에 분산됩니다. 에이전트는 메모리와 학습의 통합을 통해 과거 경험을 기반으로 적응형 추론을 개발합니다. 에이전트는 환경 내에서 행동할 때 새로운 피드백 루프를 생성합니다. 각 작업은 향후 지각에 영향을 미치며, 그 결과로 얻은 경험은 학습 모듈을 통해 메모리 및 내부 모델에 통합됩니다. 이러한 지속적인 지각, 추론 및 행동 루프를 통해 에이전트는 시간이 지남에 따라 개선되고 전체 지각, 이유, 행동 주기를 완료할 수 있습니다.

인식 모듈

지각 모듈을 사용하면 에이전트가 텍스트, 오디오 및 센서와 같은 다양한 입력 양식을 통해 환경과 인터페이스할 수 있습니다. 이러한 입력은 모든 추론 및 작업이 기반으로 하는 원시 데이터를 형성합니다. 텍스트 입력에는 자연어 프롬프트, 구조화된 명령 또는 문서가 포함될 수 있습니다. 오디오 입력에는 음성 지침 또는 환경 사운드가 포함됩니다. 센서 입력에는 시각적 피드, 모션 신호 또는 GPS 좌표와 같은 물리적 데이터가 포함됩니다. 지각의 핵심 기능은이 원시 데이터에서 의미 있는 특징과 표현을 추출하는 것입니다. 이를 통해 에이전트는 현재 컨텍스트를 정확하고 실행 가능하게 이해할 수 있습니다. 이 프로세스에는 특성 추출, 객체 또는 이벤트 인식, 의미 해석이 포함될 수 있으며 인식, 이유, 행동 루프의 중요한 첫 번째 단계를 구성합니다. 효과적인 지각은 다운스트림 추론 및 의사 결정이 관련성 있는 up-to-date 상황 인식을 기반으로 하도록 합니다.

인지 모듈

인지 모듈은 소프트웨어 에이전트의 의도적인 코어 역할을 합니다. 이는 지각을 해석하고, 의도를 형성하고, 목표 기반 계획 및 의사 결정을 통해 목적 있는 행동을 안내하는 역할을 합니다. 이 모듈은 입력을 구조화된 추론 프로세스로 변환하므로 에이전트가 사후 대응보다는 의도적으로 작동할 수 있습니다. 이러한 프로세스는 목표, 계획 및 의사 결정이라는 세 가지 주요 하위 모듈을 통해 관리됩니다.

목표 하위 모듈

목표 하위 모듈은 에이전트의 의도와 방향을 정의합니다. 목표는 명시적(예: "위치로 이동" 또는 "보고서 제출")이거나 암시적(예: "사용자 참여 극대화" 또는 "지연 시간 최소화")일 수 있습니다. 에이전트의 추론 주기의 핵심이며 계획 및 결정을 위한 대상 상태를 제공합니다.

에이전트는 목표에 대한 진행 상황을 지속적으로 평가하고 새로운 인식 또는 학습을 기반으로 목표의 우선순위를 다시 지정하거나 다시 생성할 수 있습니다. 이 목표 인식은 동적 환경에서 에이전트의 적응성을 유지합니다.

하위 모듈 계획

계획 하위 모듈은 에이전트의 현재 목표를 달성하기 위한 전략을 구성합니다. 작업 시퀀스를 생성하고, 작업을 계층적으로 분해하고, 사전 정의되거나 동적으로 생성된 계획 중에서 선택합니다.

비결정적이거나 변화하는 환경에서 효과적으로 운영하기 위해 계획은 정적이지 않습니다. 현대 에이전트는 chain-of-thought 시퀀스를 생성하고, 하위 목표를 중간 단계로 도입하고, 조건이 변경될 때 실시간으로 계획을 수정할 수 있습니다.

이 하위 모듈은 메모리 및 학습과 밀접하게 연결되며 에이전트는 과거 결과를 기반으로 시간 경과에 따른 계획을 세분화할 수 있습니다.

의사 결정 하위 모듈

의사 결정 하위 모듈은 사용 가능한 계획과 작업을 평가하여 가장 적합한 다음 단계를 선택합니다. 지각, 현재 계획, 에이전트의 목표 및 환경 컨텍스트의 입력을 통합합니다.

의사 결정 계정:

- 충돌하는 목표 간의 절충
- 신뢰도 임계값(예: 지각의 불확실성)
- 작업의 결과
- 에이전트의 학습된 경험

아키텍처에 따라 에이전트는 심볼 추론, 휴리스틱, 강화 학습 또는 언어 모델(LLMs)에 의존하여 정보에 입각한 결정을 내릴 수 있습니다. 이 프로세스를 통해 에이전트의 동작 컨텍스트 인식, 목표 정렬 및 적응을 유지할 수 있습니다.

작업 모듈

작업 모듈은 에이전트가 선택한 결정을 실행하고 외부 세계 또는 내부 시스템과 상호 작용하여 의미 있는 효과를 생성하는 역할을 합니다. 의도가 행동으로 변환되는 지각, 이유, 행동 루프의 행동 단계를 나타냅니다.

인지 모듈이 작업을 선택하면 작업 모듈은 특수 하위 모듈을 통해 실행을 조정합니다. 여기서 각 하위 모듈은 에이전트의 통합 환경에 맞게 조정됩니다.

- 물리적 작동: 로봇 시스템 또는 IoT 디바이스에 포함된 에이전트의 경우이 하위 모듈은 결정을 실제 물리적 이동 또는 하드웨어 수준 지침으로 변환합니다.

  예: 로봇 조향, 밸브 트리거, 센서 켜기.
- 통합 상호 작용:이 하위 모듈은 소프트웨어 시스템, 플랫폼 또는 APIs와의 상호 작용과 같은 비물리적이지만 외부에서 볼 수 있는 작업을 처리합니다.

  예: 클라우드 서비스에 명령 전송, 데이터베이스 업데이트, API를 호출하여 보고서 제출.
- 도구 호출: 에이전트는 특수 도구를 사용하여 다음과 같은 하위 작업을 수행하여 기능을 확장하는 경우가 많습니다.
  - 검색: 정형 또는 비정형 지식 소스 쿼리
  - 요약: 대규모 텍스트 입력을 상위 수준 개요로 압축
  - 계산: 논리적, 수치적 또는 심볼 계산 수행

  도구 호출은 모듈식의 호출 가능한 기술을 통해 복잡한 동작 구성을 가능하게 합니다.

학습 모듈

학습 모듈을 통해 에이전트는 경험을 기반으로 시간이 지남에 따라 적응, 일반화 및 개선할 수 있습니다. 인식 및 조치의 피드백을 사용하여 에이전트의 내부 모델, 전략 및 결정 정책을 지속적으로 개선하여 추론 프로세스를 지원합니다.

이 모듈은 단기 및 장기 메모리와 함께 작동합니다.

- 단기 메모리: 대화 상태, 현재 작업 정보 및 최근 관측치와 같은 일시적인 컨텍스트를 저장합니다. 에이전트가 상호 작용 및 작업 내에서 연속성을 유지하는 데 도움이 됩니다.
- 장기 메모리: 이전에 발견한 목표, 행동 결과 및 환경 상태를 포함하여 과거 경험의 지속적인 지식을 인코딩합니다. 장기 메모리를 사용하면 에이전트가 패턴을 인식하고, 전략을 재사용하고, 실수가 반복되지 않도록 할 수 있습니다.

학습 모드

학습 모듈은 다양한 환경과 에이전트 역할을 지원하는 지도 학습, 비지도 학습, 강화 학습과 같은 다양한 패러다임을 지원합니다.

- 지도 학습: 레이블이 지정된 예제를 기반으로 내부 모델을 업데이트합니다. 종종 인적 피드백 또는 훈련 데이터 세트에서 업데이트합니다.

  예: 이전 대화를 기반으로 사용자 의도를 분류하는 방법을 알아봅니다.
- 비지도 학습: 명시적 레이블 없이 데이터에서 숨겨진 패턴 또는 구조를 식별합니다.

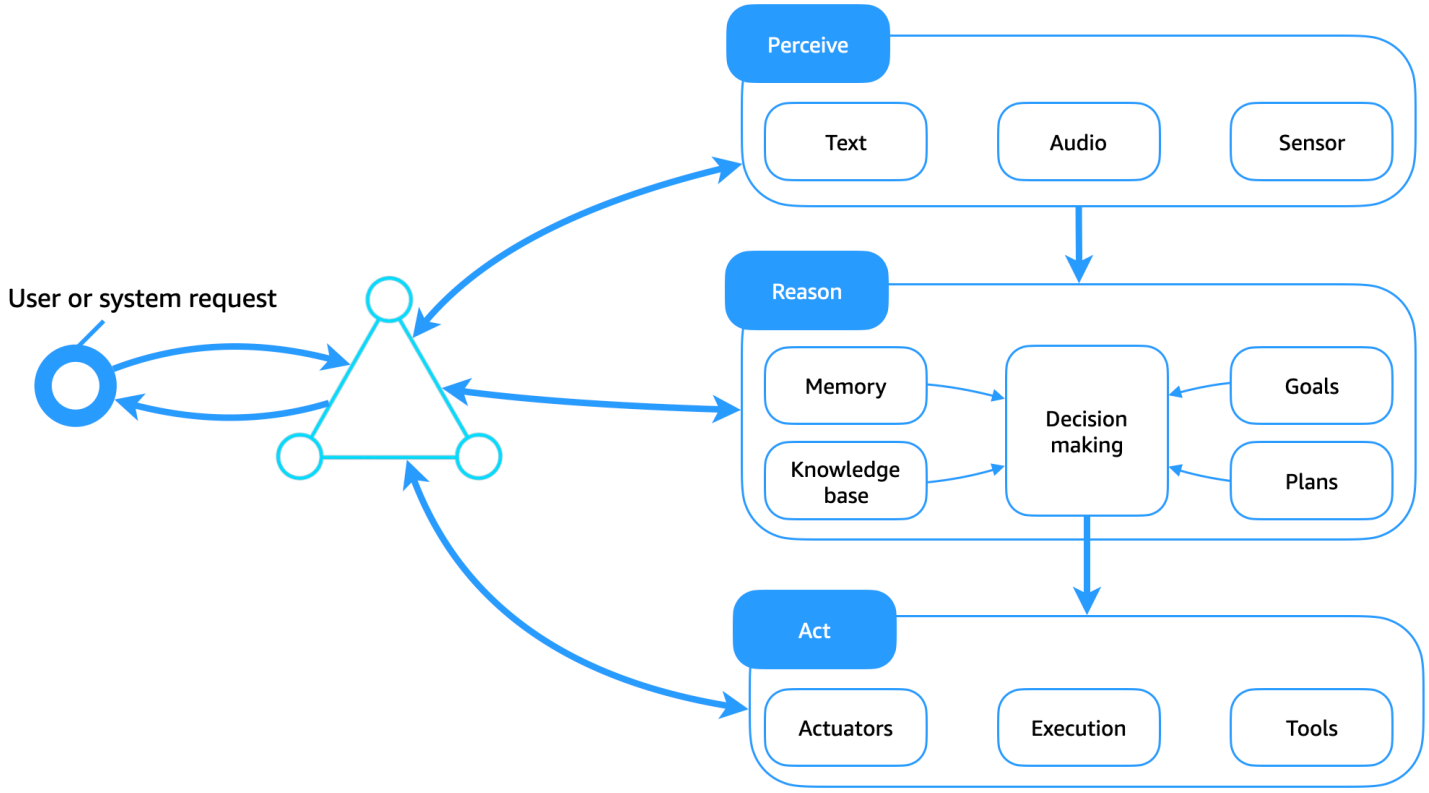
  예: 환경 신호를 클러스터링하여 이상을 감지합니다.
- 강화 학습: 대화형 환경에서 누적 보상을 극대화하여 시도 및 오류를 통해 동작을 최적화합니다.

  예: 어떤 전략이 가장 빠른 작업 완료로 이어지는지 학습합니다.

학습은 에이전트의 인지 모듈과 긴밀하게 통합됩니다. 과거 결과를 기반으로 계획 전략을 구체화하고, 과거 성공 평가를 통해 의사 결정을 개선하며, 지각과 행동 간의 매핑을 지속적으로 개선합니다. 이 닫힌 학습 및 피드백 루프를 통해 에이전트는 사후 대응 실행을 넘어 시간이 지남에 따라 새로운 목표, 조건 및 컨텍스트에 적응할 수 있는 자체 개선 시스템으로 발전합니다.

기존 에이전트 아키텍처: 인지, 이유, 행동

다음 다이어그램은 이전 섹션에서 설명한 구성 요소가 인식, 이유, 행동 주기에 따라 작동하는 방식을 보여줍니다.



인식 모듈

지각 모듈은 에이전트와 외부 세계의 감각 인터페이스 역할을 합니다. 원시 환경 입력을 추론에 정보를 제공하는 구조화된 표현으로 변환합니다. 여기에는 텍스트, 오디오 또는 센서 신호와 같은 멀티모달 데이터 처리가 포함됩니다.

- 텍스트 입력은 사용자 명령, 문서 또는 대화에서 비롯될 수 있습니다.
- 오디오 입력에는 음성 지침 또는 환경 사운드가 포함됩니다.
- 센서 입력은 모션, 시각적 피드 또는 GPS와 같은 실제 신호를 캡처합니다.

원시 입력이 수집되면 지각 프로세스가 특성 추출을 수행한 다음 객체 또는 이벤트 인식 및 의미 해석을 수행하여 현재 상황에 대한 의미 있는 모델을 생성합니다. 이러한 출력은 다운스트림 의사 결정을 위한 구조화된 컨텍스트를 제공하고 에이전트의 추론을 실제 관측치에 고정합니다.

이유 모듈

이유 모듈은 에이전트의 인지 코어입니다. 컨텍스트를 평가하고, 의도를 공식화하고, 적절한 조치를 결정합니다. 이 모듈은 학습한 지식과 추론을 모두 사용하여 목표 기반 행동을 오케스트레이션합니다.

이유 모듈은 긴밀하게 통합된 하위 모듈로 구성됩니다.

- 메모리: 대화 상태, 작업 컨텍스트 및 간헐적 기록을 단기 및 장기 형식으로 유지합니다.
- 지식 기반: 심볼 규칙, 온톨로지 또는 학습된 모델(예: 임베딩, 사실 및 정책)에 대한 액세스를 제공합니다.
- 목표 및 계획: 원하는 결과를 정의하고 이를 달성하기 위한 행동 전략을 수립합니다. 목표는 동적으로 업데이트할 수 있으며 계획은 피드백에 따라 조정적으로 수정할 수 있습니다.
- 의사 결정: 옵션을 평가하고, 장단점을 평가하고, 다음 작업을 선택하여 중앙 중재 엔진 역할을 합니다. 이 하위 모듈은 신뢰도 임계값, 목표 정렬 및 컨텍스트 제약 조건을 고려합니다.

이러한 구성 요소를 함께 사용하면 에이전트가 환경을 추론하고, 신념을 업데이트하고, 경로를 선택하고, 일관된 적응형 방식으로 행동할 수 있습니다. 이유 모듈은 지각과 행동 간의 격차를 해소합니다.

Act 모듈

작업 모듈은 작업을 수행하기 위해 디지털 또는 물리적 환경과 인터페이스하여 에이전트가 선택한 결정을 실행합니다. 여기에서 의도가 조치가 됩니다.

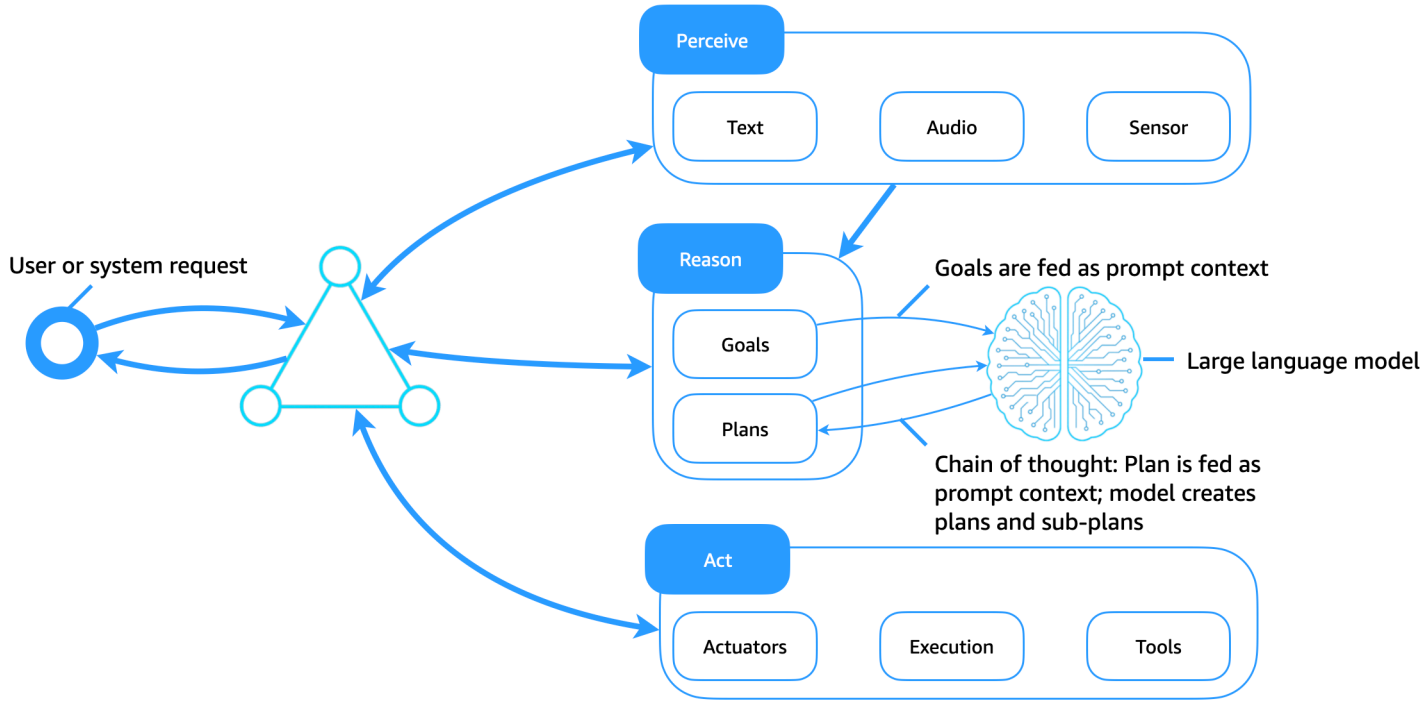
이 모듈에는 세 가지 기능 채널이 포함되어 있습니다.

- 액추에이터: 물리적으로 존재하는 에이전트(예: 로봇 및 IoT 디바이스)의 경우는 이동, 조작 또는 신호와 같은 하드웨어 수준 상호 작용을 제어합니다.
- 실행: APIs 호출, 명령 디스패치, 시스템 업데이트 등 소프트웨어 기반 작업을 처리합니다.
- 도구: 검색, 요약, 코드 실행, 계산 및 문서 처리와 같은 기능 기능을 활성화합니다. 이러한 도구는 동적이고 컨텍스트를 인식하여 에이전트의 유틸리티를 확장하는 경우가 많습니다.

작업 모듈의 출력은 환경으로 다시 공급되고 루프를 닫습니다. 이러한 결과는 에이전트가 다시 인식합니다. 에이전트의 내부 상태를 업데이트하고 향후 결정을 알려 지각, 이유, 행동 주기를 완료합니다.

생성형 AI 에이전트: 심볼 로직LLMs으로 대체

다음 다이어그램은 대규모 언어 모델(LLMs)이 이제 소프트웨어 에이전트를 위한 유연하고 지능적인 인지 코어 역할을 하는 방법을 보여줍니다. 정적 계획 라이브러리 및 수동 코딩 규칙에 의존하는 기존의 심볼 로직 시스템과 달리 LLMs 적응형 추론, 컨텍스트 계획 및 동적 도구 사용을 지원하여 에이전트가 인식, 추론 및 행동하는 방식을 변화시킵니다.



주요 개선 사항

이 아키텍처는 다음과 같이 기존 에이전트 아키텍처를 개선합니다.

- 인지 엔진으로서LLMs: 목표, 계획 및 쿼리는 프롬프트 컨텍스트로 모델에 전달됩니다. LLM은 추론 경로(예: 사고 체인)를 생성하고, 작업을 하위 목표로 분해하고, 다음 작업을 결정합니다.
- 프롬프트를 통한 도구 사용: 도구 사용 에이전트 또는 APIs 및 행동(ReAct) 프롬프트를 통해 LLMs을 지시할 수 있습니다.
- 컨텍스트 인식 계획: 에이전트는 하드 코딩된 계획 라이브러리 없이 에이전트의 현재 목표, 입력 환경 및 피드백을 기반으로 계획을 동적으로 생성하거나 수정합니다.
- 프롬프트 컨텍스트를 메모리로: 에이전트는 심볼 지식 기반을 사용하는 대신 메모리, 계획 및 목표를 모델에 전달되는 프롬프트 토큰으로 인코딩합니다.
- 단 몇 번의 컨텍스트 내 학습을 통한 학습: LLMs 프롬프트 엔지니어링을 통해 동작을 조정하므로 명시적 재훈련 또는 엄격한 계획 라이브러리의 필요성이 줄어듭니다.

LLM 기반 에이전트에서 장기 메모리 확보

구조화된 지식 기반에 장기 메모리를 저장한 기존 에이전트와 달리 생성형 AI 에이전트는 LLMs의 컨텍스트 기간 제한 내에서 작동해야 합니다. 메모리를 확장하고 지속적인 인텔리전스를 지원하기 위해 생

성형 AI 에이전트는 에이전트 스토어, 검색 증강 생성(RAG), 컨텍스트 내 학습 및 프롬프트 체인, 사전 훈련 등 여러 보완 기법을 사용합니다.

에이전트 스토어: 외부 장기 메모리

에이전트 상태, 사용자 기록, 결정 및 결과는 장기 에이전트 메모리 스토어(예: 벡터 데이터베이스, 객체 스토어 또는 문서 스토어)에 저장됩니다. 관련 메모리는 온디맨드 방식으로 검색되고 런타임 시 LLM 프롬프트 컨텍스트에 주입됩니다. 이렇게 하면 에이전트가 세션, 작업 또는 상호 작용 전반에 걸쳐 연속성을 유지하는 영구 메모리 루프가 생성됩니다.

RAG

RAG는 검색된 지식과 생성형 기능을 결합하여 LLM 성능을 향상시킵니다. 목표 또는 쿼리가 실행되면 에이전트는 검색 인덱스를 검색합니다(예: 문서 의미 체계 검색, 이전 대화 또는 구조화된 지식을 통해). 검색된 결과는 LLM 프롬프트에 추가되며, 이는 외부 사실 또는 개인화된 컨텍스트에서 생성을 기반으로 합니다. 이 방법은 에이전트의 유효 메모리를 확장하고 신뢰성과 사실적 정확성을 개선합니다.

컨텍스트 내 학습 및 프롬프트 체인

에이전트는 세션 내 토큰 컨텍스트와 구조화된 프롬프트 체인을 사용하여 단기 메모리를 유지합니다. 현재 계획, 이전 작업 결과, 에이전트 상태와 같은 컨텍스트 요소는 행동 안내를 위한 호출 간에 전달됩니다.

지속적인 사전 훈련 및 미세 조정

도메인별 에이전트의 경우 로그, LLMs을 계속 사전 훈련할 수 있습니다. 또는 인적 피드백(RLHF)을 통한 지침 미세 조정 또는 강화 학습을 통해 에이전트와 유사한 동작을 모델에 직접 포함할 수 있습니다. 이렇게 하면 추론 패턴이 프롬프트 시간 로직에서 모델의 내부 표현으로 전환되고 프롬프트 길이가 줄어들며 효율성이 향상됩니다.

에이전트 AI의 결합된 이점

이러한 기술을 함께 사용하면 생성형 AI 에이전트가 다음을 수행할 수 있습니다.

- 시간이 지남에 따라 상황 인식을 유지합니다.
- 사용자 기록 또는 기본 설정에 따라 동작을 조정합니다.
- up-to-date, 사실적 또는 비공개 지식을 사용하여 결정을 내립니다.
- 지속적이고 규정을 준수하며 설명 가능한 동작을 통해 엔터프라이즈 사용 사례로 확장합니다.

외부 메모리, 검색 계층 및 지속적인 훈련으로 LLMs을 보강하면 에이전트는 심볼 시스템만으로는 이전에는 달성할 수 없었던 수준의 인지 연속성과 목적을 달성할 수 있습니다.

기존 AI와 소프트웨어 에이전트 및 에이전트 AI 비교

다음 표에서는 기존 AI, 소프트웨어 에이전트 및 에이전트 AI를 자세히 비교합니다.

기능	기존 AI	소프트웨어 에이전트	에이전트 AI
예시	스팸 필터, 이미지 분류기, 추천 엔진	챗봇, 작업 스케줄러, 에이전트 모니터링	AI 어시스턴트, 자율 개발자 에이전트, 다중 에이전트 LLM 오케스트레이션
실행 모델	배치 또는 동기	이벤트 기반 또는 예약	비동기식, 이벤트 기반 및 목표 기반
자율성	제한적입니다. 종종 인적 또는 외부 오케스트레이션이 필요합니다.	중간, 사전 정의된 경계 내에서 독립적으로 작동	높음, 적응형 전략과 독립적으로 작동
반응성	입력 데이터에 대한 반응	환경 및 이벤트에 대응	대응적이고 선제적입니다. 조치를 예상하고 시작합니다.
사전 대응	드물게	일부 시스템에 있음	핵심 속성, 목표 지향 동작 유도
통신	미니멀, 일반적으로 독립 실행형 또는 API 바인딩	에이전트 간 또는 에이전트-인간 메시징	풍부한 다중 에이전트 및 human-in-the-loop 상호 작용
의사 결정	모델 추론 전용(분류, 예측 등)	심볼 추론 또는 규칙 기반 또는 스크립팅된 결정	컨텍스트, 목표 기반, 동적 추론(대개 LLM 강화)
위임된 의도	아니요. 사용자가 직접 정의한 작업을 수행합니다.	부분적, 범위가 제한된 사용자 또는 시스템을 대신합니다.	예. 종종 서비스, 사용자 또는 시스템 전반에

기능	기존 AI	소프트웨어 에이전트	에이전트 AI
			서 위임된 목표를 가지고 행동합니다.
학습 및 적응	종종 모델 중심(예: ML 훈련)	때때로 적응형	임베디드 학습, 메모리 또는 추론(예: 피드백, 자체 수정)
기관	없음, 인간을 위한 도구	암시적 또는 기본적	명시적, 목적, 목표 및 자기 주도적으로 운영됨
컨텍스트 인식	낮음, 상태 비저장 또는 스냅샷 기반	보통, 일부 상태 추적	높음, 메모리, 상황 컨텍스트 및 환경 모델 사용
인프라 역할	앱 또는 분석 파이프라인에 포함됨	미들웨어 또는 서비스 계층 구성 요소	클라우드, 서버리스 또는 엣지 시스템과 통합된 구성 가능한 에이전트 메시

요약하면 다음과 같습니다.

- 기존 AI는 도구 중심이며 기능적으로 좁습니다. 예측 또는 분류에 중점을 둡니다.
- 기존 소프트웨어 에이전트는 자율성과 기본 통신을 도입하지만 규칙 바인딩 또는 정적인 경우가 많습니다.
- 에이전트 AI는 자율성, 비동기성 및 기관을 결합합니다. 복잡한 시스템 내에서 추론, 행동 및 적응할 수 있는 지능형 목표 기반 엔터티를 지원합니다. 따라서 에이전트 AI는 클라우드 네이티브 AI 기반 미래에 이상적입니다.

다음 단계

이 가이드에서는 기존 소프트웨어 에이전트가 생성형 AI로 구동되는 자율적이고 지능적인 시스템으로 진화하는 것을 나타내는 에이전트 AI의 역사와 기초에 대해 설명했습니다. 초기 소프트웨어 에이전트가 사전 정의된 규칙과 로직을 따라 고정된 경계 내에서 작업을 자동화하는 방법을 설명하고 에이전트가 개방형 환경에서 동적으로 추론, 학습 및 적응할 수 있는 대규모 언어 모델을 통합하여 에이전트 AI 가이 기반을 구축하는 방법을 설명했습니다.

이 시리즈의 다음 간행물을 검토하여 에이전트 AI를 자세히 살펴볼 수 있습니다.

- 에서 에이전트 AI를 운영하면 AWS 에이전트 AI를 격리된 실험에서 엔터프라이즈 규모의 가치 창출 인프라로 변환할 수 있는 조직 전략을 제공합니다.
- 의 에이전트 AI 패턴 및 워크플로 AWS에서는 목표 지향 AI 에이전트를 설계, 구성 및 오케스트레이션하는 데 사용되는 기본 청사진과 모듈식 구문에 대해 설명합니다.
- 의 에이전트 AI 프레임워크, 프로토콜 및 도구는 AWS 에이전트 AI 솔루션을 구축할 때 고려해야 할 소프트웨어 기반, 툴킷 및 프로토콜을 다룹니다.
- 에서 에이전트 AI용 서버리스 아키텍처를 구축 AWS하면 최신 AI 워크로드의 자연스러운 기반으로서 서버리스 아키텍처에 대해 설명하고에서 AI 네이티브 서버리스 아키텍처를 구축하는 방법을 설명합니다 AWS 클라우드.
- 에서 에이전트 AI를 위한 다중 테넌트 아키텍처를 구축 AWS하면 호스팅 고려 사항, 배포 모델 및 컨트롤 플레인을 포함한 다중 테넌트 설정에서 AI 에이전트를 사용하는 방법을 설명합니다.

리소스

이 가이드에서 설명하는 개념에 대한 자세한 내용은 다음 가이드 및 문서를 참조하세요.

AWS 참조

- Amazon Bedrock Agents
- Amazon Q Developer
- Strands 에이전트 SDK

기타 참조

- Hewitt, Carl, Peter Bishop, Richard Steiger. “인공 지능을 위한 범용 모듈식 ACTOR 공식화.” 제3차 인공 지능 공동 컨퍼런스(1973) 논문: 235-245. https://www.ijcai.org/Proceedings/73/Papers/027B.pdf
- Lesser, Victor R., 관련 간행물(전체 목록 참조):
  - Lesser, Victor R. 및 Daniel D. Corkill. "함수적으로 정확하고 협력적인 분산 시스템." Systems, Man 및 Cybernetics 11, no. 1(1981): 81-96. https://ieeexplore.ieee.org/abstract/document/4308581
  - Decker, Keith S. 및 Victor R. Lesser. “협정 서비스에서의 커뮤니케이션.” AAAI Workshop on Planning for Interagent Communication (1994). https://www.researchgate.net/profile/Victor-Lesser/publication/2768884_Communication_in_the_Service_of_Coordination/links/00b7d51cc2a0750cb4000000/Communication-in-the-Service-of-Coordination.pdf
  - Durfee, Edmund H., Victor R. Lesser 및 Daniel D. Corkill. "협동 분산 문제 해결의 추세" IEEE Transactions on knowledge and data Engineering(1989). http://mas.cs.umass.edu/Documents/ieee-tkde89.pdf
  - Durfee, Edmund H., V.R. Lesser 및 D.D. Corkill, “분산 인공 지능.” 분산 문제 해결 네트워크의 통신을 통한 협력(1987): 29-58. https://www.academia.edu/download/79885643/durf94_1.pdf
  - Lâasri, Brigitte, Hassan Lâasri, Susan Lander 및 Victor Lesser. “지능형 협상 에이전트를 위한 일반 모델.” International Journal of Cooperative Information Systems 01, no. 02(1992): 291-317. https://doi.org/10.1142/S0218215792000210
  - Lander, Susan E. 및 Victor R. Lesser. “이종 에이전트 간의 분산 검색에서 협상의 역할 이해.” IJCAI'93: 인공 지능에 대한 제13차 국제 공동 회의(1993)의 절차: 438-444. https://www.ijcai.org/Proceedings/93-1/Papers/062.pdf

  - Lander, Susan, Victor R. Lesser 및 Margaret E. Connell. “전문 에이전트 협력을 위한 충돌 해결 전략” CKBS'90: 지식 기반 시스템 협력에 대한 국제 실무 회의(1990년 10월): 183-200. https://doi.org/10.1007/978-1-4471-1831-2_10
  - Prasad, M.V. Nagendra, Victor Lesser 및 Susan E. Lander. “이종 다중 에이전트 시스템에서 실험 학습.” 다중 에이전트 시스템의 적응 및 학습에 대한 IJCAI-95 워크숍(1995): 59-64. https://www.researchgate.net/publication/2784280_Learning_Experiments_in_a_Heterogeneous_Multi-agent_System
- Nwana, Hyacinth S. "소프트웨어 에이전트: 개요" 지식 엔지니어링 검토 11, 아니요. 3(1996년 10월/11월): 205-244. https://teaching.shu.ac.uk/aces/rh1/elearning/multiagents/introduction/nwana.pdf
- Selfridge, Oliver G. "판데모니움: 학습의 패러다임" 사고 프로세스의 메커니즘: 국립 물리 연구소 1(1959년)에서 개최된 심포지엄의 절차: 511~529. https://aitopics.org/download/classics:504E1BAC
- Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N.™, Lukasz Kaiser 및 Illia Meshsukhin. “주의는 필요한 것뿐입니다.” 신경 정보 처리 시스템(NIPS)에 대한 31차 컨퍼런스의 절차. 신경 정보 처리 시스템 30의 발전(2017): 5998-6008. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
- ™dridge, Michael 및 Nicholas R. Jennings. "지능형 에이전트: 이론 및 실습" 지식 엔지니어링 검토 10, 아니요. 2(1995년 1월): 115-152. https://www.cs.cmu.edu/~motionplanning/papers/sbp_papers/integrated1/woodridge_intelligent_agents.pdf

문서 기록

아래 표에 이 가이드의 주요 변경 사항이 설명되어 있습니다. 향후 업데이트에 대한 알림을 받으려면 RSS 피드를 구독하십시오.

변경 사항	설명	날짜
최초 게시	—	2025년 7월 14일

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. 예를 들어, 온프레미스 Oracle 데이터베이스를 Amazon Aurora PostgreSQL 호환 버전으로 마이그레이션합니다.
- 리플랫포밍(리프트 앤드 리셰이프) - 애플리케이션을 클라우드로 이동하고 일정 수준의 최적화를 도입하여 클라우드 기능을 활용합니다. 예: 온프레미스 Oracle 데이터베이스를 AWS 클라우드의 Amazon Relational Database Service(Amazon RDS)로 마이그레이션합니다.
- 재구매(드롭 앤드 숍) - 일반적으로 기존 라이선스에서 SaaS 모델로 전환하여 다른 제품으로 전환합니다. 예를 들어, 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.
- 재배치(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 나중으로 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, 격리성, 내구성을 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 마이그레이션보다 더 많은 작업이 필요합니다.

능동-수동 마이그레이션

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

집계 함수

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

AI

인공 지능을 참조하세요.

AIOps

인공 지능 운영을 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 프로세스의 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 가이드를 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 AWS를 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 전환 AWS 하기 위한 효율적이고 효과적인 계획을 개발하는 데 도움이 되는의 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 웹사이트와 AWS CAF 백서를 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 봇입니다.

BCP

비즈니스 연속성 계획을 참조하세요.

동작 그래프

리소스 동작과 시간 경과에 따른 상호 작용에 대한 통합된 대화형 뷰입니다. Amazon Detective에서 동작 그래프를 사용하여 실패한 로그온 시도, 의심스러운 API 호출 및 유사한 작업을 검사할 수 있습니다. 자세한 내용은 Detective 설명서의 Data in a behavior graph를 참조하십시오.

빅 엔디안 시스템

가장 중요한 바이트를 먼저 저장하는 시스템입니다. 엔디안도 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

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 입히기도 합니다.

봇넷

맬웨어에 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 접근 권한

예외적인 상황에서 승인된 프로세스를 통해 사용자가 일반적으로 액세스할 권한이 없는에 액세스할 수 AWS 계정 있는 빠른 방법입니다. 자세한 내용은 AWS Well-Architected 지침의 긴급 접근 절차 구현 표시기를 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 Framework를 참조하세요.

카나리 배포

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

CCoE

클라우드 혁신 센터를 참조하세요.

CDC

데이터 캡처 변경을 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 전달을 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

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 게시물을 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 구축을 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 가이드를 참조하세요.

CMDB

구성 관리 데이터베이스를 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 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 팩을 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 이점을 참조하십시오. CD는 지속적 배포를 의미하기도 합니다. 자세한 내용은 지속적 전달(Continuous Delivery)과 지속적인 개발을 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 분류를 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 언어를 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 제어를 AWS추가하여 리소스를 보호할 수 있습니다. 예를 들어, 심층 방어 접근 방식은 다단계 인증, 네트워크 세분화 및 암호화를 결합할 수 있습니다.

위임된 관리자

에서 AWS Organizations호환되는 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 Organizations와 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

환경을 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 제어를 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 스키마에서 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)

재해로 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 Cloud를 참조하세요.

DML

데이터베이스 조작 언어를 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 방법을 참조하십시오.

DR

재해 복구를 참조하세요.

드리프트 감지

기준이 되는 구성과의 편차 추적을 말합니다. 예를 들어 AWS CloudFormation 를 사용하여 시스템 리소스의 드리프트를 감지하거나를 사용하여 AWS Control Tower 거버넌스 요구 사항 준수에 영향을 미칠 수 있는 랜딩 존의 변경 사항을 감지할 수 있습니다.

DVSM

개발 가치 흐름 매핑을 참조하세요.

E

EDA

탐색 데이터 분석을 참조하세요.

EDI

전자 데이터 교환을 참조하세요.

엣지 컴퓨팅

IoT 네트워크의 엣지에서 스마트 디바이스의 컴퓨팅 성능을 개선하는 기술 엣지 컴퓨팅은 클라우드 컴퓨팅에 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 엔드포인트를 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 생성을 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 암호화를 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 마이그레이션 전략의 에픽에 대한 자세한 내용은 프로그램 구현 가이드를 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

스타 스키마의 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 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 SHAP(Shapley Additive Descriptions) 및 통합 그레디언트와 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 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.

퓨샷 프롬프팅

유사한 태스크를 수행하도록 요청하기 전에 LLM에 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 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 캡처를 통해 지속적 데이터 복제를 사용하여 최단 시간에 데이터를 마이그레이션하는 데이터베이스 마이그레이션 방법입니다. 목표는 가동 중지 시간을 최소화하는 것입니다.

FM

파운데이션 모델을 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 무엇인가요?를 참조하세요.

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 제한을 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 제한을 참조하십시오.

Gitflow 워크플로

하위 환경과 상위 환경이 소스 코드 리포지토리의 서로 다른 브랜치를 사용하는 방식입니다. Gitflow 워크플로는 레거시로 간주되며 트렁크 기반 워크플로는 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 디바이스 제조 작업의 속도, 확장성 및 생산성을 개선할 수 있습니다.

브라운필드 전략

새로운 환경에서 기존 인프라의 부재 시스템 아키텍처에 대한 그린필드 전략을 채택할 때 브라운필드라고도 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.

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 제공합니다.

높은 가용성(HA)

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

히스토리언 현대화

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

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.

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 리플랫포밍 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.

정보

IaC

코드형 인프라를 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 인터넷을 참조하세요.

변경 불가능한 인프라

기존 인프라를 업데이트, 패치 또는 수정하는 대신 프로덕션 워크로드에 대한 새 인프라를 배포하는 모델입니다. 변경 불가능한 인프라는 변경 가능한 인프라보다 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 Architecture에서는 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 Schwab에서 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 구축을 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 Architecture에서는 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 인터넷을 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(TSM)

조직의 IT 서비스 설계, 구현, 관리 및 지원과 관련된 활동 클라우드 운영을 ITSM 도구와 통합하는 방법에 대한 자세한 내용은 운영 통합 가이드를 참조하십시오.

ITIL

IT 정보 라이브러리를 참조하세요.

ITSM

IT 서비스 관리를 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 설정을 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 제어를 참조하세요.

최소 권한

작업을 수행하는 데 필요한 최소 권한을 부여하는 보안 모범 사례입니다. 자세한 내용은 IAM 설명서의 최소 권한 적용을 참조하십시오.

리프트 앤드 시프트

7R을 참조하세요.

리틀 엔디안 시스템

가장 덜 중요한 바이트를 먼저 저장하는 시스템입니다. 엔디안도 참조하세요.

LLM

대규모 언어 모델을 참조하세요.

하위 환경

환경을 참조하세요.

M

기계 학습(ML)

패턴 인식 및 학습에 알고리즘과 기법을 사용하는 인공 지능의 한 유형입니다. ML은 사물 인터넷(IoT) 데이터와 같은 기록된 데이터를 분석하고 학습하여 패턴을 기반으로 통계 모델을 생성합니다. 자세한 내용은 기계 학습을 참조하십시오.

기본 브랜치

브랜치를 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 서비스 는 인프라 계층, 운영 체제 및 플랫폼을 AWS 운영하고, 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 Program을 참조하세요.

메커니즘

도구를 생성하고 도구 채택을 유도한 다음 조정을 위해 결과를 검사하는 전체 프로세스입니다. 메커니즘은 작동 시 자체를 강화하고 개선하는 주기입니다. 자세한 내용은 AWS Well-Architected Framework의 메커니즘 구축을 참조하세요.

멤버 계정

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

MES

제조 실행 시스템을 참조하세요.

메시지 큐 원격 분석 전송(MQTT)

리소스 제약이 있는 IoT 디바이스에 대한 게시/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 통합을 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 전략의 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 가이드와 마이그레이션 팩토리에 대한 설명을 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 도구. 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 가이드를 참조하십시오. MRA는 AWS 마이그레이션 전략의 첫 번째 단계입니다.

마이그레이션 전략

워크로드를 AWS 클라우드로 마이그레이션하는 데 사용되는 접근 방식입니다. 자세한 내용은 이 용어집의 7R 항목과 조직을 동원하여 대규모 마이그레이션을 가속화를 참조하세요.

ML

기계 학습을 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 애플리케이션 현대화 전략을 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 평가를 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 분해를 참조하십시오.

MPA

Migration Portfolio Assessment를 참조하세요.

MQTT

메시지 큐 원격 분석 전송을 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 Framework에서는 변경 불가능한 인프라를 모범 사례로 사용할 것을 권장합니다.

O

OAC

오리진 액세스 제어를 참조하세요.

OAI

오리진 액세스 ID를 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

운영 통합을 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 Architecture를 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 가이드를 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 계정 에 생성되고 각 계정의 활동을 추적합니다. 자세한 내용은 CloudTrail 설명서의 Creating a trail for an organization을 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 가이드를 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 특

정 CloudFront 배포를 통해서만 S3 버킷의 콘텐츠에 액세스할 수 있습니다. 더 세분화되고 향상된 액세스 제어를 제공하는 OAC도 참조하십시오.

ORR

운영 준비 상태 검토를 참조하세요.

OT

운영 기술을 참조하세요.

아웃바운드(송신) VPC

AWS 다중 계정 아키텍처에서 애플리케이션 내에서 시작된 네트워크 연결을 처리하는 VPC입니다. AWS Security Reference Architecture에서는 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 경계를 참조하십시오.

개인 식별 정보(PII)

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

PII

개인 식별 정보를 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 컨트롤러를 참조하세요.

PLM

제품 수명 주기 관리를 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. 자세한 내용은 마이크로서비스에서 데이터 지속성 활성화를 참조하십시오.

포트폴리오 평가

마이그레이션을 계획하기 위해 애플리케이션 포트폴리오를 검색 및 분석하고 우선순위를 정하는 프로세스입니다. 자세한 내용은 마이그레이션 준비 상태 평가를 참조하십시오.

조건자

보통 `WHERE` 절에 있는 `true` 또는 `false`를 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 controls를 참조하십시오.

보안 주체

작업을 수행하고 리소스에 액세스할 수 AWS 있는의 엔터티입니다. 이 엔터티는 일반적으로 , AWS 계정 IAM 역할 또는 사용자의 루트 사용자입니다. 자세한 내용은 IAM 설명서의 역할 용어 및 개념의 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 작업을 참조하십시오.

선제적 제어

규정 미준수 리소스의 배포를 방지하도록 설계된 보안 제어입니다. 이러한 제어는 리소스를 프로비저닝하기 전에 리소스를 스캔합니다. 리소스가 제어를 준수하지 않으면 프로비저닝되지 않습니다. 자세한 내용은 AWS Control Tower 설명서의 제어 참조 가이드를 참조하고 보안 제어 구현의 사전 예방적 제어를 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

환경을 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 확장하는 데 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 MES에서 마이크로서비스는 다른 마이크로서비스가 구독할 수 있는 채널에 이벤트 메시지를 게시할 수 있습니다. 시스템은 게시 서비스를 변경하지 않고도 새 마이크로서비스를 추가할 수 있습니다.

Q

쿼리 계획

SQL 관계형 데이터베이스 시스템의 데이터에 액세스하는 데 사용되는 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 생성을 참조하세요.

랜섬웨어

결제가 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 제어를 참조하세요.

읽기 전용 복제본

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

리아키텍팅

7R을 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

7R을 참조하세요.

리전

지리적 영역의 AWS 리소스 모음입니다. 각 AWS 리전 는 내결함성, 안정성 및 복원력을 제공하기 위해 서로 격리되고 독립적입니다. 자세한 내용은 계정에서 사용할 수 있는 AWS 리전 지정을 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

7R을 참조하세요.

release

배포 프로세스에서 변경 사항을 프로덕션 환경으로 승격시키는 행위입니다.

재배치

7R을 참조하세요.

리플랫포밍

7R을 참조하세요.

재구매

7R을 참조하세요.

복원력

중단에 저항하거나 중단을 복구할 수 있는 애플리케이션의 기능입니다. 고가용성 및 재해 복구는 AWS 클라우드에서 복원력을 계획할 때 일반적인 고려 사항입니다. 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.

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 제어를 참조하세요.

retain

7R을 참조하세요.

사용 중지

7R을 참조하세요.

검색 증강 세대(RAG)

응답을 생성하기 전에 LLM이 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 대한 시맨틱 검색을 수행할 수 있습니다. 자세한 내용은 RAG란 무엇인가요?를 참조하세요.

교체

공격자가 자격 증명에 액세스하는 것을 더욱 어렵게 만들기 위해 보안 암호를 주기적으로 업데이트하는 프로세스입니다.

행 및 열 액세스 제어(RCAC)

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

RPO

복구 시점 목표를 참조하세요.

RTO

복구 시간 목표를 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 콘솔 사용할 수 있으므로 사용자는 조직의 모든 사용자에 대해 IAM에서 사용자를 생성하지 않고도에 로그인하거나 AWS API 작업을 호출할 수 있습니다. SAML 2.0 기반 페더레이션에 대한 자세한 내용은 IAM 설명서의 SAML 2.0 기반 페더레이션 정보를 참조하십시오.

SCADA

감독 제어 및 데이터 획득을 참조하세요.

SCP

서비스 제어 정책을 참조하세요.

보안 암호

에는 암호화된 형식으로 저장하는 암호 또는 사용자 자격 증명과 같은 AWS Secrets Manager기밀 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 제어에는 예방, 탐지, 대응 및 선제적 제어의 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)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

대상에서 데이터를 수신하는 AWS 서비스 에 의한 데이터 암호화.

서비스 제어 정책(SCP)

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

서비스 수준에 관한 계약(SLA)

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

서비스 수준 표시기(SLI)

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

서비스 수준 목표(SLO)

서비스 수준 지표로 측정되는 서비스의 상태를 나타내는 목표 지표입니다.

공동 책임 모델

클라우드 보안 및 규정 준수를 AWS 위해와 공유하는 책임을 설명하는 모델입니다. AWS 는 클라우드의 보안을 담당하는 반면, 사용자는 클라우드의 보안을 담당합니다. 자세한 내용은 공동 책임 모델을 참조하십시오.

SIEM

보안 정보 및 이벤트 관리 시스템을 참조하세요.

단일 장애점(SPOF)

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

SLA

서비스 수준 계약을 참조하세요.

SLI

서비스 수준 표시기를 참조하세요.

SLO

서비스 수준 목표를 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 방식을 참조하세요.

SPOF

단일 장애점을 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 웨어하우스에서 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 압도

하고 대체하는 것과 비슷합니다. Martin Fowler가 모놀리식 시스템을 다시 작성할 때 위험을 관리하는 방법으로 이 패턴을 도입했습니다. 이 패턴을 적용하는 방법의 예는 컨테이너 및 Amazon API Gateway를 사용하여 기존의 Microsoft ASP.NET(ASMX) 웹 서비스를 점진적으로 현대화하는 방법을 참조하십시오.

서브넷

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

감독 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 Synthetics를 사용하여 이러한 테스트를 생성할 수 있습니다.

시스템 프롬프트

LLM에 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 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.

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 사용을 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. 자세한 내용은 Quantifying uncertainty in deep learning systems 가이드를 참조하십시오.

차별화되지 않은 작업

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

상위 환경

환경을 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

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 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 Many를 참조하세요.

WQF

AWS Workload Qualification Framework를 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 불가능한 항목으로 간주됩니다.

Z

제로데이 익스플로잇

제로데이 취약성을 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 LLM에 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 프롬프팅도 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.