의 에이전트 AI 프레임워크, 플랫폼, 프로토콜 및 도구 AWS

AWS 권장 가이드

AWS 권장 가이드: 의 에이전트 AI 프레임워크, 플랫폼, 프로토콜 및 도구 AWS

Table of Contents

의 에이전트 AI 프레임워크, 플랫폼, 프로토콜 및 도구 AWS

Aaron Sempf, Ansley Verzosa, Joshua Samuel, Amazon Web Services(AWS)

2026년 1월(문서 기록)

에이전트 AI는 AI, 분산 시스템 및 소프트웨어 엔지니어링의 교차점에서 강력한 패러다임입니다. AI 모델을 사용하고 도구 및 리소스와 통합되는 자율 비동기 소프트웨어 에이전트로 구성된 지능형 시스템 클래스입니다. 에이전트는 기관을 대표하고, 컨텍스트, 목표에 대한 이유를 인식하고, 결정을 내리고, 사용자 또는 시스템을 대신하여 의도적인 조치를 취할 수 있습니다. 이러한 에이전트는 분산된 환경 내에서 독립적으로, 종종 협업하여 운영되며 임베디드 인텔리전스, 메모리 및 의도를 통해 위임된 목표를 추구하도록 설계되었습니다.

에서 AWS조직은 에이전트 AI를 활용하여 복잡한 워크플로를 자동화하고, 의사 결정 프로세스를 개선하고, 응답성이 뛰어난 시스템을 만들 수 있습니다. 이 가이드에서는 효과적인 에이전트 AI 솔루션을 구축하는 데 필요한 주요 구성 요소에 대한 정보를 제공합니다.

- 프레임워크는 이점 및 사용 사례에 대한 검토를 포함하여 현재 에이전트 AI 프레임워크를 프로파일링합니다. 이러한 프레임워크가 패턴, 프로토콜 및 도구에서 차별화되지 않은 과도한 부담을 줄이는 방법을 알아봅니다. 키 선택 기준을 이해하여 요구 사항에 적합한 프레임워크를 선택합니다.
- 플랫폼은 에이전트 AI 플랫폼(관리형 에이전트, 오픈 소스 오케스트레이션 및 하이브리드)의 개요와 선택 또는 설계 고려 사항을 제공합니다.
- 프로토콜은 에이전트 상호 작용을 위한 필수 표준화된 통신 프로토콜을 탐색합니다. 오픈 소스 모델 컨텍스트 프로토콜(MCP) 및 Agent2Agent(A2A)와 같은 Agent-to-agent 프로토콜이 다른 독점 구현과 함께 등장하고 있습니다. 공통 프로토콜을 통해 서로 다른 프로토콜이 원활하게 상호 작용하는 방법을 알아봅니다.
- 도구는 프로토콜 기반 도구(예: MCP), 프레임워크 네이티브 도구 및 메타 도구에 대한 정보를 제공합니다. 조직은 워크플로의 주요 시스템과 통합되는 툴킷을 구축하여 최종 사용자 및 서버 기반 에이전트 워크플로를 모두 활성화할 수 있습니다.

대상 독자

이 가이드는 최신 클라우드 네이티브 애플리케이션 내에서 AI 기반 소프트웨어 에이전트의 성능을 활용하려는 아키텍트, 개발자 및 기술 리더를 위한 것입니다.

목표

이 가이드는 다음을 수행하는 데 도움이 됩니다.

- 다양한 에이전트 AI 프레임워크를 비교하여 사용 사례에 가장 적합한 프레임워크를 선택합니다.
- 개별 에이전트를 조정된 적응형 시스템으로 전환하는 기능을 제공하는 에이전트 AI 플랫폼에 대해 알아봅니다.
- 지속 가능한 에이전트 AI 아키텍처를 구축하기 위한 개방형 프로토콜의 이점을 이해합니다.
- 에이전트 시스템을 구축할 때 적절한 도구 통합 전략을 수립합니다.

이 콘텐츠 시리즈 정보

이 가이드는 에이전트 AI on에 대한 시리즈의 일부입니다 AWS. 자세한 내용과이 시리즈의 다른 가이드를 보려면 AWS 권장 가이드 웹 사이트의 에이전트 AI를 참조하세요.

프레임워크

의 에이전트 AI 기반 AWS은 자율적이고 목표 지향적인 동작을 가능하게 하는 핵심 패턴과 워크플로를 검사합니다. 이러한 패턴을 구현하는 핵심에는 프레임워크 선택이 있습니다. 프레임워크는 미리 작성된 코드의 소프트웨어 기반이며, 프로덕션 지원 자율 AI 에이전트를 구축하는 데 필요한 구축 및 관리, 도구 및 오케스트레이션 기능을 위한 구조화된 환경과 공통 기능을 제공합니다.

효과적인 에이전트 AI 프레임워크는 원시 대규모 언어 모델(LLM) 상호 작용을 추론, 협업 및 조치를 수행할 수 있는 조정된 지능형 시스템으로 변환하는 몇 가지 필수 기능을 제공합니다.

- 에이전트 오케스트레이션은 단일 또는 여러 에이전트의 정보 흐름과 의사 결정을 조정하여 사람의 개입 없이 복잡한 목표를 달성합니다.
- 도구 통합을 통해 에이전트는 외부 시스템, APIs 및 데이터 소스와 상호 작용하여 언어 처리 이상으로 기능을 확장할 수 있습니다. 자세한 내용은 Strands Agents 설명서의 도구 개요를 참조하세요.
- 메모리 관리는 장기 실행 또는 적응 작업에 필수적인 상호 작용 전반에 걸쳐 컨텍스트를 유지할 수 있는 영구 또는 세션 기반 상태를 제공합니다. 고급 프레임워크는 장기 메모리를 통합하여 요약 및 사용자 기본 설정을 저장하므로 개인화되고 상황에 맞는 에이전트 경험을 제공할 수 있습니다. 자세한 내용은 LangChain 블로그의 에이전트 프레임워크에 대해 생각하는 방법을 참조하세요.
- 워크플로 정의는 정교한 자율 추론을 가능하게 하는 체인, 라우팅, 병렬화 및 반사 루프와 같은 구조화된 패턴을 지원합니다.
- 배포 및 모니터링을 통해 자율 시스템을 관찰하면서 개발에서 프로덕션으로 전환할 수 있습니다. 자세한 내용은 Amazon Bedrock AgentCore 일반 가용성 공지를 참조하세요.

이러한 기능은 프레임워크 환경 전체에서 다양한 접근 방식과 emphase로 구현되며, 각각 다양한 자율 에이전트 사용 사례 및 조직 컨텍스트에 고유한 이점을 제공합니다.

이 섹션에서는 자율 운영을 위한 장점, 제한 사항 및 이상적인 사용 사례에 중점을 두고 에이전트 AI 솔루션을 구축하기 위한 주요 프레임워크를 프로파일링하고 비교합니다.

- Strands 에이전트
- LangChain 및 LangGraph
- CrewAI
- AutoGen
- ???
- 에이전트 AI 프레임워크 비교

> **Note**
>
> 이 섹션에서는 특히 AI 기관을 지원하는 프레임워크를 다루며 기관 없이 프런트엔드 인터페이스 또는 생성형 AI는 다루지 않습니다.

Strands Agents

Strands Agents는 오픈 소스 블로그에 설명된 AWS대로에서 처음 릴리스한 오픈 소스 SDK입니다. Strands Agents는 모델 우선 접근 방식으로 자율 AI 에이전트를 구축하도록 설계되었습니다.는 타사 구성 요소와의 통합을 열린 상태로 AWS 서비스 유지하면서와 원활하게 작동하도록 설계된 유연하고 확장 가능한 프레임워크를 제공합니다. AWS Strands Agents는 완전 자율 솔루션을 구축하는 데 적합합니다.

의 주요 기능 Strands Agents

Strands Agents 에는 다음과 같은 주요 기능이 포함되어 있습니다.

- 모델 우선 설계 - 파운데이션 모델이 에이전트 인텔리전스의 핵심이라는 개념을 기반으로 구축되어 정교한 자율 추론을 가능하게 합니다. 자세한 내용은 Strands Agents 설명서의 에이전트 루프를 참조하세요.
- 다중 에이전트 공동 작업 패턴 - 분산 에이전트 네트워크에서 확장 가능한 공동 작업 및 거버넌스를 지원하는 Swarm, 그래프 및 워크플로 패턴과 같은 기본 제공 조정 모델입니다. 자세한 내용은 Strands Agents 설명서의 다중 에이전트 패턴을 참조하세요.
- MCP 통합 - 모델 컨텍스트 프로토콜(MCP)을 기본적으로 지원하므로 일관된 자율 운영을 위해 LLMs에 표준화된 컨텍스트를 프로비저닝할 수 있습니다.
- AWS 서비스 통합 - 포괄적인 자율 워크플로를 위해 Amazon Bedrock AWS Lambda AWS Step Functions, 및 기타 AWS 서비스 에 원활하게 연결합니다. 자세한 내용은 AWS 주간 라운드업(AWS 블로그)을 참조하세요.
- 파운데이션 모델 선택 - Anthropic Claude, Amazon Bedrock의 Amazon Nova(Premier, Pro, Lite, Micro) 등 다양한 파운데이션 모델을 지원하여 다양한 자율 추론 기능을 최적화합니다. 자세한 내용은 Strands Agents 설명서의 Amazon Bedrock을 참조하세요.
- LLM API 통합 - 프로덕션 배포를 위해 Amazon Bedrock, OpenAI 등 다양한 LLM 서비스 인터페이스와 유연하게 통합됩니다. 자세한 내용은 Strands Agents 설명서의 Amazon Bedrock Basic Usage를 참조하세요.

- 멀티모달 기능 - 포괄적인 자율 에이전트 상호 작용을 위해 텍스트, 음성 및 이미지 처리를 포함한 여러 모달리티를 지원합니다. 자세한 내용은 Strands Agents 설명서의 Amazon Bedrock Multimodal Support를 참조하세요.
- 도구 에코시스템 - 자율 기능을 확장하는 사용자 지정 도구의 확장성과 함께 AWS 서비스 상호 작용을 위한 다양한 도구 세트입니다. 자세한 내용은 Strands Agents 설명서의 도구 개요를 참조하세요.

Strands Agents(을)를 사용해야 하는 경우

Strands Agents는 다음과 같은 자율 에이전트 시나리오에 특히 적합합니다.

- 자율 워크플로를 AWS 서비스 위해와 네이티브 통합을 원하는 AWS 인프라를 기반으로 하는 조직
- 프로덕션 자율 시스템에 엔터프라이즈급 보안, 확장성 및 규정 준수 기능이 필요한 팀
- 특수 자율 태스크를 위해 다양한 공급자에서 모델 선택의 유연성이 필요한 프로젝트
- 엔드 투 엔드 자율 프로세스를 위해 기존 AWS 워크플로 및 리소스와 긴밀하게 통합해야 하는 사용 사례

에 대한 구현 접근 방식 Strands Agents

Strands Agents는 빠른 시작 안내서에 설명된 대로 비즈니스 이해관계자에게 간단한 구현 접근 방식을 제공합니다. 프레임워크를 통해 조직은 다음을 수행할 수 있습니다.

- 특정 비즈니스 요구 사항에 따라 Amazon Bedrock에서 Amazon Nova(Premier, Pro, Lite 또는 Micro)와 같은 파운데이션 모델을 선택합니다.
- 엔터프라이즈 시스템 및 데이터 소스에 연결하는 사용자 지정 도구를 정의합니다.
- 텍스트, 이미지, 스피치를 포함한 여러 모달리티를 처리합니다.
- 비즈니스 쿼리에 자율적으로 응답하고 작업을 수행할 수 있는 에이전트를 배포합니다.

이 구현 접근 방식을 통해 비즈니스 팀은 AI 모델 개발에 대한 심층적인 기술 전문 지식 없이 자율 에이전트를 신속하게 개발하고 배포할 수 있습니다.

의 실제 예 Strands Agents

AWS Transform for .NET은 Strands Agents를 사용하여 대규모로 .NET 애플리케이션을 현대화하기 위한 AWS Transform 최초의 에이전트 AI 서비스인 for .NET(블로그)에 설명된 대로 애플리케이션 현대화 기능을 지원합니다.AWS 이 프로덕션 서비스는 여러 전문 자율 에이전트를 사용합니다. 에이전트

는 협력하여 사람의 개입 없이 레거시 .NET 애플리케이션을 분석하고, 현대화 전략을 계획하고, 클라우드 네이티브 아키텍처로 코드 변환을 실행합니다. AWS Transform for .NET은 엔터프라이즈 자율 시스템을 Strands Agents 위한의 프로덕션 준비 상태를 보여줍니다.

LangChain 및 LangGraph

LangChain는 에이전트 AI 에코시스템에서 가장 많이 확립된 프레임워크 중 하나입니다.는 LangChain 블로그에 설명된 대로 복잡한 상태 저장 에이전트 워크플로를 지원하도록 기능을 LangGraph 확장합니다. 이들은 함께 독립적인 운영을 위한 풍부한 오케스트레이션 기능을 갖춘 정교한 자율 AI 에이전트를 구축하기 위한 포괄적인 솔루션을 제공합니다.

LangChain 및의 주요 기능 LangGraph

LangChain 및 에는 다음과 같은 주요 기능이 LangGraph 포함됩니다.

- 구성 요소 에코시스템 - 다양한 자율 에이전트 기능을 위해 사전 구축된 구성 요소의 광범위한 라이브러리로, 특수 에이전트를 신속하게 개발할 수 있습니다. 자세한 내용은 LangChain 설명서의 Quickstart를 참조하세요.
- 파운데이션 모델 선택 - Anthropic Claude, Amazon Bedrock의 Amazon Nova 모델(Premier, Pro, Lite, Micro) 등 다양한 추론 기능을 위한 다양한 파운데이션 모델을 지원합니다. 자세한 내용은 LangChain 설명서의 입력 및 출력을 참조하세요.
- LLM API 통합 - 유연한 배포를 위해 Amazon Bedrock, OpenAI등을 포함한 여러 대규모 언어 모델(LLM) 서비스 공급자를 위한 표준화된 인터페이스입니다. 자세한 내용은 LangChain 설명서의 LLMs을 참조하세요.
- 멀티모달 처리 - 텍스트, 이미지 및 오디오 처리를 기본적으로 지원하여 풍부한 멀티모달 자율 에이전트 상호 작용을 지원합니다. 자세한 내용은 LangChain 설명서의 다중 양식을 참조하세요.
- 그래프 기반 워크플로 - 복잡한 자율 에이전트 동작을 상태 머신으로 정의LangGraph하여 정교한 의사 결정 로직을 지원합니다. 자세한 내용은 LangGraph 플랫폼 GA 발표를 참조하세요.
- 메모리 추상화 - 단기 및 장기 메모리 관리를 위한 다양한 옵션으로, 시간이 지남에 따라 컨텍스트를 유지하는 자율 에이전트에 필수적입니다. 자세한 내용은 LangChain 설명서의 챗봇에 메모리를 추가하는 방법을 참조하세요.
- 도구 통합 - 다양한 서비스 및 APIs으로 자율 에이전트 기능을 확장합니다. 자세한 내용은 LangChain 설명서의 도구를 참조하세요.
- LangGraph 플랫폼 - 장기 실행 자율 에이전트를 지원하는 프로덕션 환경을 위한 관리형 배포 및 모니터링 솔루션입니다. 자세한 내용은 LangGraph 플랫폼 GA 발표를 참조하세요.

LangChain 및를 사용해야 하는 경우 LangGraph

LangChain 및 LangGraph는 다음과 같은 자율 에이전트 시나리오에 특히 적합합니다.

- 자율 의사 결정을 위해 정교한 오케스트레이션이 필요한 복잡한 다단계 추론 워크플로
- 다양한 자율 기능을 위해 사전 구축된 구성 요소 및 통합으로 구성된 대규모 에코시스템에 액세스해야 하는 프로젝트
- 자율 시스템을 구축하려는 기존 Python기반 기계 학습(ML) 인프라 및 전문 지식을 갖춘 팀
- 장기 실행 자율 에이전트 세션에서 복잡한 상태 관리가 필요한 사용 사례

LangChain 및에 대한 구현 접근 방식 LangGraph

LangChain 및는 LangGraph 설명서에 설명된 대로 비즈니스 이해관계자에게 구조화된 구현 접근 방식을 LangGraph 제공합니다. 프레임워크를 통해 조직은 다음을 수행할 수 있습니다.

- 비즈니스 프로세스를 나타내는 정교한 워크플로 그래프를 정의합니다.
- 결정점과 조건부 로직을 사용하여 다단계 추론 패턴을 생성합니다.
- 다양한 데이터 유형을 처리하기 위한 멀티모달 처리 기능을 통합합니다.
- 내장된 검토 및 검증 메커니즘을 통해 품질 관리를 구현합니다.

이 그래프 기반 접근 방식을 통해 비즈니스 팀은 복잡한 의사 결정 프로세스를 자율 워크플로로 모델링할 수 있습니다. 팀은 추론 프로세스의 각 단계와 결정 경로를 감사할 수 있는 능력을 명확하게 파악할 수 있습니다.

LangChain 및의 실제 예 LangGraph

Vodafone는 LangChain 엔터프라이즈 사례 연구에 설명된 대로 데이터 엔지니어링 및 운영 워크플로를 개선하기 위해 LangChain (및 LangGraph)를 사용하여 자율 에이전트를 구현했습니다. 이들은 자연어 상호 작용을 통해 성능 지표를 자율적으로 모니터링하고, 문서 시스템에서 정보를 검색하고, 실행 가능한 인사이트를 제공하는 내부 AI 어시스턴트를 구축했습니다.

이 Vodafone 구현에서는 LangChain 모듈식 문서 로더, 벡터 통합 및 여러 LLMs(OpenAI,LLaMA#3 및 Gemini)에 대한 지원을 사용하여 이러한 파이프라인을 신속하게 프로토타입화하고 벤치마킹합니다. 그런 다음 모듈식 하위 에이전트를 배포하여 다중 에이전트 오케스트레이션을 구성하는 LangGraph 데 사용됩니다. 이러한 에이전트는 수집, 처리, 요약 및 추론 작업을 수행합니다.는 APIs를 클라우드 시스템에 LangGraph 통합했습니다.

CrewAI

CrewAI는 자율 다중 에이전트 오케스트레이션에 중점을 둔 오픈 소스 프레임워크로, GitHub에서 사용할 수 있습니다. 사람의 개입 없이 복잡한 작업을 해결하기 위해 협업하는 특수 자율 에이전트 팀을 만드는 구조화된 접근 방식을 제공합니다.는 역할 기반 조정 및 작업 위임을 CrewAI 강조합니다.

의 주요 기능 CrewAI

CrewAI는 다음과 같은 주요 기능을 제공합니다.

- 역할 기반 에이전트 설계 - 자율 에이전트는 전문 지식을 지원하기 위해 특정 역할, 목표 및 백 스토리로 정의됩니다. 자세한 내용은 CrewAI 설명서의 효과적인 에이전트 생성을 참조하세요.
- 작업 위임 - 기능에 따라 적절한 에이전트에게 작업을 자율적으로 할당하는 기본 제공 메커니즘입니다. 자세한 내용은 CrewAI 설명서의 작업을 참조하세요.
- 에이전트 공동 작업 - 인적 중재 없이 자율적인 에이전트 간 통신 및 지식 공유를 위한 프레임워크입니다. 자세한 내용은 CrewAI 설명서의 협업을 참조하세요.
- 프로세스 관리 - 순차적 및 병렬 자율 작업 실행을 위한 구조화된 워크플로입니다. 자세한 내용은 CrewAI 설명서의 프로세스를 참조하세요.
- 파운데이션 모델 선택 - Anthropic Claude, Amazon Bedrock의 Amazon Nova 모델(Premier, Pro, Lite, Micro) 등 다양한 파운데이션 모델을 지원하여 다양한 자율 추론 작업에 맞게 최적화합니다. 자세한 내용은 CrewAI 설명서의 LLMs을 참조하세요.
- LLM API 통합 - Amazon Bedrock, OpenAI및 로컬 모델 배포를 포함한 여러 LLM 서비스 인터페이스와의 유연한 통합. 자세한 내용은 CrewAI 설명서의 공급자 구성 예제를 참조하세요.
- 다중 모달 지원 - 포괄적인 자율 에이전트 상호 작용을 위해 텍스트, 이미지 및 기타 모달리티를 처리할 수 있는 새로운 기능입니다. 자세한 내용은 CrewAI 설명서의 다중 모달 에이전트 사용을 참조하세요.

CrewAI(을)를 사용해야 하는 경우

CrewAI는 다음과 같은 자율 에이전트 시나리오에 특히 적합합니다.

- 특화된 역할 기반 전문 지식이 자율적으로 작동하는 데 도움이 되는 복잡한 문제
- 여러 자율 에이전트 간의 명시적 협업이 필요한 프로젝트
- 팀 기반 문제 분해로 자율 문제 해결이 개선되는 사용 사례
- 다양한 자율 에이전트 역할 간에 문제를 명확하게 분리해야 하는 시나리오

에 대한 구현 접근 방식 CrewAI

CrewAI는 CrewAI 설명서의 시작하기에 설명된 대로 비즈니스 이해관계자에게 AI 에이전트 팀의 역할 기반 구현 접근 방식을 제공합니다. 프레임워크를 통해 조직은 다음을 수행할 수 있습니다.

- 특정 역할, 목표 및 전문 영역을 가진 특수 자율 에이전트를 정의합니다.
- 특수 기능에 따라 에이전트에게 작업을 할당합니다.
- 작업 간에 명확한 종속성을 설정하여 구조화된 워크플로를 생성합니다.
- 여러 에이전트 간의 협업을 조정하여 복잡한 문제를 해결합니다.

이 역할 기반 접근 방식은 인적 팀 구조를 반영하므로 비즈니스 리더가 직관적으로 이해하고 구현할 수 있습니다. 조직은 인적 팀의 운영 방식과 마찬가지로 비즈니스 목표를 달성하기 위해 협업하는 전문 지식을 갖춘 자율 팀을 만들 수 있습니다. 그러나 자율 팀은 사람의 개입 없이 지속적으로 작업할 수 있습니다.

의 실제 예 CrewAI

AWS 는 CrewAI 발표된 사례 연구에 설명된 대로 Amazon Bedrock과 통합된 CrewAI를 사용하여 자율 다중 에이전트 시스템을 구현 AWS 했으며 안전한 공급업체 중립 프레임워크를 CrewAI 개발했습니다. CrewAI 오픈 소스 "flows-and-crews" 아키텍처는 Amazon Bedrock 파운데이션 모델, 메모리 시스템 및 규정 준수 가드레일과 원활하게 통합됩니다.

구현의 주요 요소는 다음과 같습니다.

- 블루프린트 및 오픈 소싱 - CrewAI 에이전트를 Amazon Bedrock 모델 및 관찰성 도구에 매핑하는 참조 설계를 CrewAI 릴리스 AWS 했습니다. https://aws.amazon.com/blogs/machine-learning/build-agentic-systems-with-crewai-and-amazon-bedrock/ 또한 다중 에이전트 AWS 보안 감사 팀, 코드 현대화 흐름, 소비자 패키지 제품(CPG) 백오피스 자동화와 같은 예시 시스템을 출시했습니다.
- 관찰성 스택 통합 -이 솔루션은 Amazon CloudWatch, AgentOps및를 사용하여 모니터링을 내장하므로 개념 증명에서 프로덕션으로 추적성 및 디버깅이 LangFuse가능합니다.
- 입증된 투자 수익률(ROI) - 초기 파일럿은 대규모 코드 현대화 프로젝트의 경우 70#% 더 빠른 실행, CPG 백오피스 흐름의 경우 처리 시간 약 90#% 단축이라는 주요 개선 사항을 보여줍니다.

AutoGen

AutoGen는에서 처음 릴리스한 오픈 소스 프레임워크입니다Microsoft.는 대화형 및 협업 자율 AI 에이전트를 활성화하는 데 중점을 AutoGen 둡니다. 복잡한 자율 워크플로를 위해 에이전트 간의 비동기식

이벤트 기반 상호 작용에 중점을 두고 다중 에이전트 시스템을 구축하기 위한 유연한 아키텍처를 제공합니다.

의 주요 기능 AutoGen

AutoGen는 다음과 같은 주요 기능을 제공합니다.

- 대화 에이전트 - 자율 에이전트 간의 자연어 대화를 중심으로 구축되어 대화를 통해 정교한 추론을 가능하게 합니다. 자세한 내용은 AutoGen 설명서의 다중 에이전트 대화 프레임워크를 참조하세요.
- 비동기 아키텍처 - 비차단 자율 에이전트 상호 작용을 위한 이벤트 기반 설계로 복잡한 병렬 워크플로를 지원합니다. 자세한 내용은 AutoGen 설명서의 비동기 채팅 시퀀스에서 여러 작업 해결을 참조하세요.
- Human-in-the-loop 필요한 경우 자율 에이전트 워크플로에 대한 선택적 인적 참여를 강력하게 지원합니다. 자세한 내용은 AutoGen 설명서의 에이전트에서 인적 피드백 허용을 참조하세요.
- 코드 생성 및 실행 - 코드를 작성하고 실행할 수 있는 코드 중심 자율 에이전트를 위한 특수 기능입니다. 자세한 내용은 AutoGen 설명서의 코드 실행을 참조하세요.
- 사용자 지정 가능한 동작 - 다양한 사용 사례를 위한 유연한 자율 에이전트 구성 및 대화 제어. 자세한 내용은 AutoGen 설명서의 agentchat.conversable_agent를 참조하세요.
- 파운데이션 모델 선택 - Anthropic Claude, Amazon Bedrock의 Amazon Nova 모델(Premier, Pro, Lite, Micro) 등 다양한 파운데이션 모델과 다양한 자율 추론 기능을 지원합니다. 자세한 내용은 AutoGen 설명서의 LLM 구성을 참조하세요.
- LLM API 통합 - Amazon Bedrock, 및 OpenAI를 포함한 여러 LLM 서비스 인터페이스에 대한 표준화된 구성입니다Azure OpenAI. 자세한 내용은 AutoGen API 참조의 oai.openai_utils를 참조하세요.
- 멀티모달 처리 - 텍스트 및 이미지 처리를 지원하여 풍부한 멀티모달 자율 에이전트 상호 작용을 활성화합니다. 자세한 내용은 AutoGen 설명서의 의 Engaging with Multimodal Models: GPT-4VAutoGen를 참조하세요.

AutoGen(을)를 사용해야 하는 경우

AutoGen는 다음과 같은 자율 에이전트 시나리오에 특히 적합합니다.

- 복잡한 추론을 위해 자율 에이전트 간에 자연스러운 대화 흐름이 필요한 애플리케이션
- 완전 자율 운영과 선택적 인적 감독 기능이 모두 필요한 프로젝트
- 사람의 개입 없이 자율 코드 생성, 실행 및 디버깅과 관련된 사용 사례
- 유연하고 비동기적인 자율 에이전트 통신 패턴이 필요한 시나리오

에 대한 구현 접근 방식 AutoGen

AutoGen는 AutoGen 설명서의 시작하기에 설명된 대로 비즈니스 이해관계자에게 대화형 구현 접근 방식을 제공합니다. 프레임워크를 통해 조직은 다음을 수행할 수 있습니다.

- 자연어 대화를 통해 통신하는 자율 에이전트를 생성합니다.
- 여러 에이전트 간에 비동기식 이벤트 기반 상호 작용을 구현합니다.
- 필요한 경우 완전 자율 운영과 선택적 인적 감독을 결합합니다.
- 대화를 통해 협업하는 다양한 비즈니스 기능을 위한 전문 에이전트를 개발합니다.

이 대화형 접근 방식을 사용하면 자율 시스템의 추론이 투명하고 비즈니스 사용자가 액세스할 수 있습니다. 의사 결정자는 에이전트 간의 대화를 관찰하여 결론에 도달하는 방법을 이해하고 사람의 판단이 필요할 때 선택적으로 대화에 참여할 수 있습니다.

의 실제 예 AutoGen

Magentic-One는 Microsoft AI Frontiers 블로그에 설명된 대로 다양한 환경에서 복잡한 다단계 작업을 자율적으로 해결하도록 설계된 오픈 소스 일반 다중 에이전트 시스템입니다. 핵심에는 상위 수준의 목표를 분해하고 구조화된 원장을 사용하여 진행 상황을 추적하는 Orchestrator 에이전트가 있습니다. 이 에이전트는 특수 에이전트(예: WebSurfer, FileSurfer, 및 CoderComputerTerminal)에게 하위 작업을 위임하고 필요한 경우 다시 계획하여 동적으로 조정합니다.

시스템은 AutoGen 프레임워크를 기반으로 구축되었으며 모델에 구애받지 않으며 기본값은 GPT-4o입니다. 작업별 튜닝 WebArena없이 GAIA, 및 AssistantBench와 같은 벤치마크 전반에서 최첨단 성능을 달성할 수 있습니다. 또한 AutoGenBench 제안 사항을 통해 모듈식 확장성과 엄격한 평가를 지원합니다.

LlamaIndex

LlamaIndex는 대규모 언어 모델(LLMs)을 외부 데이터 소스와 연결하여 정교한 검색 증강 생성(RAG) 및 에이전트 AI 애플리케이션을 지원하도록 특별히 설계된 데이터 프레임워크입니다. 프레임워크는 지식 기반 AI 솔루션의 time-to-production 단축하는 에이전트 시스템, 사용자 지정 오케스트레이션 패턴 및 시스템 통합을 위한 추상화 및 가속화된 개발 워크플로를 제공합니다.

의 주요 기능 LlamaIndex

LlamaIndex는 엔터프라이즈 에이전트 AI 애플리케이션에 특히 적합한 포괄적인 기능 세트를 제공합니다.

- 데이터 중심 아키텍처 - PDFs, Microsoft Word 문서, 스프레드시트 등 100개 이상의 데이터 형식에서 정보를 수집, 인덱싱 및 검색하는 데 유용합니다. 프레임워크는 엔터프라이즈 데이터를 AI 에이전트에 최적화된 쿼리 가능한 지식 기반으로 변환합니다. 자세한 내용은 LlamaIndex 설명서를 참조하세요.
- 프로덕션 지원 배포 - LlamaIndex는를 통해 오픈 소스 프레임워크와 관리형 서비스를 모두 LlamaCloud제공하여 보안 제어, 확장성, 관찰성 통합, 배포 유연성 등의 엔터프라이즈급 기능을 제공합니다. 자세한 내용은 LlamaIndex 프레임워크 설명서를 참조하세요.
- 고급 문서 처리 - 복잡한 레이아웃, 중첩 테이블, 다중 모달 콘텐츠 및 수기 메모를 처리하는 문서 구문 분석, 추출, 인덱싱 및 검색 기능을 LlamaCloud 제공합니다. 이 정교한 구문 분석을 통해 에이전트는 차트, 다이어그램 및 복잡한 형식이 포함된 실제 엔터프라이즈 문서를 효과적으로 사용할 수 있습니다. 자세한 내용은 LlamaCloud 설명서를 참조하세요.
- 워크플로 오케스트레이션 -는 다단계 에이전트 시스템을 구축하기 위한 이벤트 기반 비동기 우선 오케스트레이션 엔진을 LlamaAgents 제공합니다. 워크플로는 루프, 병렬 실행, 조건부 분기 및 상태 저장 재개를 비롯한 복잡한 패턴을 지원하므로 정교한 에이전트 상호 작용에 적합합니다. 자세한 내용은 LlamaIndex 워크플로 설명서를 참조하세요.
- 에이전트 검색 기능 - 각 쿼리에 대한 최상의 검색 전략을 지능적으로 결정하는 하이브리드 검색, 의미 체계 검색 및 자동 라우팅을 포함한 고급 검색 모드입니다. 프레임워크는 정확도 향상을 위해 순위를 다시 매겨 여러 지식 기반에서 복합 검색을 지원합니다. 자세한 내용은 LlamaIndex RAG 설명서를 참조하세요.
- 관찰성 및 평가 -는 다양한 관찰성 및 평가 도구와 LlamaIndex 통합됩니다. 이 통합 기능을 사용하면 애플리케이션을 추적 및 디버깅하고, 성능을 평가하고, 비용을 모니터링할 수 있습니다. 자세한 내용은 추적 및 디버깅 및 평가 LlamaIndex 설명서를 참조하세요.

LlamaIndex(을)를 사용해야 하는 경우

LlamaIndex는 데이터 집약적인 워크플로 및 지식 관리를 강조하는 에이전트 AI 시나리오에 특히 적합합니다.

- 에이전트가 계약, 보고서, 매뉴얼 및 규제 제출과 같은 대량의 엔터프라이즈 문서를 처리, 분석 및 추출해야 하는 문서가 많은 애플리케이션
- 조직이 광범위한 인프라 관리 오버헤드 없이 문서 중심 에이전트를 빠르게 구축하고 배포하려는 프로덕션 시나리오에 대한 신속한 프로토타입 생성
- 특히 테이블, 이미지 및 구조화된 데이터가 포함된 복잡한 다중 모달 문서로 작업할 때 검색 정확도 및 컨텍스트 관련성을 우선시하는 RAG 우선 아키텍처

- 구문 분석, 분석, 요약 및 규정 준수 확인과 같은 문서 처리의 다양한 측면에 전문 에이전트가 필요한 다중 에이전트 문서 워크플로

에 대한 구현 접근 방식 LlamaIndex

LlamaIndex는 다양한 구현 접근 방식을 수용하는 하위 수준 빌딩 블록과 상위 수준 추상화를 모두 제공합니다.

- LlamaIndex 상위 수준 APIs. 이 접근 방식을 사용하면 에이전트 AI를 처음 사용하는 비즈니스 팀과 개발자가 LlamaIndex 액세스할 수 있습니다.
- SharePoint, Amazon Simple Storage Service(Amazon S3), 데이터베이스, API 등 널리 사용되는 엔터프라이즈 시스템을 위한 LlamaHub를 통한 엔터프라이즈 통합. APIs 이 접근 방식을 사용하면 기존 데이터 인프라와 원활하게 통합할 수 있습니다.
- 최대 제어를 위한 오픈 소스 자체 호스팅 배포 또는 운영 오버헤드 및 엔터프라이즈 기능 감소를 위한 LlamaCloud 관리형 서비스 간의 유연한 배포 옵션.
- 애플리케이션은 간단한 쿼리 엔진으로 시작하여 요구 사항이 증가함에 따라 에이전트 기능, 다중 에이전트 오케스트레이션 및 복잡한 워크플로를 점진적으로 추가할 수 있습니다.

의 실제 예 LlamaIndex

이 예제는 항공 탐색 및 운영 솔루션을 전문으로 하는 항공 우주 회사의 자회사에 중점을 둡니다. 이들은 조정되지 않은 AI 챗봇 평가판을 파일럿하는 것과 관련된 증가하는 문제를 해결해야 합니다. 이 시도로 인해 조직 전체에서 반복적인 작업, 긴 개발 주기, 규정 준수 장애물 및 격리된 구현이 이루어졌습니다.

에이전트 생성을 훨씬 더 효율적으로 만드는 LlamaIndex 오픈 소스 프레임워크를 기반으로 구축된 재사용 가능한 템플릿 기반 솔루션인 통합 에이전트 프레임워크를 개발했습니다. 체인 지향 프레임워크와 그래프 기반 프레임워크를 비교했습니다. 궁극적으로 유연한 설계, 모듈식 구성 요소, 프로덕션 지원 오케스트레이션 제어라는 세 가지 중요한 이점을 선택LlamaIndex했습니다.

플랫폼은 에이전트 개발 및 배포 시간을 512시간에서 64시간으로 87% 단축합니다. 이러한 감소는 팀이 약 50줄의 코드와 JSON 구성 파일을 사용하여 에이전트를 구축할 수 있도록 함으로써 달성되었습니다. 팀은 보안, 규정 준수 및 권한 있는 시스템 액세스가 내장된 통합 프레임워크를 활용했습니다. 자세한 내용은 LlamaIndex고객 사례 연구를 참조하세요.

에이전트 AI 프레임워크 비교

자율 에이전트 개발을 위한 에이전트 AI 프레임워크를 선택할 때 각 옵션이 특정 요구 사항에 어떻게 부합하는지 고려하세요. 기술 역량뿐만 아니라 팀 전문 지식, 기존 인프라, 장기 유지 관리 요구 사항을 포함한 조직의 적합성도 고려합니다. 많은 조직이 자율 AI 에코시스템의 다양한 구성 요소에 여러 프레임워크를 활용하여 하이브리드 접근 방식의 이점을 누릴 수 있습니다.

다음 표에서는 주요 기술 차원에서 각 프레임워크의 성숙도 수준(가장 강력함, 강력함, 적절함 또는 약함)을 비교합니다. 각 프레임워크에 대해 테이블에는 프로덕션 배포 옵션 및 학습 곡선 복잡성에 대한 정보도 포함되어 있습니다.

프레임워크	AWS 통합	자율 다중 에이전트 지원	자율 워크플로 복잡성	멀티모달 기능	파운데이션 모델 선택	LLM API 통합	프로덕션 배포	학습 곡선
AutoGen	약함	강함	강함	적절한	적절한	강함	직접 수행(DIY)	가파름
CrewAI	약함	강함	적절한	약함	적절한	적절한	DIY	보통
LangChain / LangGraph	적절한	강함	가장 강력함	가장 강력함	가장 강력함	가장 강력함	플랫폼 또는 DIY	가파름
LlamaIndex	적절한	적절한	강함	적절한	강함	강함	플랫폼 또는 DIY	보통
Strands Agents	가장 강력함	강함	가장 강력함	강함	강함	가장 강력함	DIY	보통

에이전트 AI 프레임워크 선택 시 고려 사항

자율 에이전트를 개발할 때 다음 주요 요소를 고려하세요.

- AWS 인프라 통합 -에 많이 투자한 조직은 자율 워크플로를 AWS 서비스 위해 Strands Agents와의 기본 통합을 최대한 활용할 AWS 수 있습니다. 자세한 내용은 AWS 주간 라운드업(AWS 블로그)을 참조하세요.
- 파운데이션 모델 선택 - 자율 에이전트의 추론 요구 사항에 따라 선호하는 파운데이션 모델(예: Amazon Bedrock 또는 Anthropic Claude의 Amazon Nova 모델)에 가장 적합한 지원을 제공하는 프레임워크를 고려합니다. 자세한 내용은 Anthropic 웹 사이트의 효과적인 에이전트 구축을 참조하세요.
- LLM API 통합 - 프로덕션 배포를 위해 선호하는 대규모 언어 모델(LLM) 서비스 인터페이스(예: Amazon Bedrock 또는 OpenAI)와의 통합을 기반으로 프레임워크를 평가합니다. 자세한 내용은 설명서의 Strands Agents 모델 인터페이스를 참조하세요.
- 멀티모달 요구 사항 - 텍스트, 이미지 및 음성을 처리해야 하는 자율 에이전트의 경우 각 프레임워크의 멀티모달 기능을 고려하세요. 자세한 내용은 LangChain 설명서의 다중 양식을 참조하세요.
- 자율 워크플로 복잡성 - 정교한 상태 관리를 갖춘 보다 복잡한 자율 워크플로는 고급 상태 시스템 기능을 선호할 수 있습니다. LangGraph
- 자율 팀 협업 - 전문 에이전트 간의 명시적 역할 기반 자율 협업이 필요한 프로젝트는의 팀 지향 아키텍처의 이점을 누릴 수 있습니다CrewAI.
- 자율 개발 패러다임 - 자율 에이전트를 위한 대화형 비동기 패턴을 선호하는 팀은의 이벤트 기반 아키텍처를 선호할 수 있습니다AutoGen.
- 관리형 또는 코드 기반 접근 방식 - 코딩을 최소화하면서 완전 관리형 환경을 원하는 조직은 Amazon Bedrock Agents를 고려해야 합니다. 심층 사용자 지정이 필요한 조직은 특정 자율 에이전트 요구 사항에 더 잘 맞는 특수 기능을 갖춘 Strands Agents 또는 기타 프레임워크를 선호할 수 있습니다.
- 자율 시스템을 위한 프로덕션 준비 - 프로덕션 자율 에이전트를 위한 배포 옵션, 모니터링 기능 및 엔터프라이즈 기능을 고려합니다.

플랫폼

에이전트 AI 플랫폼은 프로덕션급 에이전트 시스템을 배포, 확장 및 관리하는 데 필요한 기본 런타임, 오케스트레이션 및 통합 계층을 제공합니다. 프레임워크는 에이전트가 구축되는 방식을 정의하고 에이전트가 통신하는 방식을 제어하는 프로토콜을 정의합니다. 플랫폼은 이러한 에이전트가 대규모로 안전하게 운영, 협업 및 발전하는 환경을 제공합니다.

에이전트 플랫폼은 모델 실행, 컨텍스트 관리, 도구 통합, 관찰성 및 거버넌스 기능을 통합 환경으로 결합합니다. 이러한 플랫폼을 통해 조직은 실험에서 엔터프라이즈 규모의 배포로 전환할 수 있습니다.

이 단원에서는 다음을 수행합니다.

- 플랫폼이 중요한 이유
- 에이전트 AI 플랫폼의 유형
- 플랫폼 선택 고려 사항
- Amazon Bedrock 에이전트
- Amazon Bedrock AgentCore

플랫폼이 중요한 이유

에이전트 AI 플랫폼은 프로덕션 환경에서 자율 시스템을 운영하려는 조직에 매우 중요합니다. 다음과 같은 기능을 제공합니다.

- 에이전트 호스팅, 규모 조정 및 조정을 위한 런타임 오케스트레이션을 제공합니다.
- 다중 에이전트 워크플로에서 상태, 컨텍스트 및 메모리를 관리합니다.
- 엔터프라이즈 표준에 맞는 보안, 자격 증명 및 거버넌스 제어를 제공합니다.
- 표준 APIs 또는 프로토콜을 통해 도구 에코시스템 및 외부 시스템과 통합합니다.
- 에이전트 상호 작용 및 이벤트 흐름 전반에서 관찰성 및 감사 가능성을 활성화합니다.
- 교차 모델 상호 운용성을 지원하여 에이전트가 단일 환경에서 여러 파운데이션 모델을 사용할 수 있습니다.

이러한 기능을 통해 개별 에이전트는 엔터프라이즈 및 규제 경계 내에서 안정적으로 작동할 수 있는 조정된 적응형 시스템으로 전환됩니다.

에이전트 AI 플랫폼의 유형

에이전트 AI 플랫폼은 일반적으로 다음 범주 중 하나 이상에 속합니다.

- 관리형 에이전트 - 완전 관리형 플랫폼은 기본 제공 인프라, 메모리 및 오케스트레이션 기능을 제공합니다. 운영 오버헤드를 줄이고 프로덕션 시간을 단축합니다.
- 오픈 소스 오케스트레이션 - 오픈 소스 에이전트 플랫폼은 사용자 지정 가능한 환경 또는 온프레미스 배포를 선호하는 조직에 유연성과 투명성을 제공합니다.
- 하이브리드 엔터프라이즈 - 하이브리드 플랫폼은 관리형 구성 요소와 자체 호스팅 구성 요소를 통합하여 클라우드 관리형 서비스의 확장성과 엔터프라이즈 시스템의 제어를 결합합니다.

플랫폼 선택 고려 사항

에이전트 AI 플랫폼을 선택하거나 설계할 때 조직은 다음을 고려해야 합니다.

- 통합 깊이 - 플랫폼이 기존 데이터 소스, 도구 및 프로토콜과 얼마나 잘 통합되는지 평가합니다.
- 확장성 - 플랫폼이 자율 워크로드 및 다중 에이전트 협업을 지원하도록 동적으로 확장할 수 있는지 확인합니다.
- 보안 및 규정 준수 - 조직 및 리전 요구 사항을 기준으로 데이터 프라이버시, 암호화 및 거버넌스 기능을 평가합니다.
- 확장성 - 시간이 지남에 따라 새로운 도구, 모델 또는 에이전트를 추가할 수 있는 모듈식 아키텍처가 있는 플랫폼을 선택합니다.
- 관찰성 - 에이전트 상호 작용에 대한 자세한 원격 측정, 추적성 및 감사 로그를 제공하는 플랫폼을 선호합니다.
- 비용 효율성 - 서버리스 또는 사용량 기반 모델을 고려하여 가변 워크로드에 대한 비용을 최적화합니다.

Amazon Bedrock 에이전트

Amazon Bedrock Agents는 애플리케이션에서 자율 에이전트를 구축하고 구성할 수 있는 완전 관리형 서비스입니다. 파운데이션 모델, 데이터 소스, 소프트웨어 애플리케이션 및 사용자 대화 간의 상호 작용을 조정할 수 있습니다. 에이전트를 생성하는 간소화된 접근 방식에서는 용량을 프로비저닝하거나 인프라를 관리하거나 사용자 지정 코드를 작성할 필요가 없습니다.

Amazon Bedrock Agents의 주요 기능

Amazon Bedrock Agents에는 다음과 같은 주요 기능이 포함되어 있습니다.

- 완전 관리형 서비스 - 용량을 프로비저닝하거나 기본 시스템을 관리할 필요 없이 인프라 관리를 완료합니다. 자세한 내용은 Amazon Bedrock 설명서의 AI 에이전트를 사용하여 애플리케이션의 태스크 자동화를 참조하세요.
- API 기반 개발 - 모델, 지침, 도구 및 구성 파라미터를 지정하여 간단한 API 직접 호출을 통해 에이전트를 정의하고 실행합니다. 자세한 내용은 Amazon Bedrock 설명서의 에이전트 수동 생성 및 구성을 참조하세요.
- 작업 그룹 - API 스키마를 사용하여 작업 그룹을 생성하여 에이전트가 수행할 수 있는 특정 작업을 정의합니다. 자세한 내용은 Amazon Bedrock 설명서에서 작업 그룹을 사용하여 에이전트가 수행할 작업 정의를 참조하세요.
- 지식 기반 통합 - Amazon Bedrock 지식 기반에 원활하게 연결하여 조직의 데이터로 에이전트 응답을 강화합니다. 자세한 내용은 Amazon Bedrock 설명서의 지식 기반을 사용하여 에이전트에 대한 Augment 응답 생성을 참조하세요.
- 고급 프롬프트 템플릿 - 사전 처리, 오케스트레이션, 지식 기반 응답 생성 및 사후 처리를 위한 프롬프트 템플릿을 통해 에이전트 동작을 사용자 지정합니다. 자세한 내용은 Amazon Bedrock 설명서의 Amazon Bedrock의 고급 프롬프트 템플릿을 사용하여 에이전트의 정확도 향상을 참조하세요.
- 추적 및 관찰성 - 내장된 추적 기능을 사용하여 에이전트의 step-by-step 추론 프로세스를 추적합니다. 자세한 내용은 Amazon Bedrock 설명서에서 추적을 사용하여 에이전트의 step-by-step 추론 프로세스 추적을 참조하세요.
- 버전 관리 및 별칭 - 에이전트의 여러 버전을 생성하고 제어된 롤아웃을 위해 별칭을 통해 배포합니다. 자세한 내용은 Amazon Bedrock 설명서의 애플리케이션에서 Amazon Bedrock 에이전트 배포 및 사용을 참조하세요.

Amazon Bedrock Agents를 사용해야 하는 경우

Amazon Bedrock Agents는 다음과 같은 자율 에이전트 시나리오에 특히 적합합니다.

- 인프라를 관리하지 않고 에이전트를 구축하고 배포하기 위한 완전 관리형 환경을 원하는 조직
- 코드 대신 구성을 통해 에이전트를 신속하게 개발하고 배포해야 하는 프로젝트
- 지식 기반 및 가드레일과 같은 다른 Amazon Bedrock 기능과 긴밀하게 통합하여 이점을 얻을 수 있는 사용 사례

- 사내 리소스가 없어 에이전트를 처음부터 구축할 수 있지만 프로덕션에 바로 사용할 수 있는 자율 기능이 필요한 팀

Amazon Bedrock Agents의 구현 접근 방식

Amazon Bedrock Agents는 비즈니스 이해관계자를 위한 구성 기반 구현 접근 방식을 제공합니다. 이 서비스를 통해 조직은 다음을 수행할 수 있습니다.

- 복잡한 코드를 작성하지 않고 AWS Management Console 또는 API 호출을 통해 에이전트를 정의합니다.
- 에이전트가 수행할 수 있는 APIs 및 작업을 지정하는 작업 그룹을 생성합니다.
- 지식 기반을 연결하여 에이전트에 도메인별 정보를 제공합니다.
- 시각적 인터페이스를 통해 에이전트 동작을 테스트하고 반복합니다.

이 관리형 접근 방식을 통해 비즈니스 팀은 AI 모델 개발 또는 인프라 관리에 대한 심층적인 기술 전문 지식 없이 자율 에이전트를 신속하게 개발하고 배포할 수 있습니다.

Amazon Bedrock Agents의 실제 예제

이 AWS 블로그 게시물에 설명된 재무 운영(FinOps) 솔루션은 Amazon Bedrock 다중 에이전트 프레임워크를 사용하여 AI 기반 클라우드 비용 관리 도우미를 생성합니다. 비용 효율적인 Amazon Nova 파운데이션 모델은 중앙 FinOps 감독자 에이전트가 특수 에이전트에게 작업을 위임하는 솔루션을 지원합니다. 이러한 에이전트는를 사용하여 AWS 지출 데이터를 가져오고 분석 AWS Cost Explorer 하며를 사용하여 비용 절감 권장 사항을 생성합니다 AWS Trusted Advisor.

시스템에는에서 호스팅되는 프런트엔드인 Amazon Cognito를 통한 보안 사용자 액세스 AWS Amplify와 실시간 분석 및 예측을 위한 AWS Lambda 작업 그룹이 포함되어 있습니다. 재무 팀은 “2025년 2월에 내 비용은 얼마였나요?”와 같은 자연어 쿼리를 요청할 수 있습니다. 시스템은를 사용하여 배포된 확장 가능한 서버리스 아키텍처 내에서 세부 분석, 최적화 제안 및 예측으로 응답합니다 AWS CloudFormation.

Amazon Bedrock AgentCore

Amazon Bedrock AgentCore는 프레임워크, 모델 또는 프로토콜을 사용하여 확장성이 뛰어난 에이전트를 안전하게 구축, 배포 및 운영하는 에이전트 플랫폼입니다. AgentCore를 사용하면 인프라 관리 없이 다음 작업을 모두 수행할 수 있습니다.

- 에이전트를 더 빠르게 구축합니다.
- 에이전트가 도구 및 데이터 전반에 걸쳐 조치를 취할 수 있도록 합니다.
- 지연 시간이 짧고 런타임이 확장된 에이전트를 안전하게 실행합니다.
- 프로덕션 중인 에이전트를 모니터링합니다.

AgentCore는 특수 에이전트 인프라를 구축하는 데 따른 차별화되지 않은 부담을 제거하여 에이전트의 프로덕션 속도를 높일 수 있습니다. 서비스는 함께 또는 독립적으로 사용할 수 있으며 , CrewAI, 및를 포함한 모든 프레임워크LangGraphLlamaIndex와 호환됩니다Strands Agents. 또한 AgentCore는 Amazon Bedrock 내부 또는 외부에서 사용할 수 있는 모든 파운데이션 모델과 호환되므로 최고의 유연성을 제공합니다.

AgentCore는 다음과 같은 여러 주요 서비스로 구성됩니다.

- Amazon Bedrock AgentCore 런타임 - AI 에이전트 또는 도구를 배포하고 실행하는 데 필요한 인프라를 관리할 필요 없이 에이전트를 호스팅하고 실행할 수 있는 안전하고 확장 가능한 서버리스 환경을 제공합니다.
- Amazon Bedrock AgentCore 메모리 - 관리형 메모리 시스템을 제공하므로 에이전트는 즉각적이고 장기적인 지식을 유지하여 보다 개인화되고 일관된 대화를 위해 상호 작용의 컨텍스트를 유지할 수 있습니다.
- Amazon Bedrock AgentCore Gateway - 에이전트에 적합한 도구를 생성, 보안 및 찾는 프로세스를 간소화합니다. 개발자는 AgentCore GatewayAPIs, Lambda 함수 및 기존 서비스를 모델 컨텍스트 프로토콜(MCP) 호환 도구로 변환하고 에이전트가 사용할 수 있도록 할 수 있습니다.
- Amazon Bedrock AgentCore 자격 증명 - AI 에이전트 개발을 가속화하는 안전하고 확장 가능한 에이전트 자격 증명 및 액세스 관리 서비스를 제공합니다. AgentCore Identity를 사용하면 에이전트에게 검증 가능한 고유 자격 증명을 할당하여 세분화된 액세스 제어와 엔터프라이즈 시스템과의 에이전트 기반 안전한 상호 작용을 지원할 수 있습니다.
- Amazon Bedrock AgentCore 기본 제공 도구 - 기본 제공 도구를 사용하여 개발 및 테스트 워크플로를 개선할 수 있습니다. 이러한 도구를 사용하여 애플리케이션과 효과적으로 상호 작용하여 AI 에이전트가 샌드박스 환경에서 코드를 안전하게 작성하고 실행할 수 있습니다. 브라우저 도구를 사용하여 AI 에이전트가 대규모로 웹 사이트와 상호 작용할 수 있도록 합니다.
- Amazon Bedrock AgentCore 관찰성 - 로깅 및 모니터링 기능을 제공하여 에이전트의 성능 및 동작에 대한 실시간 가시성을 제공하여 디버깅 및 최적화를 용이하게 합니다.

AgentCore의 주요 기능

AgentCore에는 다음과 같은 주요 기능이 포함되어 있습니다.

- 완전 관리형 및 확장 가능 - AgentCore는 완전 관리형 서비스이므로가 기본 인프라 및 유지 관리를 AWS 처리합니다. 또한 확장 가능하므로 에이전트의 기능을 사용자 지정하고 개선할 수 있습니다. 자세한 내용은 AgentCore 설명서의 AgentCore 런타임 시작하기를 참조하세요. AgentCore
- 장기 및 단기 메모리 - 에이전트에게 현재 대화 및 장기 지식의 컨텍스트를 재현할 수 있는 메모리 시스템을 제공하여 보다 개인화되고 관련성이 높은 상호 작용을 제공합니다. 자세한 내용은 AgentCore 설명서의 AgentCore 메모리 시작하기를 참조하세요. AgentCore
- 간소화된 도구 개발 및 통합 - 에이전트가 안전한 단일 엔드포인트를 통해 도구를 검색하고 사용할 수 있습니다. 몇 줄의 코드만으로 기존 엔터프라이즈 리소스를 에이전트 지원 도구로 빠르게 전환하여 개발자가 고유한 기능을 구축하는 데 집중할 수 있습니다. 자세한 내용은 AgentCore 설명서의 AgentCore Gateway 시작하기를 참조하세요. AgentCore
- 안전하고 확장 가능한 인프라 - AgentCore는 에이전트를 배포하고 운영할 수 있는 안전하고 확장 가능한 환경을 제공합니다. 여기에는 자격 증명 및 액세스 관리, 데이터 암호화 및 네트워크 보안을 위한 기능이 포함됩니다. 자세한 내용은 AgentCore 설명서의 AgentCore 자격 증명 시작하기를 참조하세요. AgentCore
- 다양한 도구와의 통합 - 에이전트를 코드 인터프리터 및 AgentCore 기본 제공 도구를 사용하여 구축할 수 있는 브라우저 도구를 비롯한 다양한 도구와 통합할 수 있습니다. 자세한 내용은 AgentCore 설명서의 AgentCore 코드 인터프리터 시작하기 및 AgentCore AgentCore 브라우저 시작하기를 참조하세요.
- 포괄적인 관찰성 및 모니터링 - 포괄적인 도구를 사용하여 에이전트를 심층적으로 파악하여 프로덕션 환경에서 성능을 추적, 디버깅 및 모니터링할 수 있습니다. 에이전트의 전체 실행 경로를 시각화하여 추론을 감사하고 실패를 해결합니다. 실시간 대시보드와 표준화된 원격 측정 데이터를 사용하여 주요 운영 지표를 추적할 수 있습니다. 자세한 내용은 AgentCore 설명서의 Amazon Bedrock AgentCore 리소스에 관찰성 추가를 참조하세요. AgentCore

AgentCore를 사용해야 하는 경우

AgentCore는 다음과 같은 자율 에이전트 시나리오에 특히 적합합니다.

- 인프라, 보안, 내장 도구, 관찰성 및 규모 조정을 처리하는 완전 관리형 서비스를 통해 개발을 가속화하고 운영 오버헤드를 줄이려는 조직
- 함께 또는 독립적으로 작동하고 CrewAI 또는와 같은 모든 프레임워크 LangGraph및 모든 소스의 모든 파운데이션 모델과 호환되는 모듈식 서비스로 유연성이 필요한 프로젝트

- 상황 정보를 유지하고 과거 상호 작용에서 학습하여 개인화되고 관련된 응답을 제공해야 하는 상태 저장 대화 에이전트가 필요한 사용 사례
- 다양한 애플리케이션, 데이터 소스 및 APIs와의 간단한 통합을 통해 복잡한 작업을 수행할 수 있는 에이전트

AgentCore의 구현 접근 방식

AgentCore는 오픈 소스 또는 사용자 지정 에이전트 프레임워크를 사용하여 구축된 AI 에이전트를 개념 증명에서 프로덕션으로 이동하려는 조직을 위해 설계되었습니다. AgentCore를 사용하면 조직은 다음을 수행할 수 있습니다.

- end-to-end 보안 및 규정 준수를 위한 세션 격리와 내장된 ID 및 액세스 관리를 통해 모든 프레임워크 및 모델을 지원하는 서버리스 인프라에 에이전트를 안전하게 배포합니다. 스타터 툴킷을 사용하여 주요 에이전트 프레임워크를 위한 AgentCore 런타임 에이전트를 빠르게 생성합니다.
- 컨텍스트 보존을 위한 영구 메모리를 통합하여 에이전트를 개선하고 AgentCore Gateway를 통한 도구 개발 및 통합을 간소화합니다. 고급 워크플로를 위해 내장 브라우저 도구 및 코드 인터프리터를 활용합니다.
- Amazon CloudWatch Application Insights 및 로 구동되는 관찰성 대시보드를 사용하여 프로덕션 환경에서 AI 에이전트를 추적, 디버깅 및 모니터링하고 AgentCore 리소스(런타임, 메모리, 게이트웨이 및 도구)의 주요 지표를 OpenTelemetry추적합니다.
- 모든 에이전트 프레임워크 및 모델 공급자와 함께 또는 독립적으로 완전 관리형 모듈식 서비스, 구성 가능한 블록을 사용하여 배포 및 혁신을 가속화합니다. 이러한 유연성을 통해 조직은 프로토타입에서 프로덕션으로 더 빠르게 이동할 수 있습니다.

이러한 관리형 접근 방식을 통해 조직은 규모에 관계없이 엔터프라이즈급 AI 에이전트 및 다중 에이전트 시스템을 빠르고 안전하게 구축, 배포 및 실행할 수 있습니다.

AgentCore의 실제 예

AWS 는 라틴 아메리카 최대 은행 중 하나가 다년간 AI/ML을 사용하여 개인화되고 안전한 디지털 뱅킹 경험을 제공하고 있음을 관찰했습니다. 은행은 AgentCore를 사용하여 고객에게 직관적인 상호 작용, 향상된 보안 및 향상된 자동화를 제공하여 에이전트 AI 서비스를 확장하고 있습니다. CTO에 따르면 AgentCore는 대규모로 고객 약정을 충족하기 위한 노력을 지원할 것으로 예상됩니다. AgentCore는 재무 규정 준수를 보장하는 데 도움이 되는 동시에 개발자에게 에이전트를 구축하고 관리할 수 있는 도구와 유연성을 제공합니다.

프로토콜

AI 에이전트는 다른 에이전트 및 서비스와 상호 작용하려면 표준화된 통신 프로토콜이 필요합니다. 에이전트 아키텍처를 구현하는 조직은 상호 운용성, 공급업체 독립성, 미래 대비 투자와 관련된 상당한 문제에 직면합니다.

이 섹션에서는 유연성과 상호 운용성을 극대화하는 개방형 표준에 중점을 두고 agent-to-agent 프로토콜 환경을 탐색하는 데 도움이 됩니다. (agent-to-tool 프로토콜에 대한 자세한 내용은이 가이드 뒷부분의 도구 통합 전략을 참조하세요.)

이 섹션에서는 2024년에에서 처음 개발한 개방형 표준인 모델 컨텍스트 프로토콜(MCP)Anthropic을 강조합니다. 오늘날는 프로토콜의 개발 및 구현에 기여하여 MCP를 AWS 적극적으로 지원합니다. AWS 는 , LangGraph CrewAI및를 비롯한 주요 오픈 소스 에이전트 프레임워크와 협력하여 프로토콜에서 에이전트 간 통신의 미래를 LlamaIndex형성합니다. 자세한 내용은 Open Protocols for Agent Interoperability Part 1: Inter-Agent Communication on MCP(AWS 블로그)를 참조하세요.

이 단원에서는 다음을 수행합니다.

- 프로토콜 선택이 중요한 이유
- Agent-to-agent 프로토콜
- 에이전트 프로토콜 선택
- 에이전트 프로토콜의 구현 전략
- MCP 시작하기
- ???

프로토콜 선택이 중요한 이유

프로토콜 선택은 AI 에이전트 아키텍처를 구축하고 발전시키는 방법을 근본적으로 형성합니다. 에이전트 프레임워크 간의 이동성을 지원하는 프로토콜을 선택하면 다양한 에이전트 시스템과 워크플로를 결합하여 특정 요구 사항을 충족할 수 있는 유연성을 얻을 수 있습니다.

개방형 프로토콜을 사용하면 여러 프레임워크에서 에이전트를 통합할 수 있습니다. 예를 들어 MCP 또는 Agent2Agent(A2A) 프로토콜과 같은 공통 프로토콜을 통해 Strands Agents통신하여를 사용하여 프로덕션 시스템을 구현하고 LangChain 신속한 프로토타입을 만드는 데를 사용합니다. 이러한 유연성은 특정 AI 제공업체에 대한 종속성을 줄이고, 기존 시스템과의 통합을 간소화하며, 시간이 지남에 따라 에이전트 기능을 향상시킬 수 있습니다.

또한 잘 설계된 프로토콜은 에이전트 에코시스템 전반에서 인증 및 권한 부여를 위한 일관된 보안 패턴을 설정합니다. 가장 중요한 것은 프로토콜 이식성을 통해 새로운 에이전트 프레임워크와 기능이 등장할 때 이를 채택할 수 있는 자유를 유지할 수 있다는 것입니다. 개방형 프로토콜을 선택하면 타사 시스템과의 상호 운용성을 유지하면서 에이전트 개발에 대한 투자를 보호할 수 있습니다.

개방형 프로토콜의 장점

자체 확장을 구현하거나 사용자 지정 에이전트 시스템을 구축할 때 개방형 프로토콜은 다음과 같은 강력한 이점을 제공합니다.

- 설명서 및 투명성 - 일반적으로 포괄적인 설명서 및 투명한 구현 제공
- 커뮤니티 지원 - 문제 해결 및 모범 사례를 위해 더 광범위한 개발자 커뮤니티에 액세스
- 상호 운용성 보장 - 확장이 다양한 구현에서 작동하도록 더 잘 보장합니다.
- 향후 호환성 - 변경 사항 또는 사용 중단 위험 감소
- 개발에 미치는 영향 - 프로토콜 진화에 기여할 수 있는 기회

Agent-to-agent 프로토콜

다음 표에는 여러 에이전트가 협업하고, 작업을 위임하고, 정보를 공유할 수 있는 에이전트 프로토콜의 개요가 나와 있습니다.

프로토콜	에 적합	고려 사항
MCP 에이전트 간 통신	유연한 에이전트 협업 패턴을 찾는 조직	• agent-to-agent 통신을 위한 기존 기반을 기반으로 하는 에서 제안 AWS 한 모델 컨텍스트 프로토콜(MCP)의 확장 • OAuth 기반 보안으로 원활한 에이전트 협업 지원
A2A 프로토콜	교차 플랫폼 에이전트 에코시스템	• 에서 지원 Google • MCP에 비해 채택이 더 제한적인 최신 표준

프로토콜 옵션 중 결정

agent-to-agent 통신을 구현할 때는 특정 통신 요구 사항을 적절한 프로토콜 기능과 일치시킵니다. 서로 다른 상호 작용 패턴에는 서로 다른 프로토콜 기능이 필요합니다. 다음 표에서는 일반적인 통신 패턴을 간략하게 설명하고 각 시나리오에 가장 적합한 프로토콜 선택을 권장합니다.

패턴	설명	이상적인 프로토콜 선택
간단한 요청 및 응답	에이전트 간의 일회성 상호 작용	상태 비저장 흐름이 있는 MCP
상태 저장 대화	컨텍스트를 사용한 지속적인 대화	세션 관리가 포함된 MCP
다중 에이전트 공동 작업	여러 에이전트 간의 복잡한 상호 작용	MCP 에이전트 간 또는 AutoGen
팀 기반 워크플로	역할이 정의된 계층적 에이전트 팀	MCP 에이전트 간, CrewAI또는 AutoGen

커뮤니케이션 패턴 외에도 여러 가지 기술적 및 조직적 요인이 프로토콜 선택에 영향을 미칠 수 있습니다. 다음 표에는 특정 구현 요구 사항에 가장 잘 맞는 프로토콜을 평가하는 데 도움이 되는 주요 고려 사항이 요약되어 있습니다.

고려 사항	설명	예제
보안 모델	인증 및 권한 부여 요구 사항	MCP의 OAuth 2.0
배포 환경	에이전트가 실행되고 통신할 위치	분산 또는 단일 시스템
에코시스템 호환성	기존 에이전트 프레임워크와 통합	LangChain 또는 Strands Agents
확장성 요구 사항	에이전트 상호 작용의 예상 증가	MCP의 스트리밍 기능

에이전트 프로토콜 선택

프로덕션 에이전트 시스템을 구축하는 대부분의 조직에 대해 모델 컨텍스트 프로토콜(MCP)은 agent-to-agent 통신을 위한 가장 포괄적이고 잘 지원되는 기반을 제공합니다. MCP는 AWS 및 오픈 소스 커뮤니티의 적극적인 개발 기여를 통해 이점을 얻습니다.

올바른 에이전트 프로토콜 선택은 에이전트 AI를 효과적으로 구현하려는 조직에 중요합니다. 고려 사항은 조직 컨텍스트에 따라 다릅니다.

에이전트 프로토콜 선택 고려 사항

조직은 에이전트 AI 시스템에 대한 프로토콜을 선택할 때 다음 모범 사례를 고려해야 합니다.

- 개방형 표준 우선 순위 지정 - 조직은 MCP와 같은 개방형 프로토콜을 채택하여 장기적인 상호 운용성, 확장성을 보장하고 공급업체 잠금 위험을 줄여야 합니다.
- 속도와 유연성의 균형 - 스타트업과 얼리 어답터는 신속한 개발을 위해 잘 지원되는 독점 프로토콜로 시작할 수 있지만 시스템이 성숙해지면 개방형 표준에 대한 마이그레이션 경로를 정의해야 합니다.
- 추상화 계층 구현 - 기업은 프로토콜 추상화를 구현하여 마이그레이션을 간소화하고 하이브리드 채택을 지원하며 미래에 대비한 통합 전략을 수립해야 합니다.
- 보안 및 규정 준수 강조 - 규제 대상 산업의 조직은 거버넌스 및 규정 준수 요구 사항을 충족하기 위해 강력한 인증, 암호화 및 감사 기능을 갖춘 프로토콜을 선택해야 합니다.
- 에코시스템 성숙도 평가 - 모든 조직은 지속 가능성을 보장하고 기술적 부채를 최소화하기 위해 각 프로토콜의 상태, 채택 및 커뮤니티 지원을 평가해야 합니다.
- 표준 개발 참여 - 조직은 표준 기관 또는 오픈 소스 커뮤니티에 참여하여 프로토콜 진화를 형성하고 모범 사례에 영향을 주어야 합니다.
- 데이터 주권 고려 - 정부 및 규제 부문은 프로토콜 선택이 배포 리전 전반의 데이터 레지던시 및 주권 요구 사항에 부합하는지 확인해야 합니다.
- 관리형 서비스 활용 - 가능하면 에이전트 프로토콜의 관리형 또는 서버리스 구현을 사용하여 운영 복잡성을 줄이고 배포를 가속화합니다.

에이전트 프로토콜의 구현 전략

조직 전체에서 에이전트 프로토콜을 효과적으로 구현하려면 다음 전략적 단계를 고려하세요.

1. 표준 조정으로 시작 - 가능한 경우 설정된 개방형 프로토콜을 채택합니다.
2. 추상화 계층 생성 - 시스템과 특정 프로토콜 간에 어댑터를 구현합니다.

3. 개방형 표준에 기여 - 프로토콜 개발 커뮤니티에 참여합니다.
4. 프로토콜 진화 모니터링 - 새로운 표준 및 업데이트에 대한 정보를 계속 확인하세요.
5. 정기적으로 상호 운용성 테스트 - 구현이 호환되는지 확인합니다.

MCP 시작하기

AWS 는 프로토콜의 개발 및 구현에 기여하여 모델 컨텍스트 프로토콜(MCP)을 적극적으로 지원합니다. AWS 는 LangGraph, CrewAI및를 비롯한 주요 오픈 소스 에이전트 프레임워크와 협력하여 프로토콜에서 에이전트 간 통신의 미래를 LlamaIndex형성합니다.

에이전트 아키텍처에서 MCP를 구현하려면 다음 작업을 수행합니다.

1. Strands Agents SDK와 같은 프레임워크에서 MCP 구현을 살펴봅니다.
2. 모델 컨텍스트 프로토콜 기술 설명서를 검토합니다.
3. 에이전트 상호 운용성에 대한 오픈 프로토콜 1부: MCP에서의 에이전트 간 통신(AWS 블로그)을 읽고 에이전트 상호 운용성에 대해 알아봅니다.
4. MCP 커뮤니티에 가입하여 프로토콜의 진화에 영향을 줍니다.

MCP는 에이전트가 외부 데이터 및 서비스와 상호 작용할 수 있도록 하는 통신 계층을 제공하며 에이전트가 다른 에이전트와 상호 작용할 수 있도록 하는 데도 사용할 수 있습니다. 프로토콜의 스트리밍 가능한 HTTP 전송 구현은 개발자에게 휠을 재창조할 필요 없이 포괄적인 상호 작용 패턴 세트를 제공합니다. 이러한 패턴은 상태 비저장 요청/응답 흐름과 영구 IDs를 사용한 상태 저장 세션 관리를 모두 지원합니다.

MCP와 같은 개방형 프로토콜을 채택하면 AI 기술이 발전함에 따라 유연하고 상호 운용 가능하며 적응 가능한 에이전트 시스템을 구축할 수 있습니다. agent-to-tool 프로토콜 구현에 대한 자세한 내용은이 가이드 뒷부분의 도구 통합 전략을 참조하세요.

A2A 시작하기

Agent2Agent(A2A) 프로토콜을 사용하면 공유 시맨틱 계층을 통해 에이전트 간에 분산된 공동 작업을 수행할 수 있습니다. 중앙 오케스트레이터를 통해 모든 작업을 라우팅하는 대신 A2A를 사용하면 에이전트가 경량 JSON 기반 프로토콜을 사용하여 서로를 검색하고, 기능을 알리고, 작업을 협상하고, 컨텍스트를 공유할 수 있습니다. 각 에이전트는 기능 매니페스트를 게시합니다.

다음 예제에서는 검색 및 작업 협상을 활성화하기 위해 에이전트가 지원하는 작업, 필수 입력 및 운영 메타데이터를 알리는 간소화된 A2A 기능 매니페스트를 보여줍니다.

```
{
  "can": ["summarize.text", "extract.keywords"],
  "needs": ["document.input"],
  "meta": { "version": "1.0.3", "latencyMs": 120 }
}
```

이 모델을 사용하면 동적 기능 매칭, 작업 중 위임 및 조직 간 협업이 가능합니다. 에이전트는 작업을 자체 구성하고, 임시 작업 그룹을 구성하고, 새로운 기능이 시스템에 들어오거나 나갈 때 적응할 수 있습니다.

A2A는 다음과 같은 간단한 상태 비저장 요청부터 다단계 협상 세션에 이르기까지 다양한 상호 작용을 지원합니다.

- 지연 시간이 짧은 협업을 위한 직접 peer-to-peer 메시징
- 에이전트가 가장 적합한 피어를 선택하는 의미 체계 작업 협상
- 기능 기반 검색, 새로운 인력 분할 지원
- 상태 저장 다단계 상호 작용을 위한 세션 고정

조직은 A2A와 같은 개방형 에이전트 네이티브 프로토콜을 채택하여 모듈식이고 상호 운용 가능하며 경계 간 협업이 가능한 AI 시스템을 생성합니다. A2A는 에이전트 에코시스템이 유연성을 유지하고 엄격한 오케스트레이션 계층이나 사전 결합 없이도 새로운 에이전트, 팀 또는 외부 시스템이 도입됨에 따라 발전할 수 있도록 합니다.

에이전트 아키텍처에서 A2A 프로토콜을 구현하려면 다음 작업을 수행합니다.

1. A2A 프로토콜 사양 검토 - 최신 버전의 Agent2Agent(A2A) 프로토콜 사양을 읽고 기능 매니페스트, 협상 흐름 및 에이전트 핸드셰이크가 작동하는 방법을 알아봅니다.
2. A2A-compatible 런타임 살펴보기 A2A-style 기능 매니페스트 및 peer-to-peer 협상을 지원하는 Strands Agents SDK 또는 사용자 지정 런타임 계층과 같은 프레임워크를 평가합니다.
3. 에이전트에 대한 기능 매니페스트 구현 - 각 에이전트의 `can`, `needs`및 `meta` 필드를 정의하여 검색, 매치메이킹 및 의도 수준 협업을 활성화합니다.
4. A2A 협상 패턴 실험 - 요청 제안 수락 루프, 구조화된 기능 쿼리 또는 gossip 기반 검색을 사용하여 에이전트가 작업을 누가 처리해야 하는지에 대한 이유를 이해합니다.
5. 혼합 인프라 환경에서 A2A 테스트 - A2A 피어 협상을 Amazon EventBridge를 AWS 통해 네이티브 이벤트 라우팅과 결합하여 하이브리드 조정 패턴을 평가합니다.
6. A2A 커뮤니티 가입 - 오픈 작업 그룹에 참여하여 확장, 보안 권장 사항 및 공급업체 간 상호 운용성 개선 사항을 최신 상태로 유지하고 프로토콜 개발에 기여합니다.

도구

AI 에이전트는 외부 도구, APIs 및 데이터 소스와 상호 작용하여 유용한 작업을 수행하여 가치를 제공합니다. 올바른 도구 통합 전략은 에이전트의 기능, 보안 태세 및 장기적인 유연성에 직접적인 영향을 미칩니다.

이 섹션에서는 자유와 유연성을 극대화하는 개방형 표준에 중점을 두고 도구 통합 환경을 탐색하는 데 도움이 됩니다. 이 섹션에서는 도구 통합을 위한 모델 컨텍스트 프로토콜(MCP)을 강조 표시하고 에이전트 워크플로를 개선하는 프레임워크별 도구 및 특수 메타 도구를 검토합니다.

이 단원에서는 다음을 수행합니다.

- 도구 범주
- 프로토콜 기반 도구
- 프레임워크 네이티브 도구
- 메타 도구
- 도구 통합 전략
- 도구 통합을 위한 보안 모범 사례

도구 범주

빌드 에이전트 시스템에는 세 가지 주요 범주의 도구가 포함됩니다.

프로토콜 기반 도구

프로토콜 기반 도구는 agent-to-tool 통신을 위해 표준화된 프로토콜을 사용합니다.

- MCP 도구 - 로컬 및 원격 실행 옵션을 모두 사용하여 프레임워크에서 작동하는 개방형 표준 도구입니다.
- OpenAI 함수 호출 - OpenAI 모델별 독점 도구입니다.
- Anthropic 도구 - Anthropic Claude 모델 전용 도구를 호출하는 OpenAI 함수의 변형입니다.

프레임워크 네이티브 도구

프레임워크 네이티브 도구는 특정 에이전트 프레임워크에 직접 구축됩니다.

- Strands Agents 도구 - Strands Agents 프레임워크와 관련된 가볍고 quick-to-implement 도구입니다.
- LangChain 도구 Python- LangChain 에코시스템과 긴밀하게 통합된 기반 도구입니다.
- LlamaIndex 도구 - 내에서 데이터 검색 및 처리에 최적화된 도구입니다LlamaIndex.

메타 도구

메타 도구는 외부 작업을 직접 수행하지 않고 에이전트 워크플로를 개선합니다.

- 워크플로 도구 - 에이전트 실행 흐름, 분기 로직 및 상태 관리를 관리합니다.
- 에이전트 그래프 도구 - 복잡한 워크플로에서 여러 에이전트를 조정합니다.
- 메모리 도구 - 에이전트 세션 전반에 걸쳐 정보를 지속적으로 저장하고 검색할 수 있습니다.
- 반영 도구 - 에이전트가 자신의 성과를 분석하고 개선할 수 있습니다.

프로토콜 기반 도구

프로토콜 기반 도구를 고려할 때 모델 컨텍스트 프로토콜(MCP)은 도구 통합을 위한 가장 포괄적이고 유연한 기반을 제공합니다. AWS 에이전트 상호 운용성에 대한 오픈 소스 블로그 게시물에 명시된 대로 AWS 는 MCP를 전략적 프로토콜로 받아들여 개발에 적극적으로 기여했습니다.

다음 표에서는 MCP 도구 배포 옵션을 설명합니다.

배포 모델	설명	에 적합	구현
로컬 stdio 기반	에이전트와 동일한 프로세스에서 실행되는 도구	개발, 테스트 및 간단한 도구	네트워크 오버헤드 없이 빠르게 구현
로컬 서버 전송 이벤트(SSE) 기반	도구는 로컬에서 실행되지만 HTTP를 통해 통신합니다.	우려 사항이 분리된 더 복잡한 로컬 도구	격리는 개선되었지만 지연 시간은 여전히 짧음
원격 HTTP 스트리밍 가능	원격 서버에서 실행되는 도구	프로덕션 환경 및 공유 도구	확장 가능하고 중앙에서 관리됨

공식 MCP SDKs는 MCP 도구를 빌드하는 데 사용할 수 있습니다.

- Python SDK - 전체 프로토콜 지원을 통한 포괄적인 구현
- TypeScript SDK - 웹 애플리케이션을 위한 JavaScript/TypeScript 구현
- Java SDK - 엔터프라이즈 애플리케이션을 위한 Java 구현

이러한 SDKs 프로토콜 사양의 일관된 구현과 함께 선호하는 언어로 MCP 호환 도구를 생성하기 위한 구성 요소를 제공합니다.

또한 AWS 는 Strands Agents SDK에서 MCP를 구현했습니다. Strands Agents SDK는 MCP 호환 도구를 생성하고 사용하는 간단한 방법을 제공합니다. 포괄적인 설명서는 Strands Agents GitHub 리포지토리에서 확인할 수 있습니다. 더 간단한 사용 사례를 위해 또는 Strands Agents 프레임워크 외부에서 작업할 때 공식 MCP SDKs 프로토콜을 여러 언어로 직접 구현합니다.

MCP 도구의 보안 기능

MCP 도구의 보안 기능은 다음과 같습니다.

- OAuth 2.0/2.1 인증 – 업계 표준 인증
- 권한 범위 지정 - 도구에 대한 세분화된 액세스 제어
- 도구 기능 검색 - 사용 가능한 도구의 동적 검색
- 구조화된 오류 처리 - 일관된 오류 패턴

MCP 도구 시작하기

도구 통합을 위한 MCP를 구현하려면 다음 작업을 수행합니다.

1. 프로덕션 지원 MCP 구현을 위한 Strands Agents SDK를 살펴보세요.
2. MCP 기술 설명서를 검토하여 핵심 개념을 이해합니다.
3. 이 AWS 오픈 소스 블로그 게시물에 설명된 실제 예제를 사용합니다.
4. 원격 도구로 진행하기 전에 간단한 로컬 도구로 시작합니다.
5. MCP 커뮤니티에 가입하여 프로토콜의 진화에 영향을 줍니다.

AgentCore 게이트웨이 살펴보기

Amazon Bedrock AgentCore Gateway는 개발자가 MCP 도구 및 기타 대상 엔드포인트를 대규모로 구축, 배포, 검색 및 연결할 수 있는 쉽고 안전한 방법을 제공합니다. AgentCore Gateway를 사용하면 개

발자APIs, AWS Lambda 함수 및 기존 서비스를 MCP 호환 도구로 변환할 수 있습니다. 그런 다음 몇 줄의 코드만으로 에이전트가 AgentCore Gateway 엔드포인트를 통해 이러한 도구를 사용할 수 있도록 할 수 있습니다. AgentCore Gateway는 OpenAPI, Smithy및 Lambda를 입력 유형으로 지원하며 완전 관리형 서비스에서 포괄적인 수신 인증과 송신 인증을 모두 제공하는 유일한 솔루션입니다.

프레임워크 네이티브 도구

모델 컨텍스트 프로토콜(MCP)은 가장 유연한 기반을 제공하지만 프레임워크 네이티브 도구는 특정 사용 사례에 대한 이점을 제공합니다.

Strands Agents SDK는 간단한 작업을 위해 최소한의 오버헤드가 필요한 경량 설계를 특징으로 하는 Python기반 도구를 제공합니다. 이를 통해 빠르게 구현할 수 있으며 개발자는 몇 줄의 코드만으로 도구를 생성할 수 있습니다. 또한 Strands Agents 프레임워크 내에서 원활하게 작동하도록 긴밀하게 통합됩니다.

다음 예제에서는를 사용하여 간단한 날씨 도구를 생성하는 방법을 보여줍니다Strands Agents. 개발자는 코드 오버헤드를 최소화하면서 Python 함수를 에이전트 액세스 가능 도구로 빠르게 변환하고 함수의 문서스트링에서 적절한 문서를 자동으로 생성할 수 있습니다.

```
#Example of a simple Strands native tool

@tool

def weather(location: str) -> str:

"""Get the current weather for a location""" #

Implementation here

return f"The weather in {location} is sunny."
```

빠른 프로토타입 생성 또는 간단한 사용 사례의 경우 프레임워크 네이티브 도구를 사용하면 개발을 가속화할 수 있습니다. 그러나 프로덕션 시스템의 경우 MCP 도구는 프레임워크 네이티브 도구보다 더 나은 상호 운용성과 향후 유연성을 제공합니다.

다음 표에는 다른 프레임워크별 도구에 대한 개요가 나와 있습니다.

프레임워크	도구 유형	장점	고려 사항
AutoGen	함수 정의	강력한 다중 에이전트 지원	Microsoft 에코시스템

LangChain	Python 클래스	사전 구축된 도구의 대규모 에코시스템	프레임워크 잠금
LlamaIndex	Python 함수	데이터 작업에 최적화	로 제한됨 LlamaIndex

메타 도구

메타 도구는 외부 시스템과 직접 상호 작용하지 않습니다. 대신 에이전트 패턴을 구현하여 에이전트 기능을 개선합니다. 이 섹션에서는 워크플로, 에이전트 그래프 및 메모리 메타 도구에 대해 설명합니다.

워크플로 메타 도구

워크플로 메타 도구는 에이전트 실행 흐름을 관리합니다.

- 상태 관리 - 여러 에이전트 상호 작용에서 컨텍스트 유지
- 분기 로직 - 조건부 실행 경로 활성화
- 재시도 메커니즘 - 정교한 재시도 전략으로 실패 처리

워크플로 메타 도구가 포함된 프레임워크 예제에는 LangGraph 및 Strands Agents 워크플로 기능이 포함됩니다.

에이전트 그래프 메타 도구

에이전트 그래프 메타 도구는 함께 작업하는 여러 에이전트를 조정합니다.

- 작업 위임 - 특수 에이전트에게 하위 작업 할당
- 결과 집계 - 여러 에이전트의 출력 결합
- 충돌 해결 - 에이전트 간의 불일치 해결

AutoGen 및와 같은 프레임워크는 에이전트 그래프 조정을 CrewAI 전문으로 합니다.

메모리 메타 도구

메모리 메타 도구는 영구 스토리지 및 검색을 제공합니다.

- 대화 기록 - 세션 간 컨텍스트 유지

- 지식 기반 - 도메인별 정보 저장 및 검색
- 벡터 스토어 - 의미 체계 검색 기능 활성화

MCP의 리소스 시스템은 다양한 에이전트 프레임워크에서 작동하는 메모리 메타 도구를 구현하는 표준화된 방법을 제공합니다.

도구 통합 전략

선택한 도구 통합 전략은 에이전트가 수행할 수 있는 작업과 시스템이 얼마나 쉽게 발전할 수 있는지에 직접적인 영향을 미칩니다. 프레임워크 네이티브 도구와 메타 도구를 사용하여 전략적으로 모델 컨텍스트 프로토콜(MCP)과 같은 개방형 프로토콜의 우선 순위를 지정합니다. 이렇게 하면 AI 기술이 발전함에 따라 유연하고 강력한 도구 에코시스템을 구축할 수 있습니다.

도구 통합에 대한 다음과 같은 전략적 접근 방식은 조직의 즉각적인 요구 사항을 충족하면서 유연성을 극대화합니다.

1. MCP를 기반으로 채택 - MCP는 강력한 보안 기능을 갖춘 도구에 에이전트를 연결하는 표준화된 방법을 제공합니다. MCP를 기본 도구 프로토콜로 시작합니다.
   - 여러 에이전트 구현에서 사용할 전략적 도구입니다.
   - 강력한 인증 및 권한 부여가 필요한 보안에 민감한 도구입니다.
   - 프로덕션 환경에서 원격 실행이 필요한 도구입니다.
2. 적절한 경우 프레임워크 네이티브 도구 사용 - 다음을 위한 프레임워크 네이티브 도구를 고려합니다.
   - 초기 개발 중 신속한 프로토타입 생성.
   - 최소한의 보안 요구 사항이 있는 중요하지 않은 간단한 도구입니다.
   - 고유한 기능을 활용하는 프레임워크별 기능입니다.
3. 복잡한 워크플로를 위한 메타 도구 구현 - 메타 도구를 추가하여 에이전트 아키텍처를 개선합니다.
   - 기본 워크플로 패턴으로 간단하게 시작합니다.
   - 사용 사례가 성숙해지면 복잡성을 추가합니다.
   - 에이전트와 메타 도구 간의 인터페이스를 표준화합니다.
4. 진화 계획 - 향후 유연성을 염두에 두고 구축합니다.
   - 구현과 독립적으로 도구 인터페이스를 문서화합니다.
   - 에이전트와 도구 간에 추상화 계층을 생성합니다.
   - 독점 프로토콜에서 개방형 프로토콜로의 마이그레이션 경로를 설정합니다.

도구 통합을 위한 보안 모범 사례

도구 통합은 보안 태세에 직접적인 영향을 미칩니다. 이 섹션에서는 조직에 고려해야 할 모범 사례를 간략하게 설명합니다.

인증 및 권한 부여

다음과 같은 강력한 액세스 제어를 사용합니다.

- OAuth 2.0/2.1 사용 - 원격 도구에 대한 업계 표준 인증을 구현합니다.
- 최소 권한 구현 - 필요한 권한만 도구에 부여합니다.
- 자격 증명 교체 - API 키와 액세스 토큰을 정기적으로 업데이트합니다.

데이터 보호

데이터를 보호하려면 다음 조치를 채택합니다.

- 입력 및 출력 검증 - 모든 도구 상호 작용에 대한 스키마 검증을 구현합니다.
- 민감한 데이터 암호화 - 모든 원격 도구 통신에 TLS를 사용합니다.
- 데이터 최소화 구현 - 필요한 정보만 도구에 전달합니다.

모니터링 및 감사

다음 메커니즘을 사용하여 가시성과 제어를 유지합니다.

- 모든 도구 호출 로깅 - 포괄적인 감사 추적을 유지 관리합니다.
- 이상 모니터링 - 비정상적인 도구 사용 패턴을 감지합니다.
- 속도 제한 구현 - 과도한 도구 호출을 통해 남용을 방지합니다.

모델 컨텍스트 프로토콜(MCP) 보안 모델은 이러한 문제를 포괄적으로 해결합니다. 자세한 내용은 MCP 설명서의 보안 고려 사항을 참조하세요.

결론

에이전트 AI의 환경은 지속적으로 빠르게 발전하여 조직이 지능적인 자율 시스템을 구축할 수 있는 강력한 새로운 방법을 제공합니다. 이 가이드에서는 성공적인 구현을 위한 세 가지 필수 구성 요소, 즉 기반을 제공하는 프레임워크, 환경을 제공하는 플랫폼, 통신을 지원하는 프로토콜, 기능을 확장하는 도구를 살펴보았습니다.

프레임워크가 성숙해지면 상호 운용성 향상, 모델 컨텍스트 프로토콜(MCP)과 같은 프로토콜 관련 표준화, 자율 에이전트를 위한 보다 정교한 오케스트레이션 기능을 기대할 수 있습니다. 오늘날 이러한 프레임워크에 대한 전문 지식을 확립하는 조직은 상당한 비즈니스 가치를 제공하는 점점 더 자율적이고 지능적인 에이전트를 구축할 수 있는 유리한 위치에 서게 될 것입니다.

플랫폼은 에이전트 시스템이 작동하는 실행, 거버넌스 및 수명 주기 환경을 제공합니다. ID, 보안 경계, 관찰성, 메모리 관리, 세션 근거, 도구 및 데이터와의 안전한 상호 작용과 같은 문제를 처리합니다. 환경에서 AWS 는 관리형 에이전트 런타임 및 오케스트레이션 서비스와 같은 플랫폼을 통해 조직은 자율 에이전트 및 에이전트 시스템을 대규모로 배포, 모니터링, 발전 및 관리할 수 있습니다. 플랫폼은 기본 프레임워크를 실제 운영 요구 사항과 연결합니다.

에이전트 프로토콜 선택은 즉각적인 개발 요구 사항과 장기적인 유연성 및 상호 운용성의 균형을 맞추는 전략적 결정을 나타냅니다. 개방형 프로토콜의 우선순위를 정하고 적절한 추상화 계층을 생성하면 조직은 진화하는 기술에 계속 적응하면서 현재 비즈니스 요구 사항을 충족하는 에이전트 시스템을 구축할 수 있습니다.

대부분의 조직에서 MCP는 개방형 표준, 성장하는 에코시스템, agent-to-agent 통신 패턴 지원, 도구 통합 기능으로 인해 강력한 기반을 나타냅니다. AWS 는 MCP 및 Agent2Agent(A2A)를 전략적 프로토콜로 받아들여 Strands Agents SDK와 같은 서비스 전반에 걸쳐 개발 및 구현에 적극적으로 기여했습니다. MCP 또는 A2A를 적절한 프레임워크 네이티브 도구 및 메타 도구와 함께 사용하면 향후 혁신에 적응하면서 즉각적인 가치를 제공하는 에이전트 시스템을 구축할 수 있습니다.

리소스

자율 에이전트 개발 AWS 과 관련된 다음 및 기타 리소스를 사용합니다.

AWS 블로그

- Amazon Bedrock AgentCore 메모리: 컨텍스트 인식 에이전트 구축
- Amazon Bedrock Agents를 사용하여 강력한 생성형 AI 애플리케이션을 빌드하는 모범 사례 - 1부
- Amazon Bedrock Agents를 사용하여 강력한 생성형 AI 애플리케이션을 구축하는 모범 사례 - 2부
- LlamaIndex 및 Amazon Bedrock을 사용하여 강력한 RAG 파이프라인 구축
- Amazon Bedrock AgentCore 관찰성을 사용하여 신뢰할 수 있는 AI 에이전트 구축
- Amazon Bedrock LlamaIndex 및 RAGAS를 사용하여 RAG 응답 평가
- Amazon Bedrock AgentCore 코드 인터프리터 소개
- Amazon Bedrock AgentCore Gateway 소개: 엔터프라이즈 AI 에이전트 도구 개발 혁신
- Amazon Bedrock AgentCore Identity 소개: 대규모 에이전트 AI 보호
- 오픈 소스 AI 에이전트 SDKStrands Agents인 소개
- 에이전트 상호 운용성 1부: MCP에서의 에이전트 간 통신을 위한 개방형 프로토콜
- Amazon Bedrock AgentCore 런타임에서 에이전트와 도구를 안전하게 시작하고 규모 조정
- AWS Transform 대규모로 .NET 애플리케이션을 현대화하기 위한 최초의 에이전트 AI 서비스인 for .NET
- AWS 주간 반올림: Strands Agents

AWS 권장 가이드

- 에서 에이전트 AI 운영 AWS
- 의 에이전트 AI 기반 AWS
- 의 에이전트 AI 패턴 및 워크플로 AWS
- 에서 에이전트 AI를 위한 서버리스 아키텍처 구축 AWS
- 에서 에이전트 AI를 위한 멀티 테넌트 아키텍처 구축 AWS
- 의 에이전트 AI에 대한 보안 AWS
- 에서 증강 생성 옵션 및 아키텍처 검색 AWS

AWS 리소스

- Amazon Bedrock 설명서
- Amazon Bedrock AgentCore 설명서
- Amazon Bedrock AgentCore Starter Toolkit(GitHub 리포지토리)
- Amazon Nova 설명서
- AWS MCP 서버(GitHub 리포지토리)

기타 리소스

- AutoGen 설명서(Microsoft)
- 효과적인 에이전트 구축(Anthropic)
- CrewAI GitHub 리포지토리
- LangChain 설명서
- LangGraph 플랫폼
- LlamaIndex 설명서
- 모델 컨텍스트 프로토콜 설명서
- Strands Agents 설명서
- Strands Agents 도구 개요
- Strands Agents 빠른 시작 안내서

문서 기록

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

변경 사항	설명	날짜
새로운 섹션	플랫폼 섹션 추가	2026년 1월 16일
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. 예: 온프레미스 Oracle 데이터베이스를 Amazon Aurora PostgreSQL 호환 에디션으로 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) for Oracle로 마이그레이션합니다.
- 재구매(드롭 앤드 숍) - 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.
- 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 직접 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 입히기 위한 봇입니다.

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 액세스 권한

예외적인 상황에서 승인된 프로세스를 통해 사용자가 일반적으로 액세스할 권한이 없는에 액세스할 수 AWS 계정 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 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 전송을 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 프롬프팅도 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?을 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 딥이스 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.

I

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(ITSM)

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 자체적으로 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 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 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 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 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 마이그레이션 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 애플리케이션을 현대화하기 위한 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 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 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.

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.

릴리스

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 검색 증강 생성(RAG)이란 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

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 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 암호 또는 사용자 자격 증명과 같이 암호화된 형식으로 저장하는 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 제어는 예방, 감지, 대응, 선제적과 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

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 지표(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 지표를 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 압도하고 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, 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 Many(WORM)를 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 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.