의 에이전트 AI 패턴 및 워크플로 AWS

AWS 권장 가이드

AWS 권장 가이드: 의 에이전트 AI 패턴 및 워크플로 AWS

Table of Contents

의 에이전트 AI 패턴 및 워크플로 AWS

Aaron Sempf 및 Andrew Hooker, Amazon Web Services

2025년 7월(문서 기록)

조직은 대규모 언어 모델(LLMs)과 소프트웨어 에이전트를 채택하여 에이전트 패턴이라는 새로운 아키텍처 원칙을 사용하여 동적 다중 도메인 문제를 해결하고 있습니다. 에이전트 패턴은 여러 컨텍스트에서 목표 지향 AI 에이전트를 설계하고 오케스트레이션하는 데 사용되는 기본 청사진 및 모듈식 구문입니다.

대상 독자

이 가이드는 정적 로직, 심볼 로직 및 결정적 자동화를 넘어 지능형 애플리케이션을 구축하려는 아키텍트, 개발자 및 제품 리더를 대상으로 합니다.

목표

이 가이드는 제어 가능하고 목표에 맞게 조정하면서 자율적으로 작동하는 AI 에이전트 시스템에 대한 설계 프레임워크 및 구현 접근 방식을 제공합니다. 이벤트 기반 아키텍처 패턴을 다양한 에이전트 대안과 연결하여 클라우드 네이티브 아키텍처를 사용하여 프로덕션급 에이전트 시스템을 구축하는 방법을 보여줍니다. 이 가이드에서는 다음 주제에 대해 설명합니다.

- 에이전트 패턴 - 에이전트 패턴은 개별 에이전트의 구조와 동작을 설명하는 재사용 가능한 설계 템플릿입니다. 여기에는 추론 에이전트, 검색 증강 에이전트, 코딩 에이전트, 음성 인터페이스, 워크플로 오케스트레이터 및 협업 다중 에이전트 시스템이 포함됩니다. 각 패턴은 에이전트가 어떻게 인지하고, 추론하고, 행동하고, 학습하고, 매핑되는지 보여줍니다 AWS 서비스.
- LLM 워크플로 - 워크플로는 에이전트가 추론을 위해 LLMs 사용하는 방법에 중점을 둡니다. 프롬프트 전략과 계획 메커니즘을 살펴보고 LLMs 사용하여 텍스트를 생성할 뿐만 아니라 에이전트 루프 내에서 구조화되고 해석 가능하며 신뢰할 수 있는 동작을 유도하는 방법을 간략하게 설명합니다.
- 에이전트 워크플로 패턴 - 워크플로 패턴은 여러 에이전트, 도구 및 환경이 상호 작용하여 자율 시스템을 형성하는 방법을 설명합니다. 여기에는 작업 오케스트레이션, 하위 에이전트 위임, 이벤트 기반 조정, 관찰성 및 제어에 대한 패턴이 포함됩니다. 이러한 측면은 확장 가능하고 구성 가능하며 감사 가능한 AI 아키텍처를 촉진합니다.

이 콘텐츠 시리즈 정보

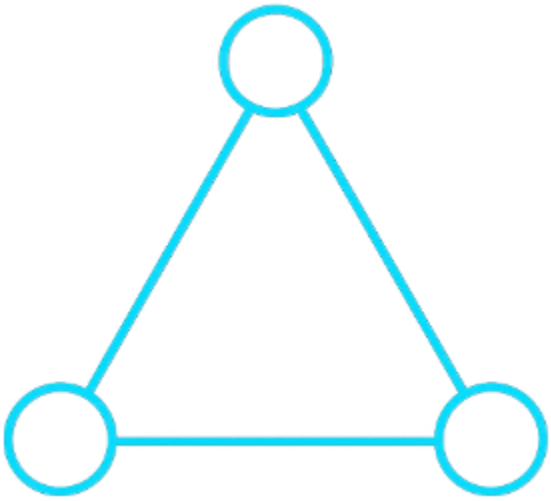
이 가이드는 에이전트 AI on에 대한 시리즈의 일부입니다 AWS. 자세한 내용과이 시리즈의 다른 가이드를 보려면 AWS 권장 가이드 웹 사이트의 에이전트 AI를 참조하세요.

에이전트 패턴

에이전트 패턴은 특정 도메인, 사용 사례 및 복잡성 수준에 맞게 조정할 수 있는 재사용 가능하고 구성 가능한 구성 요소입니다. 그러나 에이전트 시스템은 기존 애플리케이션과 다릅니다. 모든 AI 에이전트 설계의 핵심은 다음 세 가지 기본 원칙을 기반으로 하는 개념 모델입니다.

- 비동기식 - 에이전트는 느슨하게 결합되고 이벤트가 풍부한 환경에서 작동합니다.
- 자율성 - 에이전트는 인간 또는 외부 제어 없이 독립적으로 행동합니다.
- 에이전트 - 에이전트는 특정 목표를 향해 사용자 또는 시스템을 대신하여 목적을 가지고 행동합니다.

다음 다이어그램의 삼각형은 소프트웨어 에이전트의 핵심 구성 요소인 지각, 이유 및 작업을 나타냅니다. 이를 통해 에이전트 시스템은 환경 내에서 관찰하고, 결정을 내리고, 조치를 취할 수 있습니다.



설계상 에이전트 패턴은 AI 시스템을 구축하기 위한 모듈식 설계 언어를 제공하므로 액세스, 운영, 확장 및 프로덕션 준비가 가능합니다. 이러한 시스템을 설계하려면 다음 세 가지 상호 관련된 차원에 주의해야 하며, 이에 대해서는이 가이드의 뒷부분에서 자세히 설명합니다.

이 섹션의 내용

- 음성 및 음성 에이전트
- 워크플로 오케스트레이션 에이전트
- 메모리 증강 에이전트
- 시뮬레이션 및 테스트베드 에이전트
- 관찰자 및 모니터링 에이전트
- 다중 에이전트 공동 작업

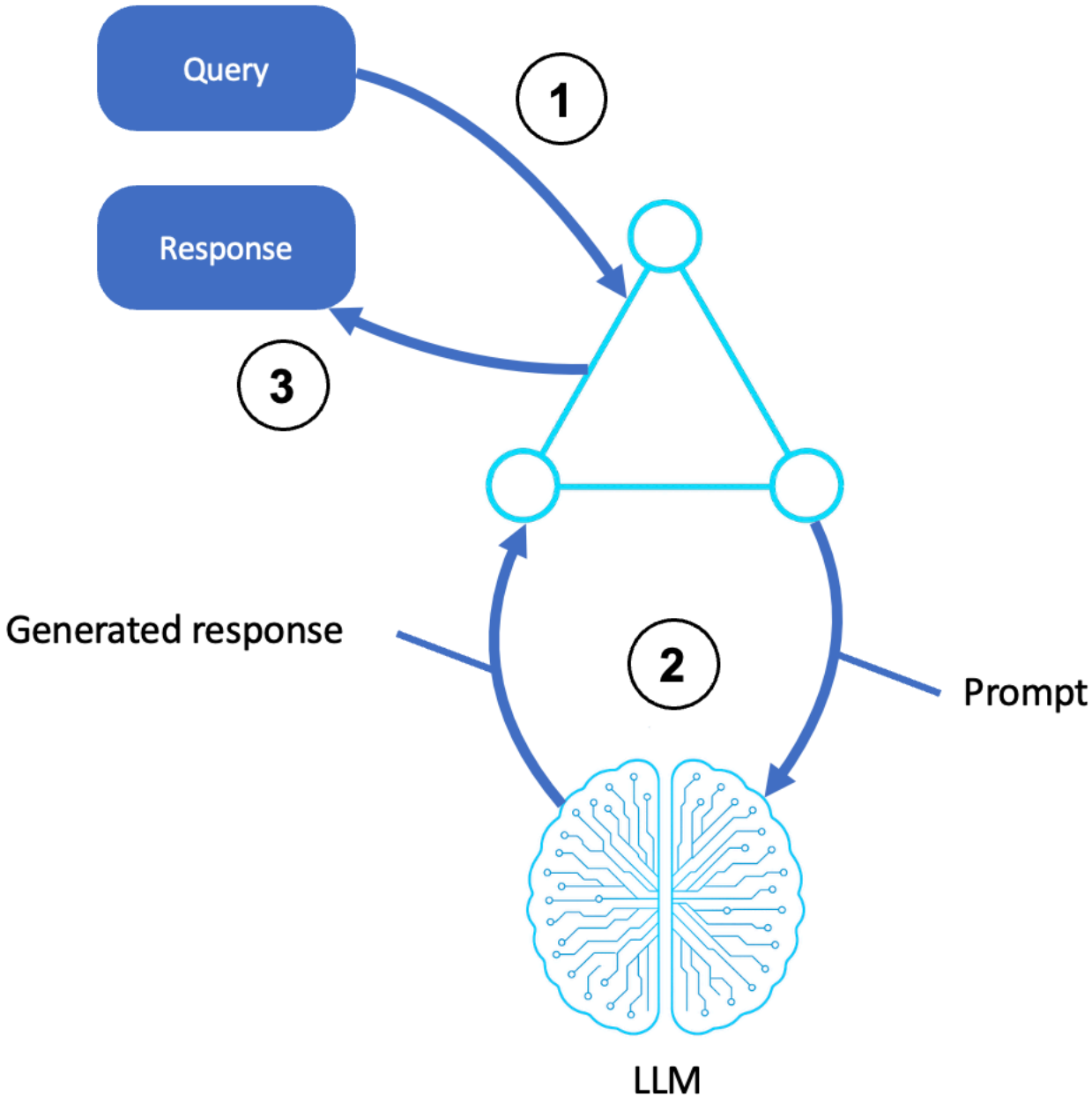
기본 추론 에이전트

기본 추론 에이전트는 쿼리에 대한 응답으로 논리적 추론 또는 의사 결정을 수행하는 가장 간단한 형태의 에이전트 AI입니다. 사용자 또는 시스템의 입력을 수락하고 쿼리를 처리하고 구조화된 프롬프트를 사용하여 응답을 생성합니다.

이 패턴은 주어진 컨텍스트를 기반으로 한 단일 단계 추론, 분류 또는 요약이 필요한 작업에 유용합니다. 메모리, 도구 또는 상태 관리를 사용하지 않으므로 대규모 워크플로에서 상태 비저장, 경량 및 고도로 구성 가능합니다.

Architecture

기본 추론 에이전트의 흐름은 다음 다이어그램에 나와 있습니다.



설명

1. 입력을 수신합니다.
   - 사용자, 시스템 또는 업스트림 에이전트가 쿼리 또는 명령을 제출합니다.
   - 입력은 에이전트 쉘 또는 오케스트레이션 계층으로 전달됩니다.
   - 이 단계에는 모든 사전 처리, 프롬프트 템플릿 지정 및 목표 식별이 포함됩니다.
2. LLM을 호출합니다.

- 에이전트는 쿼리를 구조화된 프롬프트로 변환하여 LLM으로 보냅니다(예: Amazon Bedrock을 통해).
- LLM은 사전 훈련된 지식 및 컨텍스트를 사용하여 프롬프트를 기반으로 응답을 생성합니다.
- 생성된 출력에는 추론 단계(chain-of-thought), 최종 답변 또는 순위 옵션이 포함될 수 있습니다.

3. 응답을 반환합니다.
   - 생성된 출력은 에이전트의 인터페이스로 릴레이됩니다.
   - 여기에는 형식 지정, 사후 처리 또는 API 응답이 포함될 수 있습니다.

기능

- 자연어 또는 구조화된 입력 지원
- 프롬프트 엔지니어링을 사용하여 동작을 안내합니다.
- 상태 비저장 및 확장 가능
- UI, CLI, APIs 및 파이프라인에 포함할 수 있습니다.

제한 사항

- 메모리 없음 또는 기록 인식
- 외부 도구 또는 데이터 소스와의 상호 작용 없음
- 추론 시 LLM이 알고 있는 내용으로 제한됨

일반 사용 사례

- 대화 질문 및 답변
- 정책 설명 및 요약
- 의사 결정을 위한 지침
- 가볍고 자동화된 챗봇 흐름
- 분류, 레이블 지정 및 점수

구현 지침

다음 도구 및 서비스를 사용하여 기본 추론 에이전트를 생성할 수 있습니다.

- LLM 호출용 Amazon Bedrock(Anthropic, AI21, Meta)
- Amazon API Gateway 또는 AWS Lambda - 상태 비저장 마이크로서비스로 노출
- 파라미터 스토어 또는 코드 AWS Secrets Manager로 저장된 프롬프트 템플릿

요약

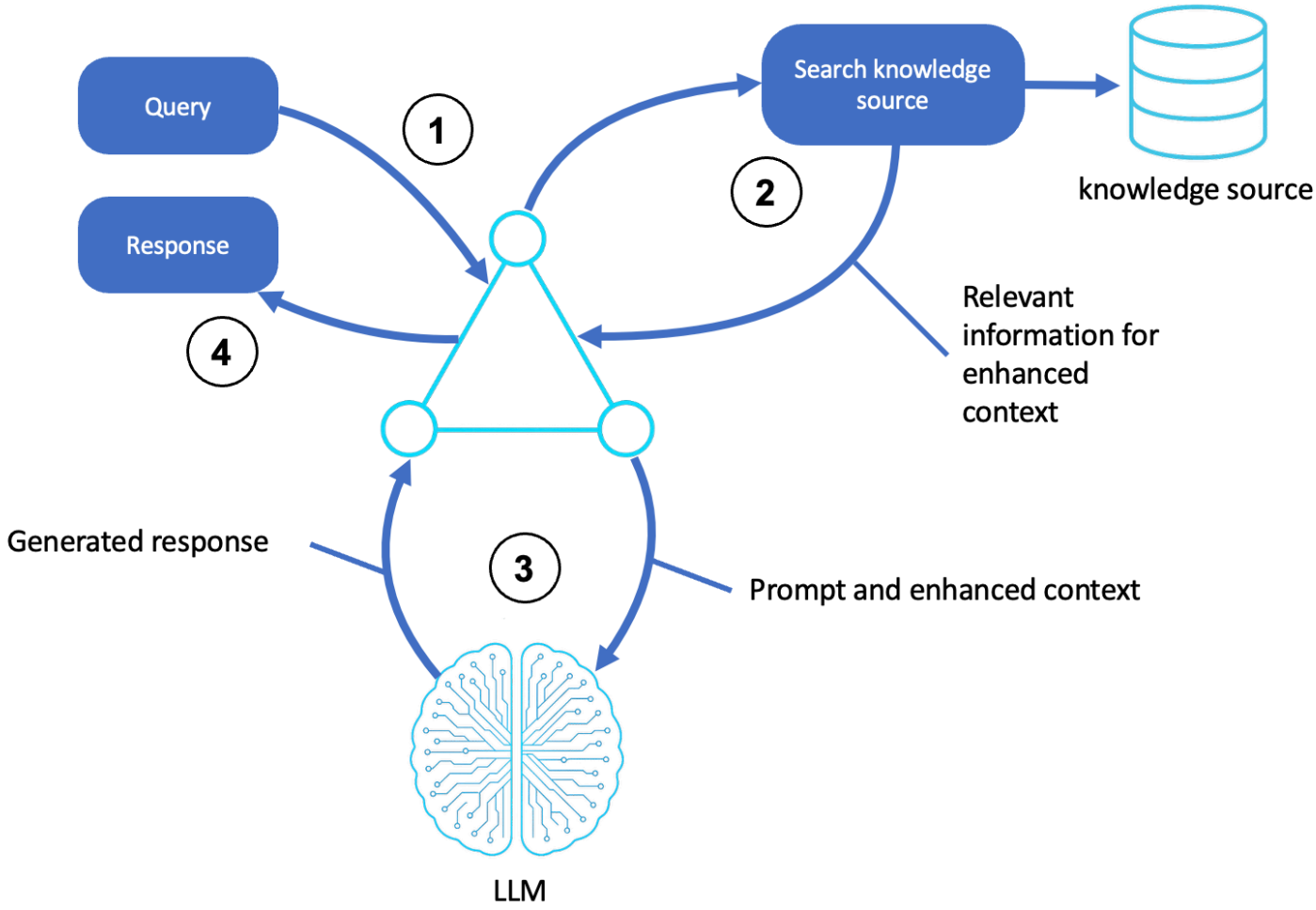
기본 추론 에이전트는 단순한 구조로 인해 기본적입니다. 목표를 지능형 출력으로 이어지는 추론 경로로 전환하는 핵심 기능이 있습니다. 이 패턴은 도구 기반 에이전트 및 검색 증강 생성(RAG)을 사용하는 에이전트와 같은 고급 패턴의 출발점인 경우가 많습니다. 또한 대규모 워크플로의 신뢰할 수 있는 모듈식 구성 요소입니다.

에이전트 RAG

검색 증강 생성(RAG)은 정보 검색과 텍스트 생성을 결합하여 정확하고 상황에 맞는 응답을 생성하는 기법입니다. RAG를 사용하면 에이전트가 LLM을 사용하기 전에 관련 외부 정보를 검색할 수 있습니다. 최신 정보, up-to-date 사실 정보 또는 도메인별 정보를 바탕으로 에이전트의 효과적인 메모리와 추론 정확도를 높입니다. 사전 훈련된 가중치에만 의존하는 상태 비저장 LLMs과 달리 RAG에는 컨텍스트를 사용하여 프롬프트를 동적으로 개선하는 외부 지식 검색 계층이 있습니다.

Architecture

RAG 패턴의 로직은 다음 다이어그램에 나와 있습니다.



설명

1. 쿼리를 수신합니다.
   - 사용자 또는 업스트림 시스템은 에이전트에게 쿼리 또는 목표를 제출합니다.
   - 에이전트 쉘은 요청을 수락하고 추론을 위한 프롬프트로 형식을 지정합니다.
2. 외부 소스를 검색합니다.
   - 에이전트는 쿼리에서 개념과 의도를 식별합니다.
   - 의미 체계 검색 또는 키워드 일치를 사용하여 벡터 저장소, 데이터베이스 또는 문서 인덱스와 같은 지식 소스를 쿼리합니다.
   - 가장 관련성이 높은 구절, 문서 또는 엔터티는 다음 단계에서 사용할 수 있도록 검색됩니다.
3. 컨텍스트 응답을 생성합니다.
   - 에이전트는 검색된 정보로 프롬프트를 보강하여 LLM에 대한 컨텍스트 강화 입력을 형성합니다.

- LLM은 생성형 추론(예: chain-of-thought 또는 반영)을 사용하여 모든 입력을 처리하여 정확한 응답을 생성합니다.

4. 최종 출력을 반환합니다.
   - 에이전트는 출력을 모든 통신 헤더 또는 필수 형식으로 래핑하여 준비한 다음 사용자 또는 호출 시스템에 반환합니다.
   - (선택 사항) 검색된 문서 및 LLM 출력은 향후 쿼리를 위해 로깅되고 점수가 매겨지고 메모리에 저장될 수 있습니다.

기능

- 롱테일 또는 엔터프라이즈별 도메인에서도 사실 기반 출력
- 모델을 미세 조정하지 않는 메모리 확장
- 각 쿼리 및 사용자 상태에 따른 동적 컨텍스트
- 벡터 데이터베이스, 의미 체계 인덱스 및 메타데이터 필터링과 완벽하게 호환

일반 사용 사례

- 엔터프라이즈 지식 도우미
- 규정 준수 봇
- 고객 지원 부조종사
- 검색 강화 챗봇
- 개발자 설명서 에이전트

구현 지침

다음 도구 및 서비스를 사용하여 RAG를 사용하는 에이전트를 생성합니다.

- LLM 호출을 위한 Amazon Bedrock
- 설명서 또는 구조화된 데이터 검색을 위한 Amazon Kendra, OpenSearch 또는 Amazon Aurora
- 문서 스토리지용 Amazon Simple Storage Service(Amazon S3)
- AWS Lambda 검색, 프롬프트 및 LLM 추론을 오케스트레이션하려면
- 에이전트와의 지식 기반 통합(메모리 플러그인, 의미 체계 리트리버 또는 Amazon Bedrock 사용)

요약

에이전트 RAG는 정적 모델 추론을 동적 실제 인텔리전스에 연결합니다. 에이전트가 모르는 내용을 조회하고, 검색된 지식에서 답변을 합성하고, 신뢰도가 높고 감사 가능한 응답을 생성할 수 있는 기능을 제공합니다.

RAG 패턴은 재훈련 없이 지식 액세스를 확장하는 지능형 에이전트를 구축하는 기반입니다. 도구 사용, 계획 및 장기 메모리와 관련된 보다 복잡한 오케스트레이션 패턴의 전조인 경우가 많습니다.

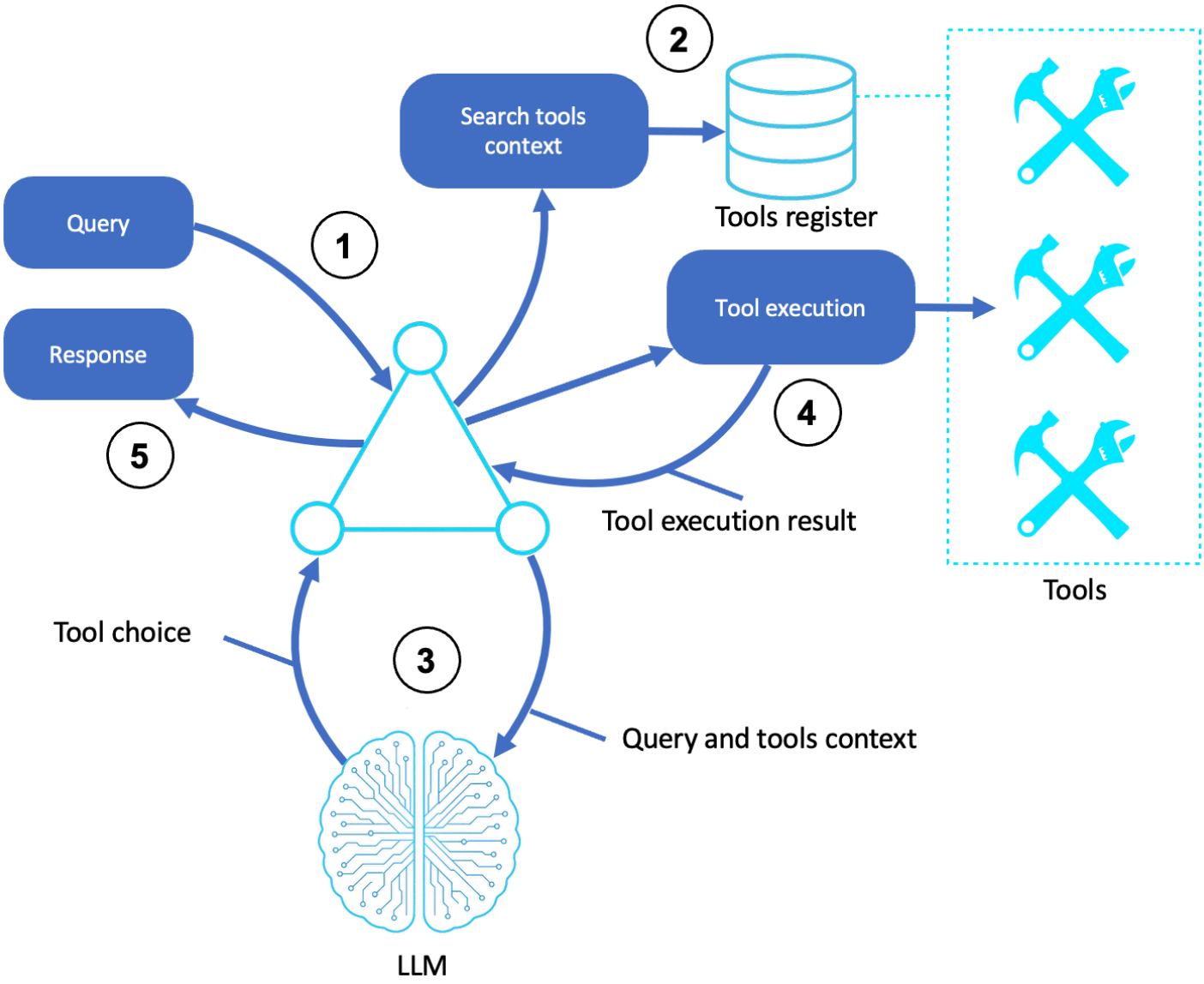
함수 호출을 위한 도구 기반 에이전트

도구 기반 에이전트는 외부 함수 또는 APIs를 호출하여 언어 전용 추론을 넘어선 작업을 완료함으로써 에이전트 추론 기능을 확장합니다. 이 패턴은 LLM을 사용하여 사용할 도구를 결정한 다음 호출 인수를 생성하고 도구의 출력을 추론 루프에 통합합니다.

이 패턴을 사용하면 에이전트가 단순히 응답을 제공하는 대신 조치를 취할 수 있습니다. 도구 인터페이스는 산술 계산 및 데이터베이스 조회부터 외부 APIs 및 클라우드 서비스에 이르기까지 호출 가능한 모든 기능을 나타냅니다.

Architecture

함수를 호출하기 위한 도구 기반 에이전트는 다음 다이어그램에 나와 있습니다.



설명

1. 쿼리를 수신합니다.
   - 에이전트는 사용자 또는 호출 시스템으로부터 자연어 쿼리 또는 작업을 수신합니다.
2. 도구 검색
   - 에이전트는 내부 메타데이터 또는 도구 레지스트리를 사용하여 사용 가능한 도구, 스키마 및 관련 기능을 검색합니다.
3. 도구를 선택하고 호출합니다.
   - LLM은 프롬프트에서 쿼리 및 도구 메타데이터(예: 함수 이름, 입력 유형 및 설명)를 수신합니다.
   - 가장 관련성이 높은 도구를 선택하고, 입력 인수를 구성하고, 구조화된 함수 호출을 반환합니다.

4. 선택한 도구를 실행합니다.
   - 에이전트 셸 또는 도구 실행기는 선택한 함수를 실행하고 결과(예: API 출력, 데이터베이스 값 또는 계산)를 반환합니다.
5. 응답을 반환합니다.
   - LLM은 직접 또는 업데이트된 프롬프트의 일부로 결과를 에이전트에 전달합니다. 그런 다음 자연어 결과를 반환합니다.

기능

- 작업 컨텍스트에 따른 동적 도구 선택
- 스키마 기반 프롬프트(OpenAPI, JSON 스키마, AWS 함수 인터페이스)
- 결과 해석 및 출력을 추론으로 연결
- 상태 비저장 또는 세션 인식 작업

일반 사용 사례

- 외부 데이터 액세스가 가능한 가상 어시스턴트
- 재무 계산기 및 예측기
- API 기반 지식 작업자
- 호출LLMs AWS Lambda, Amazon SageMaker 엔드포인트 및 SaaS 서비스

구현 지침

다음을 사용하여 함수 호출을 위한 도구 기반 에이전트를 생성합니다.

- 함수 호출을 지원하는 Amazon Bedrock(Anthropic Claude)
- AWS Lambda 도구 실행 백엔드로 사용
- Amazon API Gateway 또는 도구 오케스트레이션 AWS Step Functions 용
- 컨텍스트 인식 도구 메타데이터를 위한 Amazon DynamoDB 또는 Amazon Relational Database Service(RDS)
- 출력을 라우팅하기 위해 상태를 매핑 AWS Step Functions 하는 Amazon EventBridge 파이프라인 또는

요약

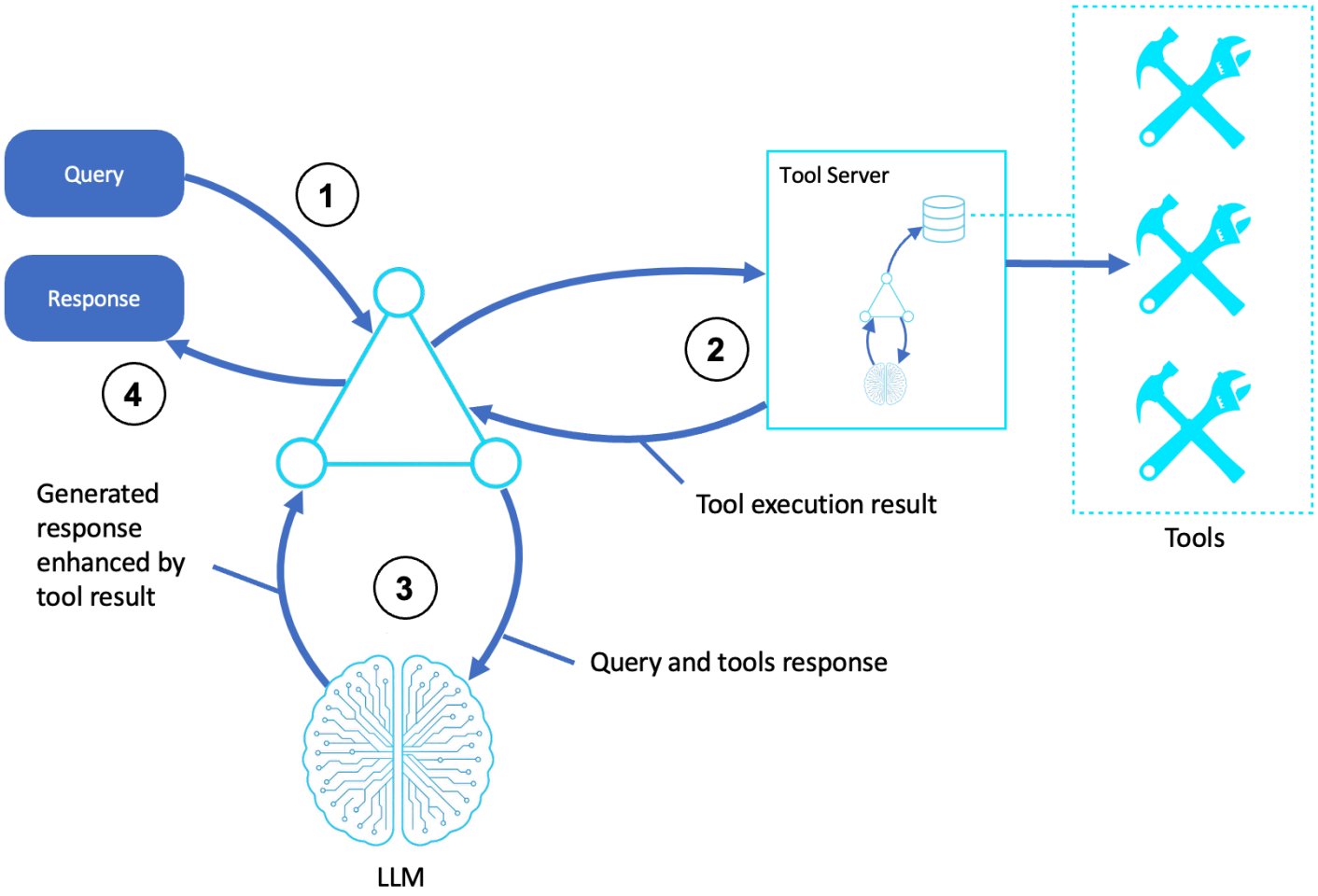
도구 기반 함수 호출 에이전트는 언어 이해에서 작업 수행으로의 전환을 나타냅니다. 이러한 에이전트는 LLM 추론을 유지하면서 동적 컨텍스트 인식 도구를 호출하여 수동 어시스턴트를 작업을 완료하고, 서비스에 액세스하고, 비즈니스 운영을 통합하는 시스템으로 변환합니다. 이 패턴은 특히 선언적 스키마, 권한 부여 프레임워크 및 다중 에이전트 시스템과 결합할 때 엔터프라이즈 설정에서 에이전트 AI의 중요한 구성 요소입니다.

서버용 도구 기반 에이전트

서버용 도구 기반 에이전트는 도구, 스크립트 및 복합 에이전트를 위한 전용 런타임 환경이 있는 외부 서버에 도구 실행을 위임하여 함수 호출 에이전트를 개선합니다. 에이전트 루프가 선택하고 호출하는 인라인 함수 호출과 달리 서버 기반 에이전트는 로직 및 실행 파이프라인을 다른 에이전트 또는 시스템에 아웃소싱합니다. 이를 통해 다중 도구 체인, 격리된 실행 및 특수 추론과 같은 고급 기능을 제공합니다. 도구 서버는 도구 자체에 별도의 AI 모델, 비즈니스 규칙 또는 환경이 포함될 수 있는 복잡하거나 상태 저장적이거나 리소스 집약적인 작업에 적합합니다.

Architecture

다음은 서버용 도구 기반 에이전트의 패턴입니다.



설명

1. 쿼리를 수신합니다.
   - 사용자 또는 시스템이 에이전트 셸에 요청을 제출합니다.
   - 에이전트는 쿼리를 해석하고 도구 서버로 디스패치할 준비를 합니다.
2. 도구 서버 프로세스를 실행합니다.
   - 에이전트는 구조화된 파라미터와 함께 작업을 도구 서버로 전송합니다.
   - 그러면 도구 서버가 다음을 수행할 수 있습니다.
     - 전용 컴퓨팅 시스템(예: AWS Lambda컨테이너 또는 Amazon SageMaker)에서 스크립트 또는 로직 실행
     - LLM 추론과 함께 자체 하위 에이전트를 사용하여 도구 선택 및 실행
     - 종속성, 재시도 또는 다단계 실행 흐름 관리
     - 작업이 완료되면 기본 에이전트에 결과 출력

3. 도구 출력과 함께 LLM 추론 사용
   - 에이전트는 LLM을 호출하여 원본 쿼리와 도구 서버 결과를 프롬프트의 일부로 전달합니다.
   - LLM은 새로 획득한 정보를 통합하는 응답을 합성합니다.
4. 응답을 반환합니다.
   - 에이전트는 사용자 또는 호출 시스템에 자연어 또는 구조화된 응답을 반환합니다.
   - (선택 사항) 결과는 메모리 또는 감사 로그에 저장될 수 있습니다.

기능

- 기본 에이전트 실행 루프 외부에서 도구가 호출됩니다.
- 도구 실행에는 LLM 호출, 로직 체인 또는 하위 에이전트가 포함될 수 있습니다.
- 에이전트는 도구 래퍼뿐만 아니라 컨트롤러 또는 디스패처 역할을 합니다.
- 로직의 구성 가능성, 확장성 및 격리를 활성화합니다.

일반 사용 사례

- 모델 체인 오케스트레이션(예: LLM, 비전 및 코드 결합)
- AI 기반 자동화 파이프라인
- 스크립트 실행기가 있는 DevOps 어시스턴트 에이전트
- 복잡한 재무 계산, 시뮬레이션 또는 최적화 에이전트
- 멀티모달 도구(예: 오디오, 설명서 및 작업 결합)

구현 지침

AWS 서비스다음을 사용하여이 패턴을 빌드할 수 있습니다.

- Amazon Bedrock(에이전트 호스트 및 LLM 추론)
- AWS Lambda AWS Fargate, Amazon ECS 또는 Amazon SageMaker 엔드포인트를 도구 서버 런타임으로 사용
- Amazon API Gateway 또는 AWS App Runner - 도구 서버 APIs 노출
- 분리된 agent-to-tool 메시징을 위한 Amazon EventBridge
- AWS Step Functions 도구 서버에 다중 에이전트 로직을 구성하는 AWS AppFabric 경우 또는

요약

서버를 사용하는 도구 기반 에이전트는 모듈식이며 확장 가능합니다. 결정 로직을 실행과 분리하므로 프라이머리 에이전트는 복잡하거나 민감한 작업을 다른 시스템으로 오프로드하면서 경량 상태를 유지할 수 있습니다. 이는 엔터프라이즈급 에이전트 AI, 특히 거버넌스, 관찰성, 격리, 동적 구성 또는 이들의 조합이 필요한 환경에 중요합니다.

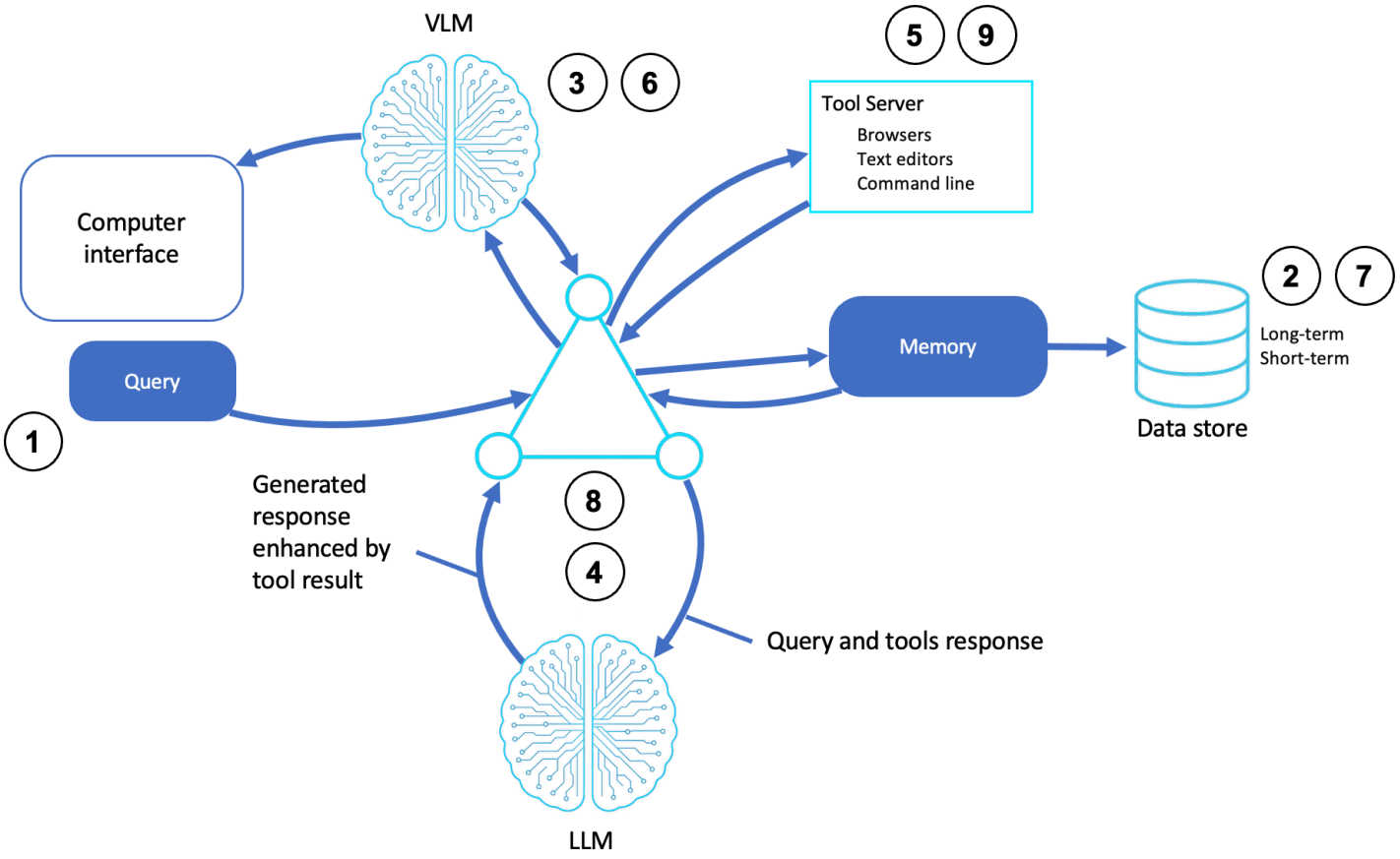
컴퓨터 사용 에이전트

컴퓨터 사용 에이전트는 브라우저, 터미널, 파일 시스템 및 애플리케이션과 같은 디지털 환경을 시뮬레이션하거나 제어할 수 있습니다. 이러한 에이전트는 LLM 추론, 시각적 언어 모델(VLMs) 및 명령을 실행하거나 입력 이벤트를 시뮬레이션하는 도구 서버를 결합하여 사용자 의도를 해석하고, 시각적 및 텍스트 인터페이스와 상호 작용하고, 목표 지향적 작업을 수행합니다.

이 패턴은 에이전트가 어시스턴트뿐만 아니라 동일한 도구와 환경을 사용하여 인간처럼 작업을 수행하는 프록시 역할을 하는 실용적인 AI 자동화에 중요합니다.

Architecture

컴퓨터 사용 에이전트 패턴은 다음 다이어그램에 나와 있습니다.



설명

1. 쿼리를 수신합니다.
   - 작업 또는 요청은 UI, API 또는 자연어 인터페이스를 통해 제공됩니다.
2. 메모리에 액세스합니다.
   - 에이전트는 단기 및 장기 메모리를 검색하여 과거 명령, 목표 및 시스템 상태를 재현합니다.
3. 시각적 컨텍스트 분석
   - VLM은 컴퓨터 화면, 시스템 상태 또는 UI 요소를 관찰하여 주어진 컨텍스트를 이해하고 실행 가능한 항목을 식별합니다.
4. LLM을 통한 이유
   - LLM은 쿼리, 메모리 상태, 도구 및 서버 응답을 결합하여 다음 작업을 결정합니다.
5. 도구 서버와 상호 작용
   - 에이전트는 서버에서 호스팅되는 도구를 호출하며, 여기에는 다음이 포함될 수 있습니다.
     - 브라우저(예: 헤드리스 Chrome) 및 쉘 환경
     - 텍스트 및 코드 편집기

    - 사용자 지정 스크립트 인터페이스
6. 시각적 입력 업데이트
    - 시스템 UI가 변경되거나 추가 관찰이 필요한 경우 VLM은 화면 상태 또는 텍스트 버퍼를 다시 분석할 수 있습니다.
7. 메모리 업데이트
    - 새로운 인사이트, 시스템 상태 또는 사용자 피드백은 단기 및 장기 메모리에 기록됩니다.
8. 최종 결정 및 설명 공식화
    - LLM은 쿼리 및 도구 출력을 기반으로 결과를 합성하거나 작업을 권장합니다.
9. 응답을 반환합니다.
    - 에이전트는 인터페이스에 결과(예: 완료된 작업, 확인 또는 생성된 콘텐츠)를 반환합니다.

기능

- 시각적 입력과 텍스트 입력을 사용한 멀티모달 추론
- 시뮬레이션된 입력 또는 API 기반 입력을 통한 애플리케이션 제어
- 영구 상태에 대한 메모리 관리
- 시퀀스 실행의 자율성(다단계 흐름)

일반 사용 사례

- IDEs에서 코드를 작성하고 실행하는 AI 개발자
- 반복적인 디지털 워크플로를 위한 컴퓨터 사용 에이전트
- 소프트웨어 테스트 및 품질 보증을 위해 시뮬레이션된 사용자
- 음성 또는 상위 수준 지침을 통해 UIs를 탐색하기 위한 접근성 에이전트
- 추론을 통해 향상된 스마트 로봇 프로세스 자동화(RPA)

구현 지침

- AWS 서비스다음을 사용하여이 패턴을 빌드할 수 있습니다.
- LLM 기반 계획 및 추론을 위한 Amazon Bedrock
- 시뮬레이션된 UI 환경으로 도구 서버를 실행하기 위한 Amazon Elastic Compute Cloud(Amazon EC2) AWS Lambda또는 Amazon SageMaker 노트북

- 메모리 지속성을 위한 Amazon Simple Storage Service(Amazon S3) 또는 Amazon DynamoDB
- 하이브리드 시나리오의 UI 이미지 분석을 위한 Amazon Rekognition(또는 사용자 지정 모델)
- 관찰성 및 감사 추적을 AWS X-Ray 위한 Amazon CloudWatch Logs 또는

요약

컴퓨터 사용 에이전트는 자율 디지털 운영자 역할을 하여 인간-컴퓨터 상호 작용과 AI 기반 작업 간의 격차를 해소합니다. 이러한 에이전트는 메모리, 도구 오케스트레이션 및 VLMs 통합하여 인간을 위해 설계된 시스템과 적응형으로 상호 작용하고, 작업을 실행하고, 파일을 업데이트하고, 메뉴를 탐색하고, 응답을 생성할 수 있습니다.

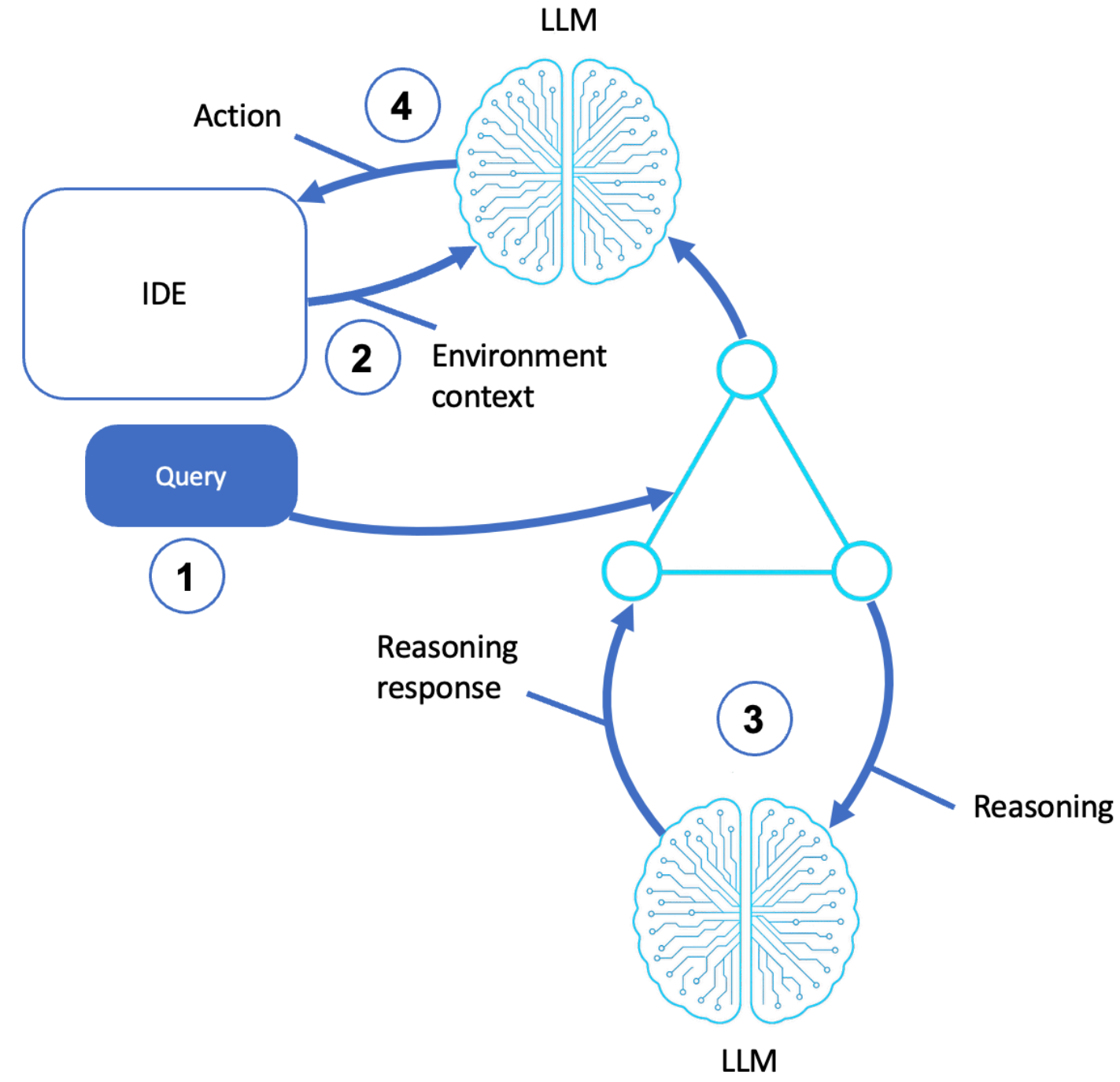
코딩 에이전트

코딩 에이전트는 프로그래밍 작업을 추론하고, 코드를 생성 또는 수정하고, IDEs 및 CLIs. 이러한 에이전트는 자연어 이해와 구조화된 추론을 결합하여 함수 생성부터 버그 수정 및 테스트 작성에 이르기까지 소프트웨어 개발을 지원, 보강 및 자동화합니다.

자동 완성 도구와 달리 코딩 에이전트는 사용자 목표를 적극적으로 해석하고, 개발 환경에서 컨텍스트를 쿼리하고(예: 파일을 열고 오류를 추적), 요구 사항을 식별한 다음 작업을 제안하고 수행합니다.

Architecture

코딩 에이전트 패턴은 다음 다이어그램에 나와 있습니다.



설명

1. 쿼리를 수신합니다.
   - 사용자는 명령 팔레트, 채팅 창 또는 CLI(예: “이 함수에 로깅 추가” 또는 “가독성을 위한 리팩터링”)를 통해 자연어 지침을 제공합니다.
2. 환경 컨텍스트 추출

- 에이전트는 활성 파일, 커서 위치, 코드 조각 및 기호 테이블을 포함하여 IDE에서 컨텍스트를 수집합니다.
- 오류 메시지, 테스트 결과 및 다른 에이전트의 출력을 출력합니다.

3. LLM 추론
   - 에이전트는 쿼리 및 환경 컨텍스트를 포함한 프롬프트를 LLM으로 전송합니다.
     - LLM은 추론 패스를 수행하여 다음을 결정합니다.
     - 변경해야 할 사항
     - 솔루션을 생성하는 방법
     - 모든 리팩터링, 재작성 또는 코딩 단계
4. 작업을 실행합니다.
   - LLM은 에이전트에 출력을 반환하고 IDE 또는 런타임 환경으로 가져옵니다.
   - 여기에는 코드 삽입 또는 수정, 설명 또는 설명서 생성, 다운스트림 빌드, 테스트 및 린팅 작업 트리거가 포함될 수 있습니다.

기능

- 높은 컨텍스트 인식(예: IDE 상태, 커서 및 구문 트리)
- 목표 및 피드백의 반복적인 추론
- 선택적 코드 계획 및 작업 분리(예: 첫 번째 이유와 조치)
- 동기식 또는 비동기식 개발자 워크플로에서 작동

일반 사용 사례

- 작업 설명에서 코드 생성
- 코드 리팩터링 및 최적화
- 테스트 사례 생성 및 검증
- 오류 설명 및 디버깅
- 설명서 도우미
- 페어링된 프로그래밍 코파일럿

구현 지침

- 다음 도구 및를 사용하여이 패턴을 빌드할 수 있습니다 AWS 서비스.
- LLM 기반 생성 및 추론을 위한 Amazon Bedrock
- 제안 및 완료 코딩을 위한 Amazon Q Developer
- AWS Lambda 샌드박스 환경을 실행하고 테스트하기 위한 또는 Amazon Elastic Container Service(Amazon ECS)
- AWS Cloud9, VS Code 확장 또는 컨텍스트를 호스팅하고 평가하기 위한 사용자 지정 IDE 통합
- 중간 프롬프트, 응답 및 개정 기록을 저장하기 위한 Amazon Simple Storage Service(Amazon S3)

요약

코딩 에이전트는 자연어를 해석하고, 컨텍스트를 분석하고, 다단계 코드 변경을 생성하고, 소프트웨어 개발 수명 주기와 통합할 수 있는 새로운 AI 기반 개발 도구입니다.

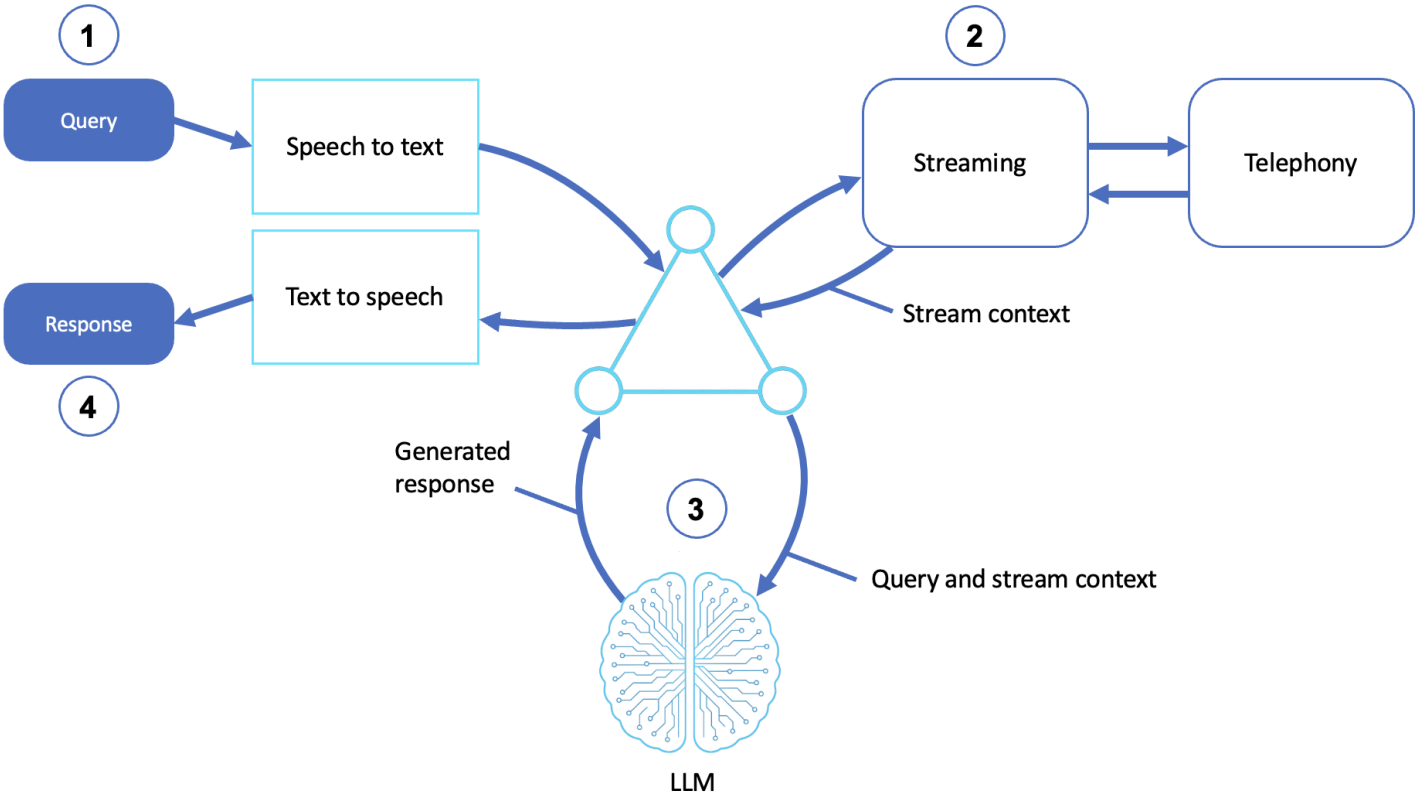
음성 및 음성 에이전트

음성 및 음성 에이전트는 음성 대화를 통해 사용자와 상호 작용합니다. 이러한 에이전트는 음성 인식, 자연어 이해 및 음성 합성을 통합하여 텔레포니, 모바일, 웹 및 임베디드 플랫폼에서 대화형 AI를 지원합니다.

음성 에이전트는 핸즈프리, 실시간 또는 접근성 기반 환경에서 특히 효과적입니다. 스트리밍 인터페이스와 LLM 기반 추론을 결합하면 사용자에게 자연스럽게 느껴지는 풍부하고 동적인 상호 작용을 촉진할 수 있습니다.

Architecture

음성 및 음성 에이전트는 다음 다이어그램에 나와 있습니다.



설명

1. 음성 쿼리를 수신합니다.
   - 사용자가 전화, 마이크 또는 임베디드 시스템에 대한 요청을 음성으로 보냅니다.
   - 스피speech-to-text(STT) 모듈은 오디오를 텍스트로 변환합니다.
2. 스트리밍 및 텔레포니 컨텍스트 통합
   - 에이전트는 스트리밍 인터페이스를 사용하여 오디오 I/O를 실시간으로 관리합니다.
   - 고객 센터 또는 통신 컨텍스트에 배포된 경우 텔레포니 통합은 세션 라우팅, 듀얼 톤 다중 주파수(DTMF) 입력 및 미디어 전송을 처리합니다.

참고: DTMF는 전화 키패드에서 버튼을 누를 때 생성되는 톤을 나타냅니다. 음성 에이전트 내 스트리밍 및 텔레포니 컨텍스트 통합의 맥락에서 DTMF는 전화 통화 중, 특히 대화형 음성 응답(IVR) 시스템에서 신호 입력 메커니즘으로 사용됩니다. DTMF 입력을 통해 에이전트는 다음을 수행할 수 있습니다.

- 메뉴 선택 항목을 인식합니다(예: 결제는 "1을 누릅니다. 지원하려면 2를 누릅니다.")
- 숫자 입력 수집(예: 계정 번호, PINs 및 확인 번호)
- 통화 흐름에서 워크플로 또는 상태 전환 트리거

- 필요한 경우 음성에서 터치 톤으로 되돌리기

1. LLM 스트림 컨텍스트를 통한 이유
   - 쿼리는 에이전트로 전송되어 세션 메타데이터(예: 호출자 ID, 이전 컨텍스트)와 함께 에이전트를 LLM으로 전달합니다.
   - LLM은 상호 작용이 진행 중인 경우 chain-of-thought 전략을 사용하거나 멀티턴 메모리를 사용하여 응답을 생성합니다.
2. 음성 응답을 반환합니다.
   - 에이전트는 text-to-speech(텍스트 음성 변환)를 사용하여 응답을 음성으로 변환합니다.
   - 음성 채널을 통해 사용자에게 오디오를 반환합니다.

기능

- 실시간 음성 이해 및 생성
- STT 및 TTS를 지원하는 다국어 I/O
- 텔레포니 또는 스트리밍 APIs와 통합
- 전환 간 세션 인식 및 메모리 핸드오프

일반 사용 사례

- 대화형 IVR 시스템
- 가상 리셉셔니스트 및 약속 스케줄러
- 음성 기반 헬프데스크 에이전트
- 웨어러블 보이스 어시스턴트
- 스마트 홈 및 접근성 도구를 위한 음성 인터페이스

구현 지침

다음 도구 및를 사용하여이 패턴을 빌드할 수 있습니다 AWS 서비스.

- Amazon Lex V2 또는 STT용 Amazon Transcribe
- TTS용 Amazon Polly

- 스트리밍 및 텔레포니를 위한 Amazon Chime SDK, Amazon Connect 또는 Amazon Interactive Video Service(Amazon IVS)
- Anthropic, AI21 또는 기타 파운데이션 모델을 사용한 추론을 위한 Amazon Bedrock
- AWS Lambda STT, LLM, TTS 및 세션 컨텍스트를 연결하는 방법

(선택 사항) 추가 개선 사항에는 다음이 포함될 수 있습니다.

- 컨텍스트 인식 RAG용 Amazon Kendra 또는 OpenSearch
- 세션 메모리용 Amazon DynamoDB
- 추적성을 AWS X-Ray 위한 Amazon CloudWatch Logs 및

요약

음성 및 음성 에이전트는 자연스러운 대화를 통해 상호 작용하는 지능형 시스템입니다. 음성 인터페이스를 LLM 추론 및 실시간 스트리밍 인프라와 통합하여 음성 에이전트는 원활하고 액세스 가능하며 확장 가능한 상호 작용을 가능하게 합니다.

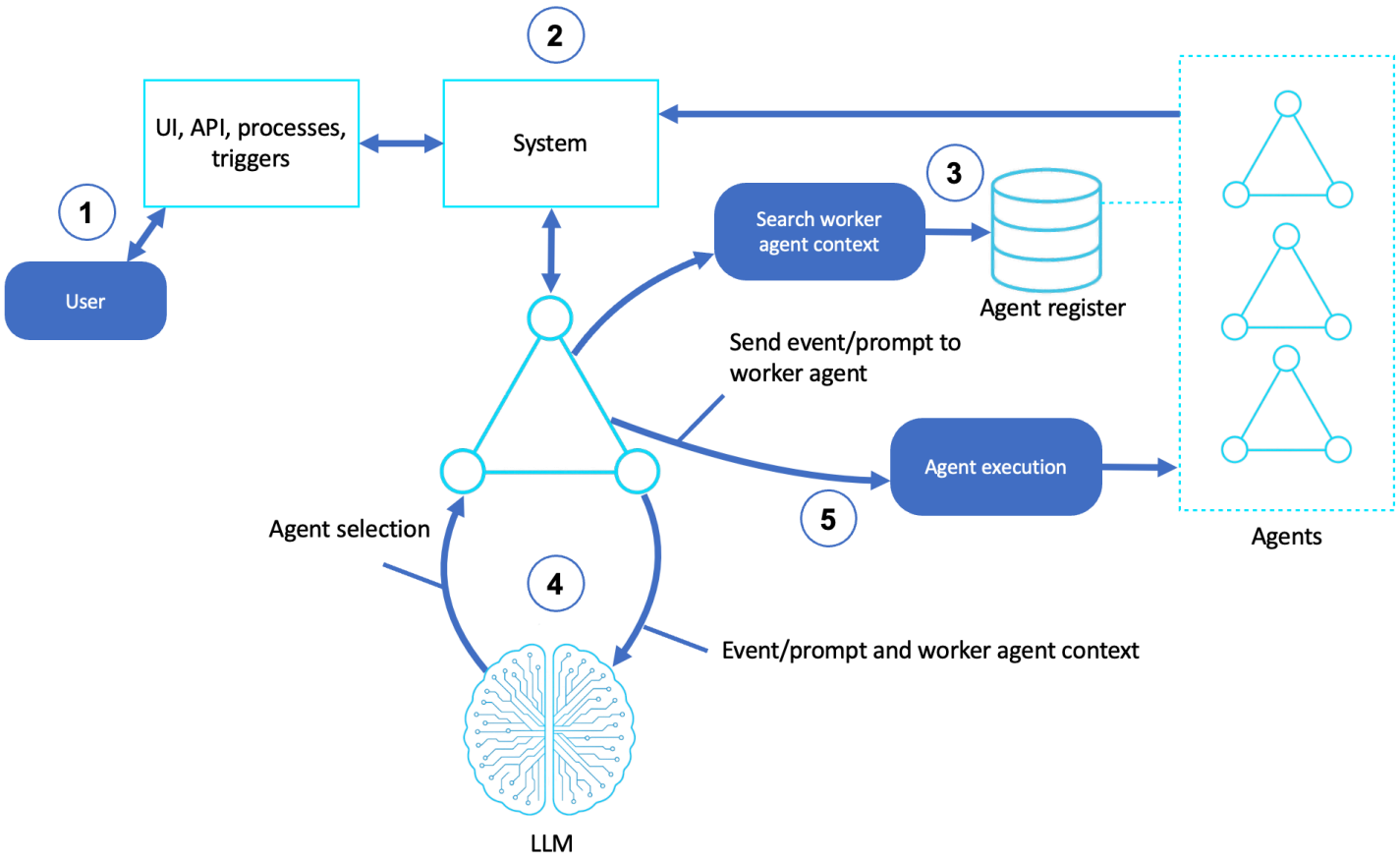
워크플로 오케스트레이션 에이전트

워크플로 오케스트레이션 에이전트는 분산 시스템에서 다단계 작업, 프로세스 및 서비스를 관리하고 조정합니다. 이러한 에이전트는 추론하고 독립적으로 행동하는 대신 하위 에이전트 또는 다른 시스템에 작업을 위임하고 실행 컨텍스트를 유지하며 중간 결과에 따라 조정합니다.

이러한 에이전트는 자동화 흐름의 기본 부분입니다. 다양한 에이전트와 도구를 순차적으로 또는 조건부로 호출해야 하는 장기 실행 작업, 다중 에이전트 구성 및 교차 도메인 통합을 처리할 때 특히 유용합니다.

Architecture

워크플로 오케스트레이션 에이전트는 다음 다이어그램에 나와 있습니다.



설명

1. 사용자 입력을 수신합니다.
   - 사용자(또는 외부 트리거)는 UI, API 또는 시스템 이벤트를 통해 작업을 시작합니다.
2. 시스템 이벤트 처리
   - 시스템 구성 요소는 요청을 수신하고 오케스트레이션이 필요한 이벤트 또는 명령을 내보냅니다.
3. 컨텍스트를 검색합니다.
   - 워크플로 에이전트는 지식 기반 및 에이전트 레지스트리를 쿼리하여 메타데이터, 도메인 및 이전 성공률을 기반으로 작업에 적합한 작업자 에이전트를 찾습니다.
4. LLM 에이전트를 선택합니다.
   - LLM은 작업 설명과 사용 가능한 옵션을 분석하여 최상의 에이전트 또는 워크플로 계획을 선택하는 데 도움이 됩니다.
   - 또한 선택한 에이전트에게 전송할 작업별 프롬프트를 공식화할 수도 있습니다.
5. 위임 및 실행
   - 선택한 작업자 에이전트가 이벤트 또는 프롬프트를 수신하고 명령 실행을 시작합니다.

- 실행 상태를 추적하고, 실패 시 재시도하고, 중간 결과를 시퀀스의 다음 에이전트에 전달할 수 있습니다.

기능

- 에이전트 구성(예: 감독자, 공동 작업자 에이전트 및 도구)
- 이벤트 기반 또는 예약된 실행
- 시간 경과에 따른 메모리 및 상태 추적
- 계층적 또는 병렬 작업 오케스트레이션(비동기식 워크플로와 비동기식 워크플로 비교)
- 동적 에이전트 선택 및 연결

일반 사용 사례

- 다단계 자동화(예: 데이터 수집 및 보고)
- 고객 서비스 라우팅 및 에스컬레이션(예: agent-as-coordinator)
- AI 에이전트가 동일한 루프 내에서 인간 및 봇과 조정
- LLM 기반 로직을 사용하여 엔터프라이즈 프로세스 자동화
- 하이브리드 시스템은 AI 에이전트와 기존 오케스트레이션 도구를 결합합니다.

구현 지침

다음 도구 및를 사용하여이 패턴을 빌드할 수 있습니다 AWS 서비스.

- 추론 및 에이전트 선택을 위한 Amazon Bedrock
- AWS Step Functions 워크플로 구성을 위한 또는 Amazon EventBridge
- AWS Lambda 실행 단위 또는 작업 실행기로 사용
- 상태 및 결과를 추적하기 위한 Amazon DynamoDB, Amazon Simple Storage Service(Amazon S3) 또는 Amazon RDS
- AWS AppFabric 또는 시스템 간 조정을 위한 Amazon AppFlow
- (선택 사항) Amazon SageMaker 실행 에이전트를 사용하여 도메인별 작업자 에이전트 호스팅

요약

워크플로 에이전트는 다중 에이전트 환경에서 목표를 조정, 조정 및 조정합니다. 즉, AI 에이전트는 설명 가능한 모듈식 워크플로를 통해 협업하고, 런타임 조건에 적응하고, 복잡한 결과를 제공할 수 있습니다.

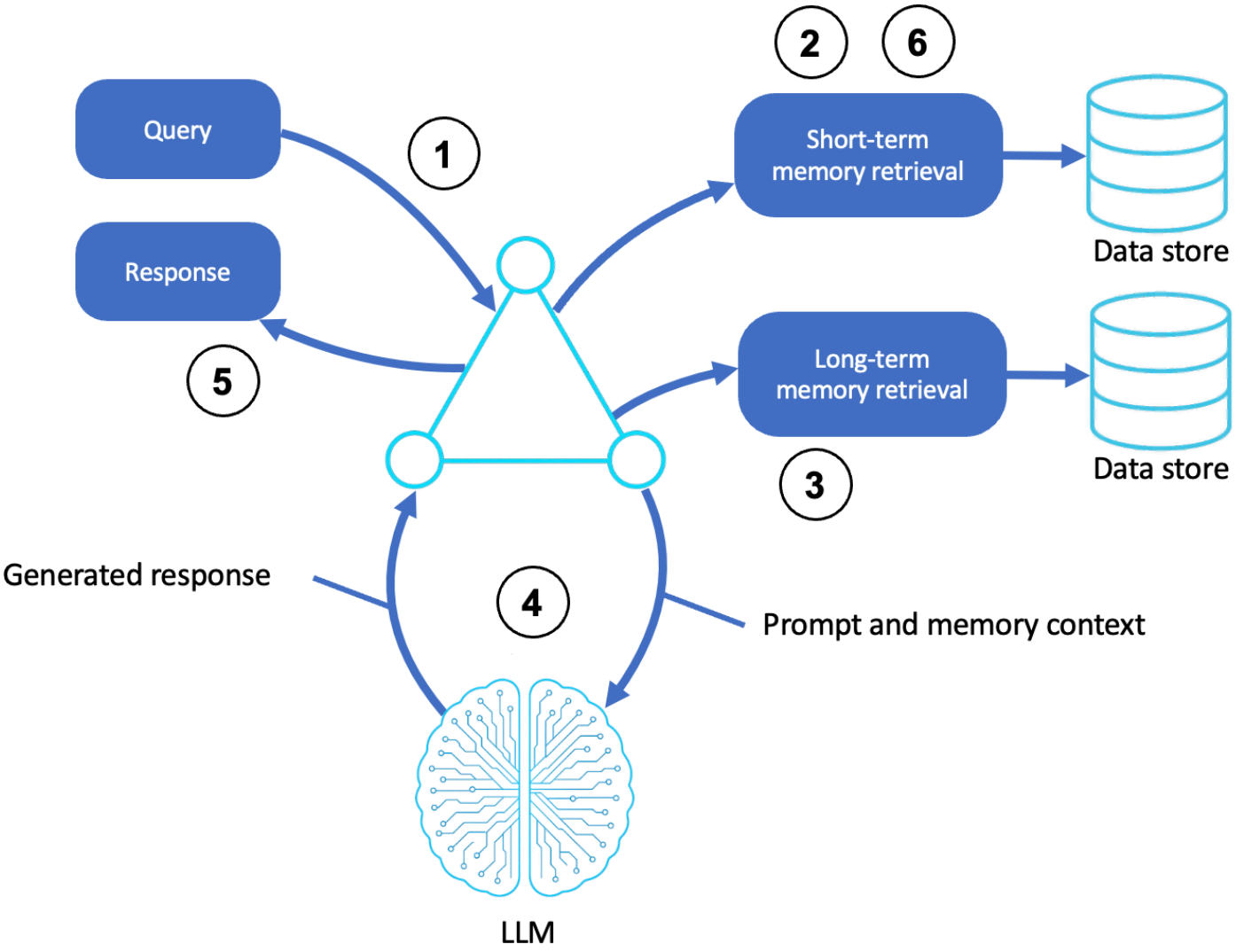
메모리 증강 에이전트

단기 및 장기 메모리를 사용하여 저장, 검색 및 추론할 수 있는 기능으로 메모리 증강 에이전트가 향상됩니다. 이를 통해 여러 작업, 세션 및 상호 작용에서 컨텍스트를 유지할 수 있으므로 보다 일관성 있고 개인화된 전략적 응답을 얻을 수 있습니다.

상태 비저장 에이전트와 달리 메모리 증강 에이전트는 과거 데이터를 참조하여 적응하고, 이전 결과에서 학습하고, 사용자의 목표, 기본 설정 및 환경에 맞는 결정을 내립니다.

Architecture

메모리 증강 에이전트는 다음 다이어그램에 나와 있습니다.



설명

1. 입력 또는 이벤트를 수신합니다.
   - 에이전트는 사용자 쿼리 또는 시스템 이벤트를 수신합니다. 텍스트, API 트리거 또는 환경 변경일 수 있습니다.
2. 단기 메모리를 검색합니다.
   - 에이전트는 세션 또는 워크플로와 관련된 최근 대화 기록, 작업 컨텍스트 또는 시스템 상태를 검색합니다.
3. 장기 메모리를 검색합니다.
   - 에이전트는 장기 메모리(예: 벡터 데이터베이스 및 키-값 저장소)를 쿼리하여 다음과 같은 과거 인사이트를 확인합니다.

- 사용자 기본 설정
- 과거 결정 및 결과
- 학습한 개념, 요약 또는 경험

4. LLM을 통한 이유
   - 메모리 컨텍스트는 LLM 프롬프트에 내장되어 에이전트가 현재 입력과 이전 지식을 모두 기반으로 추론할 수 있습니다.
5. 출력을 생성합니다.
   - 에이전트는 작업 기록 및 사용자의 입력에 따라 개인화된 컨텍스트 인식 응답, 계획 또는 작업을 생성합니다.
6. 메모리 업데이트
   - 업데이트된 목표, 성공 및 실패 신호, 구조화된 응답과 같은 새로운 정보는 향후 작업을 위해 저장됩니다.

기능

- 대화 또는 이벤트 간 세션 연속성
- 시간 경과에 따른 목표 지속성
- 진화하는 상태에 따른 상황 인식
- 이전 성공 및 실패로 인한 적응성
- 사용자 기본 설정 및 기록과 일치하는 개인화

일반 사용 사례

- 사용자 기본 설정을 기억하는 대화식 코파일럿
- 코드베이스 변경 사항을 추적하는 코딩 에이전트
- 작업 기록에 따라 조정되는 워크플로 에이전트
- 시스템 지식에서 발전하는 디지털 트윈
- 중복 검색을 방지하는 에이전트 조사

메모리 증강 에이전트 구현

메모리 증강 에이전트에는 다음 도구 및 AWS 서비스 를 사용합니다.

메모리 계층	AWS 서비스	용도
단기	Amazon DynamoDB, Redis, Amazon Bedrock 컨텍스트	최근 상호 작용 상태의 빠른 검색
장기(정형)	Amazon Aurora, Amazon DynamoDB, Amazon Neptune	팩트, 관계 및 로그
장기(시맨틱)	OpenSearch, PostgreSQL, Pinecone	임베딩 기반 검색(즉, RAG)
스토리지	Amazon S3	트랜스크립트, 구조화된 메모리 및 파일 저장
오케스트레이션	AWS Lambda 또는 AWS Step Functions	메모리 주입 및 업데이트 수명 주기 관리
추론	Amazon Bedrock	메모리 프롬프트가 있는 Anthropic Claude 또는 Mistral

메모리 주입 프롬프트 구현

메모리를 에이전트 추론에 통합하려면 구조화된 상태와 검색 증강 컨텍스트 주입의 조합을 사용합니다.

- 언어 모델에 대한 프롬프트를 구성할 때 최신 에이전트 상태 및 최근 대화 기록을 구조화된 입력으로 포함시켜 전체 컨텍스트로 추론할 수 있도록 합니다.
- 검색 증강 생성(RAG)을 사용하여 장기 메모리에서 관련 문서 또는 사실을 가져옵니다.
- 압축 및 관련성에 대한 이전 계획, 컨텍스트 및 상호 작용을 요약합니다.
- 추론 중에 벡터 저장소 또는 구조화된 로그와 같은 외부 메모리 모듈을 주입하여 의사 결정을 안내합니다.

요약

메모리 증강 에이전트는 경험을 통해 배우고 사용자 컨텍스트를 기억하여 사고 연속성을 유지합니다. 이러한 에이전트는 장기 협업, 개인화 및 전략적 추론을 사용하여 사후 대응 인텔리전스를 능가합니다.

에이전트 AI 측면에서 메모리를 사용하면 에이전트가 적응형 디지털 대응 도구처럼 동작하고 상태 비저장 도구처럼 동작하지 않을 수 있습니다.

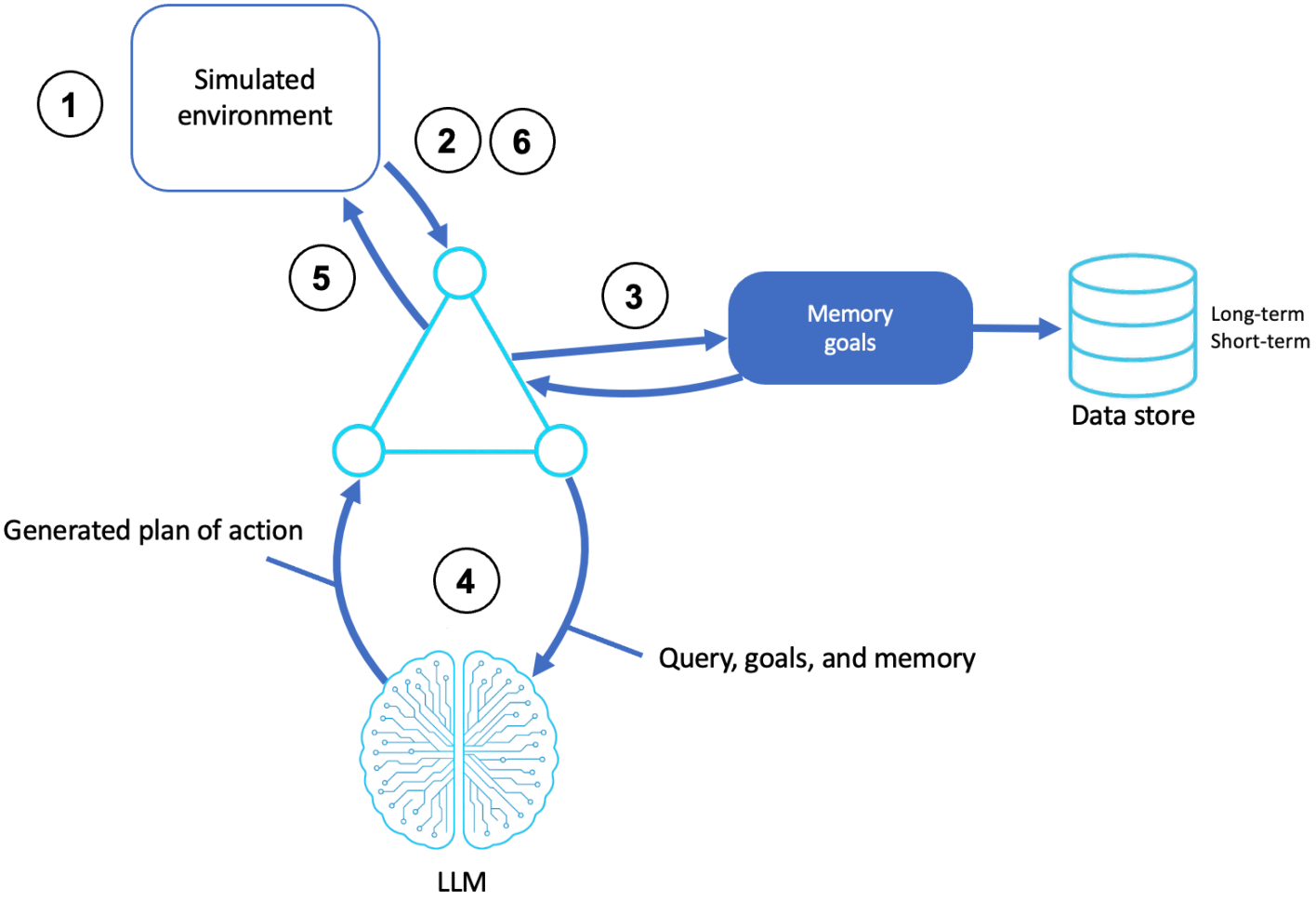
시뮬레이션 및 테스트베드 에이전트

시뮬레이션 및 테스트 기반 에이전트는 추론, 행동 및 학습을 하는 가상화되거나 제어된 환경에서 작동합니다. 이러한 에이전트는 실제 환경에 적용하기 전에 반복 가능한 설정으로 동작, 모델 결과 및 훈련 전략을 시뮬레이션합니다.

이 패턴은 반복 개발, 강화 학습(RL), 자율 의사 결정 평가 및 긴급 동작 테스트에 유용합니다. 시뮬레이션 에이전트는 종종 닫힌 루프에서 작동하여 환경으로부터 피드백을 받고 그에 따라 동작을 조정하므로 공간 추론, 실시간 제어 또는 복잡한 시스템 역학과 관련된 작업에 매우 중요합니다.

Architecture

다음 다이어그램은 시뮬레이션 또는 테스트베드 에이전트를 보여줍니다.



설명

1. 환경을 시작합니다.
   - 에이전트는 시뮬레이션된 환경(예: 3D 월드, 물리 엔진, CLI 샌드박스 또는 합성 데이터 스트림)을 시작합니다.
   - 에이전트는 초기 작업, 목표 또는 정책을 사용하여 환경에 로드됩니다.
2. 에이전트를 인식합니다.
   - 에이전트는 시뮬레이션 원격 측정(예: 센서 에뮬레이션, 가상 카메라 및 구조화된 로그)을 통해 현재 상태를 인식합니다.
3. 목표 및 메모리를 검색합니다.
   - 에이전트는 할당된 목표, 시나리오 지침 또는 컨텍스트 목표를 검색합니다.
   - 또한 다음을 포함하여 이전 메모리를 검색할 수 있습니다.
     - 장기 전략 또는 정책
     - 환경 맵 또는 알려진 제약 조건
     - 유사한 시뮬레이션의 과거 성공 또는 실패
4. 이유 및 계획
   - LLM은 시뮬레이션된 상태, 작업 목표 및 학습된 지식을 해석합니다.
   - 작업 계획 또는 제어 명령을 생성합니다.
5. 시뮬레이션된 작업을 실행합니다.
   - 에이전트는 계획을 실행하거나, 상태를 수정하거나, 공간을 탐색하거나, 가상 엔터티와 상호 작용합니다.
6. 학습
   - 에이전트가 작업 결과 평가
   - 에이전트의 구성에 따라 다음을 수행할 수 있습니다.
     - RL 수행
     - 향후 미세 조정을 위한 로그 결과
     - 전략을 실시간으로 조정

기능

- 합성 또는 가상 환경 내에서 작동
- trial-and-error 학습, 정책 구체화 및 시스템 모델링 지원

- 동작, 장애 처리 및 엣지 케이스에 대한 위험이 낮은 테스트
- 다중 에이전트 설정에서 긴급 에이전트 동작 분석 활성화
- 폐쇄 루프 제어 및 human-in-the-loop 루프 탐색 모두 지원

일반 사용 사례

- 로봇, 드론 및 게임을 위한 강화 학습
- 가상 도로에서의 자율 차량 훈련
- DevOps 및 테스트베드 시나리오를 위한 시뮬레이션된 UIs 또는 CLIs
- 소셜 시뮬레이션에서 발생한 행동 실험
- 프로덕션 전 결정 로직의 안전성 검증

구현 지침

다음 도구 및를 사용하여 시뮬레이션 및 테스트용 에이전트를 빌드할 수 있습니다 AWS 서비스.

구성 요소	AWS 서비스	용도
환경	Amazon SageMaker 스튜디오 랩의 Amazon ECS, Amazon Amazon EC2 또는 사용자 지정 시뮬레이터	가상 세계(Gazebo, Unity, Unreal) 또는 샌드박스 CLIs 실행
에이전트 로직	Amazon Bedrock, Amazon SageMaker 또는 AWS Lambda	LLM 기반 플래너 또는 RL 에이전트
피드백 루프	Amazon SageMaker 강화 학습, Amazon CloudWatch 또는 사용자 지정 로그	보상 추적, 결과 점수 및 동작 로깅
메모리 및 재생	Amazon S3, Amazon DynamoDB 또는 Amazon RDS	영구 상태, 에피소드 기록 또는 시나리오 데이터

시각화	Amazon CloudWatch 대시보드 또는 Amazon SageMaker 노트북	정책 변경 사항, 결과 및 훈련 지표 관찰

다음은 추가 애플리케이션입니다.

- AWS SimSpace Weaver 대규모 공간 시뮬레이션용
- AWS IoT Core 섀도우 디바이스 테스트용
- 에이전트 평가 및 벤치마킹을 위한 Amazon SageMaker 실험

요약

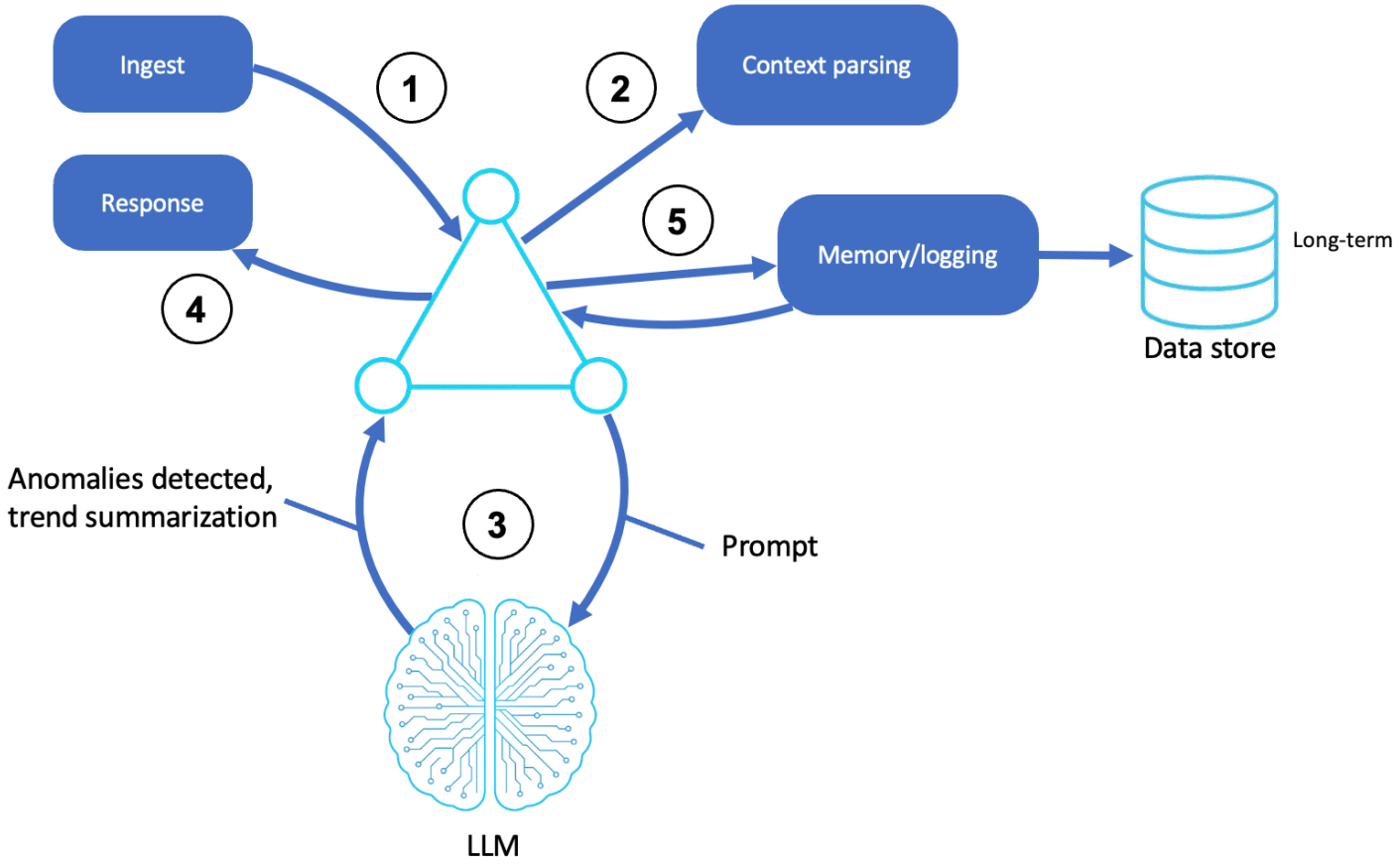
시뮬레이션 및 테스트용 에이전트는 프로덕션 시스템에 배포되기 전에 구조화된 탐색을 위한 것입니다. 이러한 에이전트를 사용하여 자율 탐색 정책을 훈련하고, 합성 환경에서 비즈니스 프로세스를 테스트하고, 조정 패턴에 대한 스웜을 평가할 수 있습니다.

관찰자 및 모니터링 에이전트

관찰자 및 모니터링 에이전트는 시스템, 환경 및 상호 작용을 수동적으로 관찰하여 패턴을 감지하고 인사이트를 생성하며 작업을 트리거합니다. 지능형 감시자는 동작을 직접 시작하지 않고도 알림, 진단 및 감사를 개선합니다.

이러한 에이전트는 특히 AI-in-the-loop 모니터링, 이상 탐지, 규정 준수 감독 및 보안 인텔리전스에서 기존 모니터링에 적응성 또는 추론이 부족한 경우에 뛰어납니다. 옵저버 에이전트는 시스템 원격 측정 및 사용자 상호 작용을 지속적으로 모니터링하는 이벤트 리스너입니다. 에이전트는 지각, 해석 및 조건부 에스컬레이션 또는 보고에 따라 달라집니다.

Architecture



설명

1. 원격 측정 수집
   - 에이전트는 다음과 같은 하나 이상의 시스템 소스로부터 입력을 수신합니다.
     - 로그(애플리케이션, 인프라, 보안)
     - 지표(성능, 지연 시간, 사용량)
     - 이벤트(API 호출, 사용자 작업, 센서 데이터)
2. 구문 분석 컨텍스트
   - 원시 입력은 구문 분석되고 구조화되며 타임스탬프, 액터 자격 증명, 시스템 상태 및 추적 ID와 같은 메타데이터로 보강됩니다.
3. LLM을 사용하는 이유
   - 에이전트는 LLM 또는 로직 모듈을 사용하여 이상을 식별하고, 추세를 요약하고, 분산 트레이스 또는 기간 간에 상호 연관시켜 구문 분석된 입력을 해석합니다.
4. 분류 또는 알림

- 에이전트는 관찰된 동작이 다음을 정당화하는지 확인합니다.
  - 알림 또는 에스컬레이션
  - 보고서 또는 대시보드 업데이트
  - 응답 트리거(예: 자동 문제 해결 및 정책 적용)

5. 로그 메모리 또는 피드백 루프
   - 에이전트는 장기 학습, 감사 또는 다른 에이전트에 대한 향후 참조를 위해 이벤트와 결정을 저장합니다.

기능

- 수동 및 비침습적(에이전트가 직접 작동하지 않음)
- 확장성과 비동기성이 뛰어남
- 노이즈 또는 분산 신호 간 AI 기반 상관관계
- 감사, 규정 준수 및 실시간 인사이트 지원
- 다운스트림 에이전트 또는 인적 워크플로에 피드 가능

일반 사용 사례

- 마이크로서비스 및 APIs에 대한 AI 증강 관찰성
- 모델 드리프트, 정책 위반 또는 out-of-band 동작 모니터링
- 고객 활동 분석 또는 상호 작용 요약
- 커밋 또는 배포를 모니터링하는 코드 검토 에이전트
- LLM 추론을 사용한 보안 또는 규정 준수 로그 모니터링

구현 지침

다음 도구 및를 사용하여 관찰자 및 모니터링 에이전트를 빌드할 수 있습니다 AWS 서비스.

구성 요소	AWS 서비스	용도
이벤트 수집	Amazon EventBridge, Amazon CloudWatch Logs, Amazon Kinesis, Amazon S3	정형 및 비정형 원격 측정 수집

사전 처리	AWS Lambda, AWS Glue, AWS Step Functions	원시 데이터를 구조화된 프롬프트로 변환
추론 엔진	Amazon Bedrock, Amazon SageMaker, AWS Lambda	이벤트 분석, 동작 분류, 인사이트 생성
스토리지 및 메모리	Amazon S3, Amazon DynamoDB, OpenSearch	지속적인 관찰, 요약 및 출력
알림 및 에스컬레이션	Amazon SNS, AWS AppFabric, Amazon EventBridge	다운스트림 시스템 또는 에이전트 트리거

다음은 추가 애플리케이션입니다.

- AWS Security Hub CSPM 보안 로그 모니터링을 위한
- 에이전트 출력을 시각화하기 위한 Amazon Quick Suite

요약

관찰자 및 모니터링 에이전트는 시스템과 동작을 실시간으로 추적합니다. 인간 또는 규칙이 간과할 수 있는 패턴을 식별하여 이상을 탐지하고, 보안을 감사하고, 운영 인텔리전스를 수집합니다. 이 기능은 변화하는 조건에 적응하고 포괄적인 데이터 분석을 기반으로 결정을 내릴 수 있는 시스템을 만드는 데 도움이 됩니다.

다중 에이전트 공동 작업

다중 에이전트 공동 작업은 각각 고유한 역할, 전문화 또는 목표를 가진 여러 자율 에이전트가 협상하여 복잡한 작업을 해결하는 패턴을 말합니다. 이러한 에이전트는 정보를 공유하고, 책임을 나누고, 목표에 대한 공동 추론을 통해 독립적으로 또는 다른 에이전트와 함께 운영할 수 있습니다.

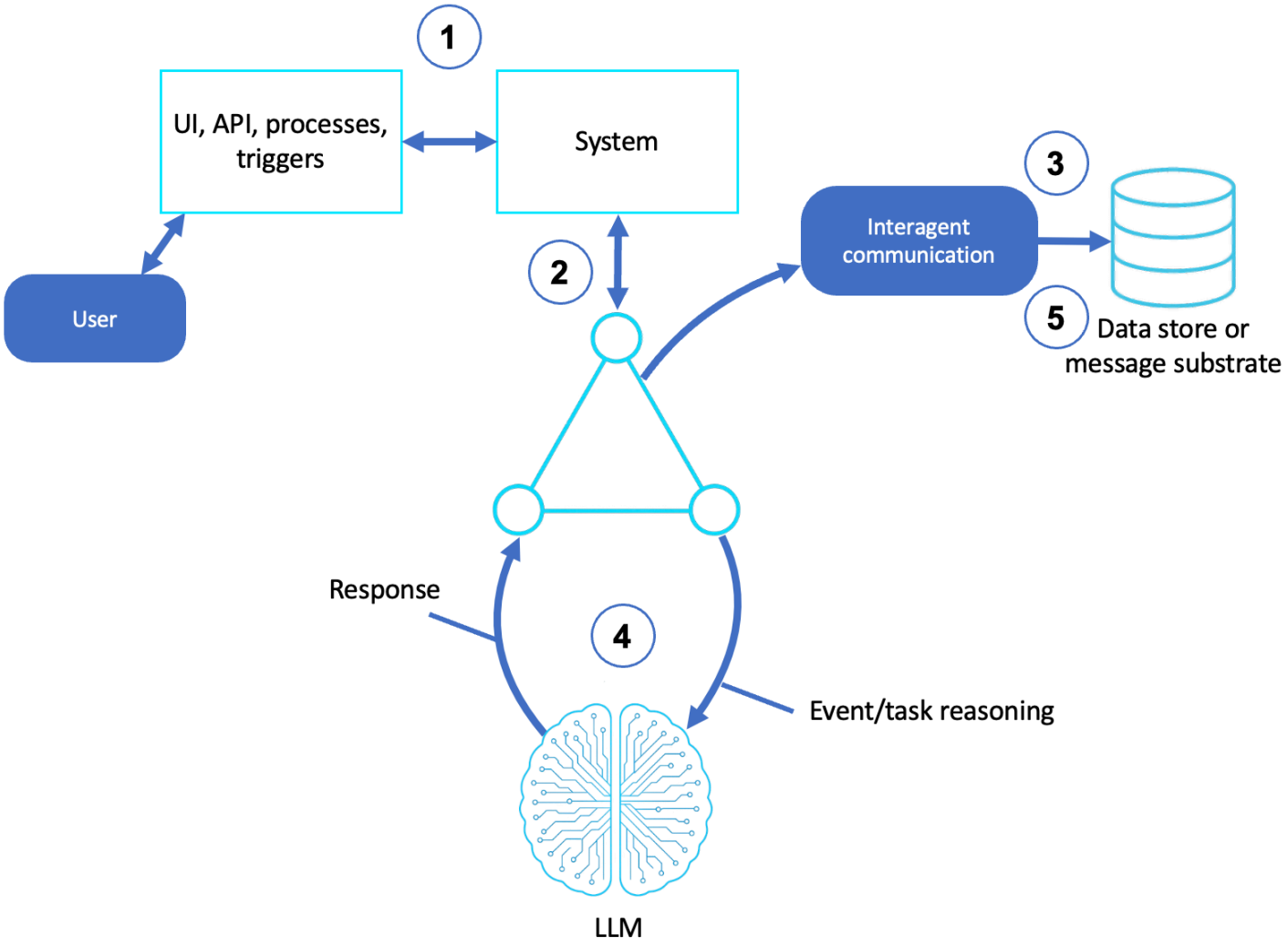
이 패턴은 구조화된 흐름에서 하위 에이전트에게 작업을 중앙에서 조정하고 위임하는 워크플로 에이전트와 다릅니다. 반대로 다중 에이전트 공동 작업은 적응성, 병렬 처리 및 인식 분할을 활성화하여 peer-to-peer 또는 긴급 조정을 강조합니다. 다음 표에서는 다중 에이전트 공동 작업을 워크플로 에이전트와 비교합니다.

Feature	워크플로 에이전트	용도

컨트롤	중앙 집중식 코디네이터	분산형, 분산형 또는 역할 기반 피어
상호 작용	에이전트 한 명이 실행을 위임하고 추적합니다.	여러 에이전트가 협상, 공유 및 조정
설계	미리 정의된 작업 시퀀스	긴급하고 유연한 작업 배포
조정	절차 오케스트레이션	협력적 또는 경쟁적 상호 작용
사용 사례	엔터프라이즈 프로세스 자동화	복잡한 추론, 탐색 및 긴급 전략

Architecture

다음 다이어그램은 다중 에이전트 공동 작업을 보여줍니다.



설명

1. 작업을 시작합니다.
   - 사용자 또는 시스템이 높은 수준의 목표 또는 문제를 내보냅니다.
   - "관리자" 에이전트 또는 시작 컨텍스트가 목표를 정의합니다.
2. 역할을 할당하거나 검색합니다.
   - 에이전트가 플래너, 연구원, 실행자, 비평가 또는 설명자와 같은 다른 역할에 자체 할당(대칭 로직 또는 추론)되거나 위임(이벤트 브로커)됩니다.
3. 다른 에이전트와 통신합니다.
   - 에이전트는 공유 메모리, 메시징 대기열 또는 프롬프트 체인을 통해 통신합니다.
   - 서로 토론하거나 쿼리하거나 하위 작업을 제안할 수 있습니다.
4. 특수 추론 사용
   - 각 에이전트는 자체 모델 또는 도메인 로직을 사용하여 문제의 부분을 해결합니다.
   - 에이전트는 역할별 프롬프트 및 메모리와 함께 LLMs을 사용할 수 있습니다.
5. 출력 또는 목표를 조정합니다.
   - 에이전트는 기여를 최종 답변, 계획 또는 조치로 합성합니다.
   - (선택 사항) 감독 에이전트는 합성된 출력을 검증하거나 요약할 수 있습니다.

기능

- 특수 역할 또는 기술을 갖춘 피어 레벨 에이전트
- 커뮤니케이션 또는 협상을 통해 발생한 행동
- 복잡하거나 다면적인 문제의 병렬 처리
- 숙고, 자체 수정 및 반사 반복 지원
- 소셜 역학, 과학 협업 또는 엔터프라이즈 팀 역할 모델링

일반 사용 사례

- 자율 연구 팀(검색 에이전트, 요약기 및 검사기)
- 소프트웨어 개발(플래너, 코더 및 테스터)
- 비즈니스 시나리오 모델링(재무, 정책 및 규정 준수)
- 협상, 입찰 또는 다자간 추론

- 멀티모달 작업(이미지, 텍스트 및 로직)

구현 지침

다음 도구 및를 사용하여 다중 에이전트 시스템을 빌드할 수 있습니다 AWS 서비스.

구성 요소	AWS 서비스	용도
에이전트 호스팅	Amazon Bedrock, Amazon SageMaker, AWS Lambda	개별 LLM 기반 에이전트 호스팅
통신 계층	Amazon SQS, Amazon EventBridge, AWS AppFabric	에이전트 간의 메시징 및 조정
공유 메모리	Amazon DynamoDB, Amazon S3 또는 OpenSearch	다중 에이전트 메모리 또는 블랙보드 시스템
오케스트레이션 계층	AWS Step Functions, AWS Lambda 파이프라인	킥오프, 제한 시간, 대체 및 재시도 로직
에이전트 식별	Amazon Bedrock 에이전트(역할 정의) AWS AppConfig 및 Amazon Bedrock Converse API(Amazon Bedrock 외부의 에이전트)	역할 기반 도구 또는 에이전트 호출 및 경계 적용
발생한 상호 작용	Amazon EventBridge 파이프라인 또는 에이전트 레지스트리	동적 작업 라우팅 또는 에스컬레이션 활성화

요약

다중 에이전트 공동 작업은 모듈식 역할 기반 에이전트에 문제 해결 작업을 분산합니다. 워크플로 오케스트레이션과 달리 협업 패턴은 인간이 문제를 해결하는 방법을 반영하는 새로운 인텔리전스, 복원력 및 확장성을 사용합니다. 개방형 도메인, 창의적인 작업, 멀티모달 추론 및 다양한 관점에서 이점을 얻을 수 있는 환경에 특히 유용합니다.

결론

앞서 설명한 패턴은 에이전트 AI의 실제 구현에 대한 기본 접근 방식을 보여줍니다. 기본 추론에서 메모리 증강 인텔리전스에 이르기까지 각 패턴은 자율성, 비동기성 및 기관을 기반으로 하는 지각, 인식 및 작업에 맞게 고유하게 구성됩니다.

이러한 패턴은 지능형 목표 지향 시스템을 구축하기 위한 어휘와 기술 청사진을 공유합니다. 패턴이 사용자 인터페이스에 포함되든, 클라우드 서비스를 통해 오케스트레이션되든, 에이전트 팀 간에 조정되든, 각 패턴은 조정 가능하고 모듈화됩니다.

요점

- 에이전트 패턴 구성 가능 - 대부분의 실제 에이전트는 둘 이상의 패턴(예: 도구 기반 추론 및 메모리가 있는 음성 에이전트)을 혼합합니다.
- 에이전트 설계는 상황에 따라 다름 - 상호 작용 표면, 작업 복잡성, 지연 시간 허용 범위 및 도메인별 제약 조건을 기반으로 패턴을 선택합니다.
- AWS 네이티브 구현 달성 가능 - Amazon Bedrock, Amazon SageMaker AWS Lambda AWS Step Functions및 이벤트 기반 아키텍처를 사용하면 모든 에이전트 패턴을 대규모로 제공할 수 있습니다.

LLM 워크플로

에이전트 패턴에서는 일반적인 AI 에이전트 패턴을 살펴보았습니다. 각 패턴은 지각, 행동, 학습 및 인식과 같은 모듈식 기능 세트를 중심으로 구축되었습니다. 많은 에이전트 패턴에서 인지 모듈의 핵심은 추론, 계획 및 의사 결정을 할 수 있는 대규모 언어 모델(LLM)입니다. 그러나 LLM만 호출하는 것만으로는 지능적이고 목표 지향적인 동작을 생성하기에 충분하지 않습니다.

복잡한 작업을 안정적으로 수행하려면 에이전트가 구조화된 워크플로 내에 LLM을 포함해야 합니다. 여기서 모델의 기능은 도구, 메모리, 계획 루프 및 조정 로직으로 보강됩니다. 이러한 LLM 워크플로를 사용하면 에이전트가 목표를 세분화하고, 하위 작업을 라우팅하고, 외부 서비스를 호출하고, 결과를 반영하고, 다른 에이전트와 조정할 수 있습니다.

이 장에서는 재사용 가능한 워크플로를 중심으로 구성된 강력하고 확장 가능하며 지능적인 LLM 기반 인지 모듈을 구축하기 위한 핵심 설계 패턴을 소개합니다.

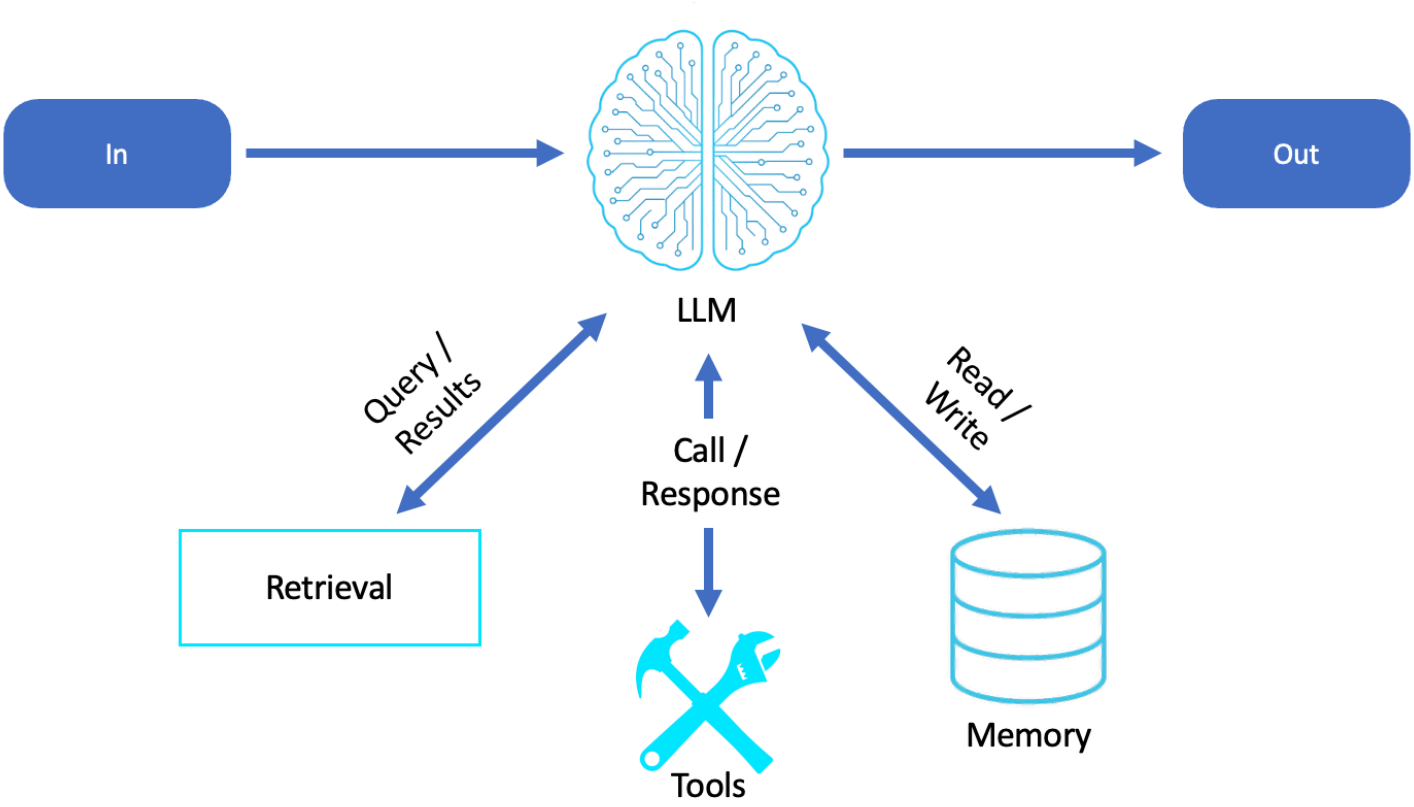
이 섹션의 내용

LLM 증강 인식 개요

핵심에서 소프트웨어 에이전트의 인지 모듈은 증강으로 래핑된 LLM으로 볼 수 있습니다. 에이전트는 다음 구성 요소를 사용하여 환경 내에서 효과적으로 추론할 수 있습니다.

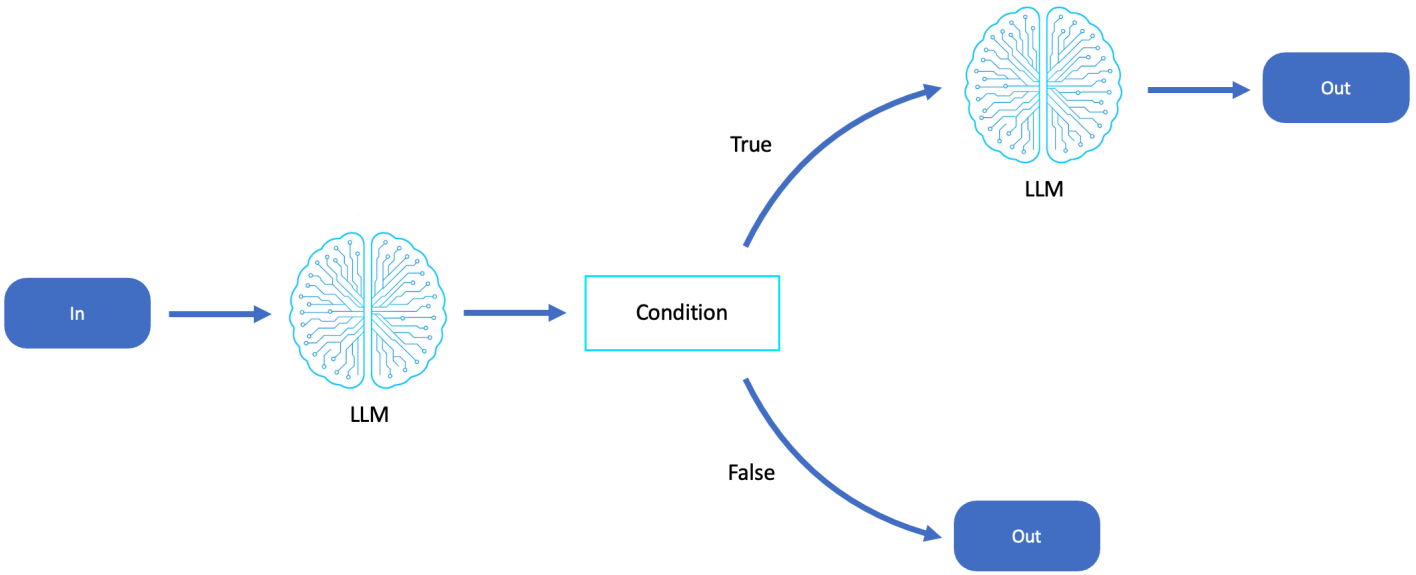
- 프롬프트 - 컨텍스트, 지침, 예제 및 메모리를 사용하여 입력 프레이밍
- 검색 - 검색 증강 생성(RAG)과 같은 벡터 검색 또는 의미 체계 메모리를 통해 LLM 프롬프트에 up-to-date 또는 도메인별 지식 제공
- 도구 사용 - LLM이 APIs를 호출하거나 함수를 호출하여 정보를 검색하거나 작업할 수 있습니다.
- 메모리 - 구조화된 데이터베이스 또는 컨텍스트 요약을 사용하여 영구 또는 세션 기반 상태를 추론 루프에 통합

이러한 증강은 시간 경과에 따라 여러 작업에서 LLM이 사용되는 방식을 정의하는 워크플로로 구성되어 상태 비저장 엔진에서 동적 추론 에이전트로 변환합니다.



프롬프트 체인을 위한 워크플로

프롬프트 체인은 복잡한 작업을 일련의 단계로 분해합니다. 여기서 각 단계는 이전 단계의 출력을 처리하거나 기반으로 빌드하는 개별 LLM 호출입니다.



프롬프트 체인 워크플로는 작업을 논리적으로 순차적 추론 단계로 나눌 수 있고 중간 출력이 다음 단계를 알려주는 시나리오에 적합합니다. 문서 검토, 코드 생성, 지식 추출 및 콘텐츠 구체화와 같은 구조화된 사고, 점진적 변환 또는 계층화된 분석이 필요한 워크플로에 탁월합니다.

설명

- 작업의 복잡성이 단일 LLM 호출의 컨텍스트 기간 또는 추론 깊이를 초과합니다.
- 한 단계의 출력(예: 분석, 요약 또는 계획)은 후속 결정 또는 생성 단계의 입력이 됩니다.
- 추론 단계에서 투명성과 제어가 필요합니다(예: 감사 가능한 중간 결과).
- 단계 사이에 외부 검증, 필터링 또는 보강 로직을 연결하려고 합니다.
- 연구 에이전트, 에디토리얼 어시스턴트, 계획 시스템, 다단계 코파일럿 등 파이프라인 스타일의 추론 루프에서 작업하는 에이전트에게 적합합니다.

기능

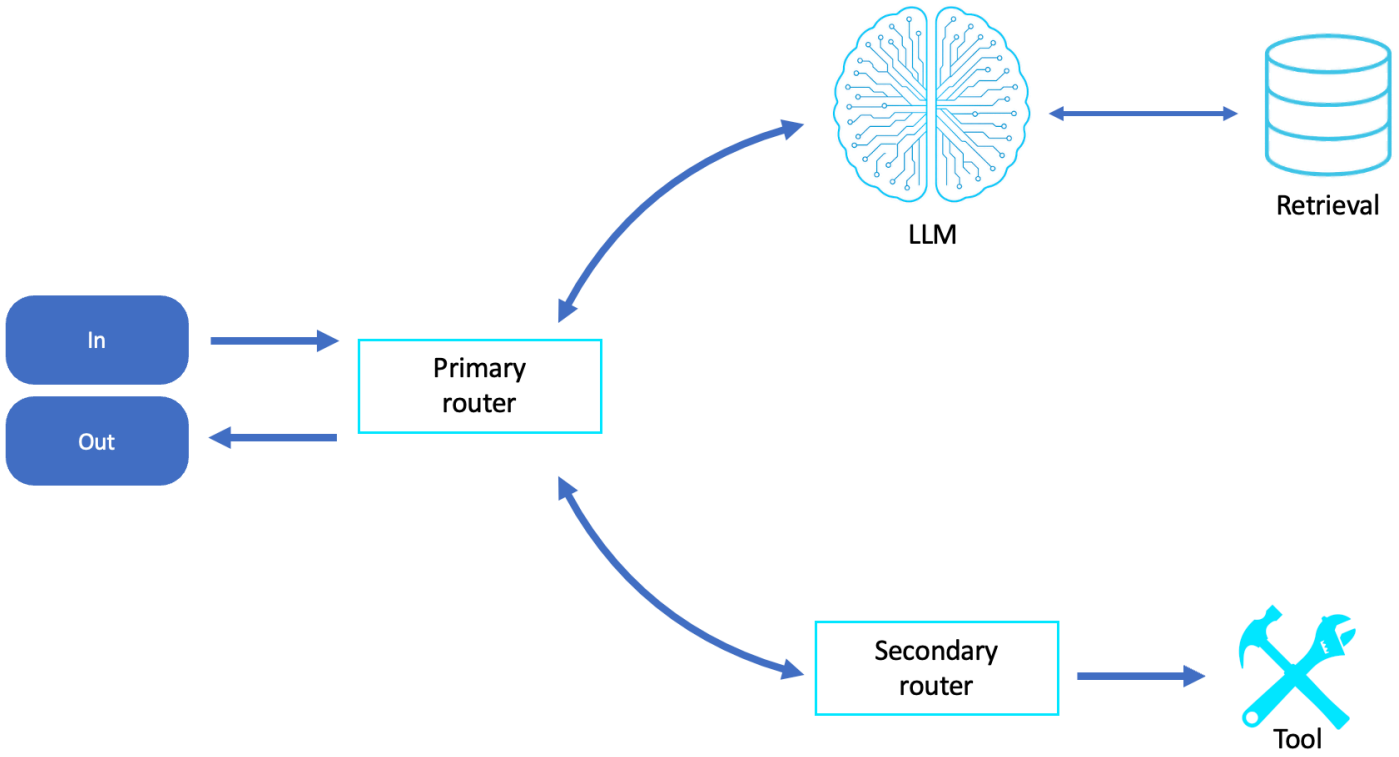
- LLM 호출의 선형 또는 분기 체인
- 구조화된 입력으로 전달되거나 후속 프롬프트에 포함된 중간 결과
- AWS Step Functions AWS Lambda또는 에이전트별 실행기로 오케스트레이션할 수 있습니다.

일반 사용 사례

- 다단계 추론 작업(예: “평가 재작성 요약”)
- 계층화된 출력을 합성하는 연구 도우미(예: "사실 답변 질문 검색")
- 코드 생성 파이프라인(“계획 작성 코드 테스트 코드 설명 출력 생성”)

라우팅을 위한 워크플로

라우팅 패턴에서 분류자 또는 라우터 에이전트는 LLM을 사용하여 쿼리의 의도 또는 범주를 해석한 다음 입력을 특수 다운스트림 작업 또는 에이전트로 라우팅합니다.



라우팅 워크플로는 에이전트가 입력 의도, 작업 유형 또는 도메인을 빠르게 분류한 다음 특수한 하위 에이전트, 도구 또는 워크플로에 요청을 위임해야 하는 시나리오에서 사용됩니다. 이는 일반 어시스턴트 역할을 하는 에이전트, 엔터프라이즈 함수의 정문 또는 도메인에 걸쳐 있는 사용자 대면 AI 인터페이스와 같은 기능 에이전트에 특히 유용합니다.

라우팅은 다음과 같은 경우에 특히 효과적입니다.

- 다양한 작업(예: 검색, 요약, 예약, 계산)에서 요청을 분류합니다.
- 보다 전문화된 워크플로에 들어가기 전에 입력을 사전 처리하거나 정규화해야 합니다.

- 다양한 입력 유형(예: 이미지와 텍스트, 정형 쿼리와 비정형 쿼리)에는 사용자 지정 처리가 필요합니다.
- 에이전트는 대화 전환판 역할을 하여 특수 에이전트 또는 마이크로서비스에 작업을 위임합니다.
- 이 워크플로는 도메인별 코파일럿, 고객 지원 봇, 엔터프라이즈 서비스 라우터 및 멀티모달 에이전트에서 일반적이며, 지능형 디스패치가 에이전트 동작의 품질과 효율성을 모두 결정합니다.

기능

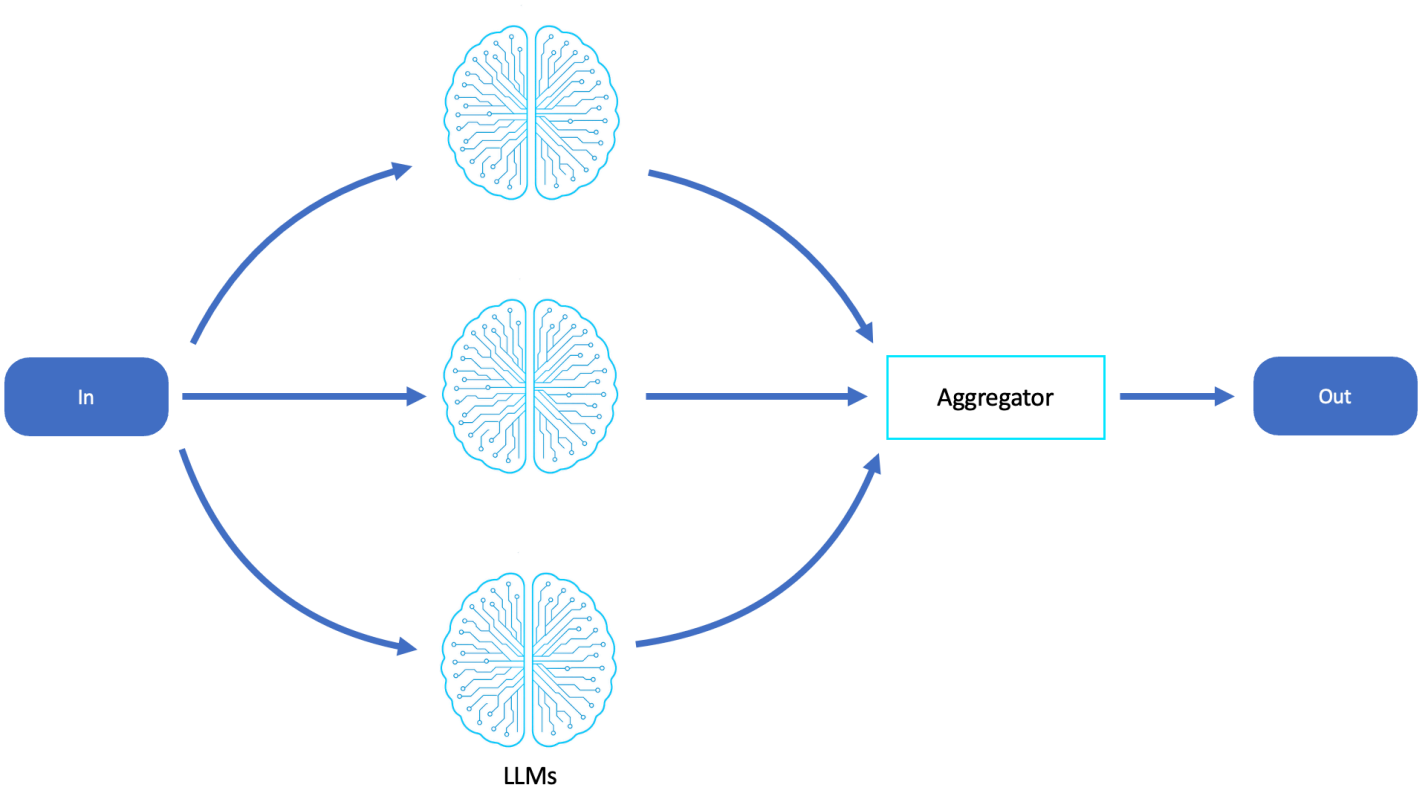
- 1차 LLM은 디스패처 역할을 합니다.
- 경로는 고유한 워크플로 또는 다른 에이전트 패턴을 호출할 수 있습니다.
- 기능의 모듈식 확장 지원

일반 사용 사례

- 다중 도메인 어시스턴트("법적, 의학적 또는 재정적 질문입니까?")
- LLM 추론으로 향상된 의사 결정 트리
- 동적 도구 선택(예: 검색 대 코드 생성)

병렬화를 위한 워크플로

이 워크플로에는 여러 LLM 호출 또는 에이전트가 동시에 처리할 수 있는 독립적인 하위 작업으로 작업을 분류하는 작업이 포함됩니다. 그런 다음 출력이 프로그래밍 방식으로 집계되고 결과로 합성됩니다.



병렬화 워크플로는 작업을 동시에 처리할 수 있는 독립적인 비순차 하위 작업으로 분할하여 효율성, 처리량 및 확장성을 크게 개선할 수 있는 경우에 사용됩니다. 에이전트가 여러 입력에 걸쳐 콘텐츠를 분석하거나 생성해야 하는 데이터 중심, 배치 지향 또는 다중 관점 문제 공간에서 특히 강력합니다.

병렬화는 다음과 같은 경우에 특히 효과적입니다.

- 하위 작업은 서로의 중간 결과에 의존하지 않으므로 조정 없이 병렬로 실행할 수 있습니다.
- 작업에는 여러 항목에서 동일한 추론 프로세스를 반복하는 작업이 포함됩니다(예: 여러 문서 요약 또는 옵션 목록 평가).
- 다양성, 창의성 또는 견고성을 높이기 위해 여러 가설 또는 관점을 동시에 탐색합니다.
- 동시 LLM 실행을 통해 대용량 또는 고주파 요청의 지연 시간을 줄여야 합니다.
- 이 워크플로는 문서 처리 에이전트, 설문 조사 또는 비교 엔진, 배치 요약기, 다중 에이전트 브레인스토밍, 확장 가능한 분류 또는 레이블 지정 작업, 특히 신속한 병렬 추론이 성능 이점인 경우에 일반적으로 사용됩니다.

기능

- LLM 작업의 병렬 실행(AWS Lambda AWS Fargate또는 AWS Step Functions 맵 상태 사용)

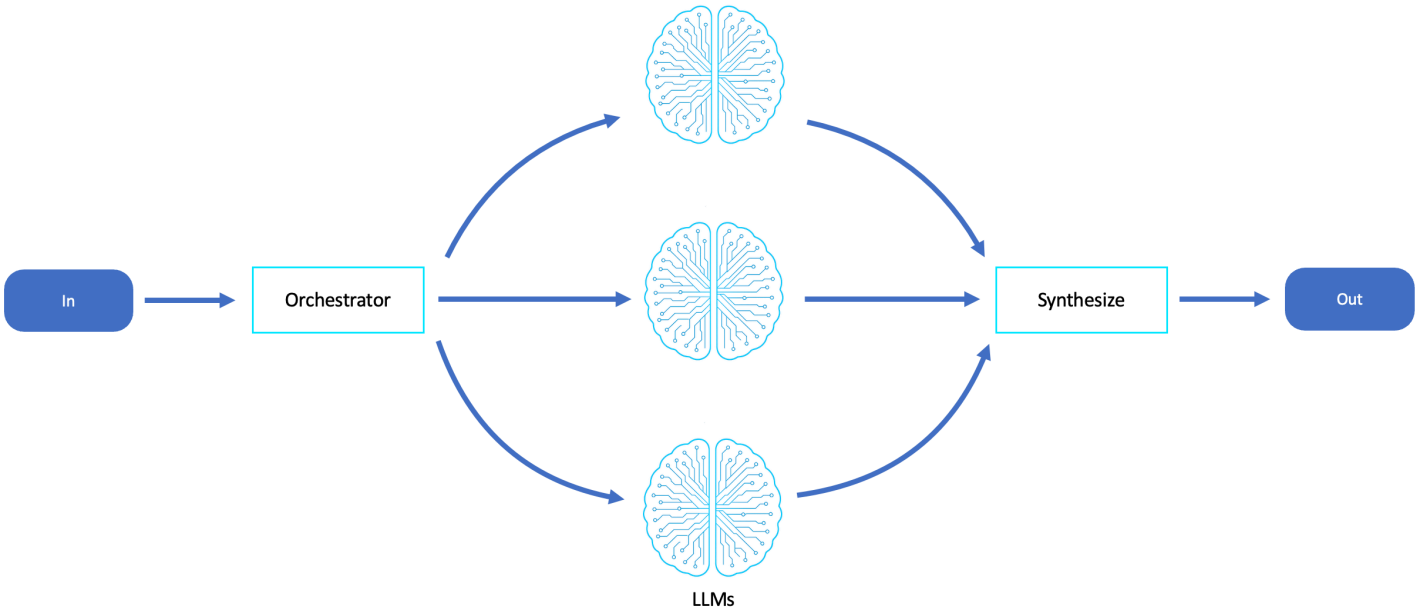
- 합성 단계에서 결과 정렬, 검증 또는 중복 제거 필요
- 상태 비저장 에이전트 루프에 적합

일반 사용 사례

- 여러 문서 또는 관점을 병렬로 분석
- 다양한 초안, 요약 또는 계획 생성
- 배치 작업 간 처리량 가속화

오케스트레이션을 위한 워크플로

중앙 오케스트레이터 에이전트는 LLM을 사용하여 각각 특정 역할 또는 도메인 전문 지식을 가진 전문 작업자 에이전트 또는 모델에 하위 작업을 계획, 분해 및 위임합니다. 이는 인적 팀 구조를 미러링하고 여러 에이전트의 긴급 동작을 지원합니다.



오케스트레이션 워크플로는 구조화된 분해 및 특수 실행이 필요한 복잡하거나 계층적 또는 다학제적인 시나리오에 적합합니다. 특히 작업의 다양한 하위 구성 요소를 고유한 기능, 지식 또는 도구 세트를 가진 에이전트가 처리하는 데 가장 적합한 작업 분할이 필요한 작업에 적합합니다.

이 워크플로는 다음과 같은 경우에 특히 효과적입니다.

- 작업은 범위, 유형 또는 추론(예: 계획, 연구, 구현 및 테스트)이 다양한 하위 작업으로 나눌 수 있습니다.

- LLM 또는 메타 에이전트는 다른 에이전트를 조정하고 진행 상황을 모니터링하며 결과를 합성해야 합니다.
- 에이전트 책임을 모듈화하여 확장성, 재사용 및 특수 조정을 지원하려고 합니다.
- 이 시스템에는 역할 기반 동작이 필요하며, 이는 인간 팀(예: 프로젝트 관리자, 개발자, 검토자)이 공동으로 작동하는 방식을 모방한 것입니다.

오케스트레이션은 멀티턴 계획 에이전트, 소프트웨어 개발 부조종사, 엔터프라이즈 프로세스 에이전트 및 자율 프로젝트 실행기에 적합합니다. 중앙 집중식 작업 분류가 필요하지만 분산 실행 로직이 필요한 다중 에이전트 시스템을 구현하여 에이전트 계층 간에 확장성과 설명 가능한 동작을 활성화할 때 특히 유용합니다.

기능

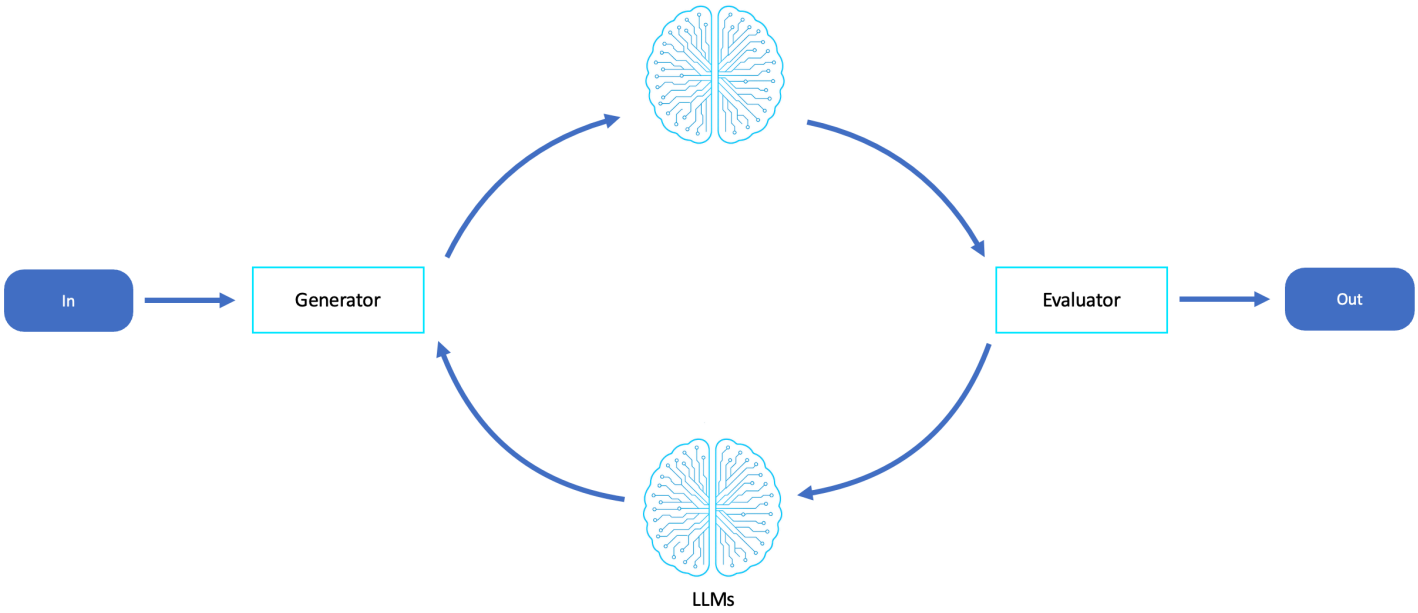
- 오케스트레이터가 목표 메타 이유 수행
- 작업자 에이전트에는 도구 액세스, 메모리 또는 도메인별 프롬프트가 포함될 수 있습니다.
- 계층적일 수 있음(다단계 작업 위임)

일반 사용 사례

- 프로젝트 관리자, 조정 연구원, 작성자 및 품질 보증 에이전트
- 계획, 실행 및 테스트를 결합하는 코파일럿 코딩
- 도구 체인 또는 API 액세스 패턴을 감독하는 에이전트

평가자 및 반사 구체화 루프를 위한 워크플로

이 워크플로는 한 LLM이 결과를 생성하고 다른 LLM이 결과를 평가하거나 평가하는 피드백 루프를 제공합니다. 이를 통해 자기 성찰, 최적화 및 반복적인 개선이 촉진됩니다.



평가자 워크플로는 출력 품질, 정확도 및 정렬이 중요하고 단일 패스 생성을 신뢰할 수 없거나 충분하지 않은 시나리오에 적합합니다. 이 워크플로는 에이전트가 더 높은 정확성 표준을 충족하거나 피드백을 기반으로 개선된 대안을 탐색하기 위해 자체 비판, 반복 및 개선해야 하는 경우에 유용합니다.

이 워크플로는 다음과 같은 경우에 특히 효과적입니다.

- 출력에는 주관적 품질 지표(예: 스타일, 어조 및 가독성) 또는 목표 기준(예: 정확성, 안전성 및 성능)이 포함됩니다.
- 에이전트는 절충을 통해 추론하거나, 제약 조건을 평가하거나, 목표를 향해 최적화해야 합니다.
- 특히 규제, 고객 대면 또는 크리에이티브 도메인에서 기본 제공되는 중복성과 품질 보증이 필요합니다.
- Human-in-the-loop 검토는 비용이 많이 들거나 사용할 수 없으며 자율 검증이 필요합니다.

이 워크플로는 콘텐츠 생성, 코드 합성 및 검토, 정책 적용, 정렬 검사, 명령 튜닝 및 RAG 사후 처리에 사용됩니다. 또한 지속적인 피드백이 시간이 지남에 따라 더 나은 응답을 형성하여 신뢰할 수 있고 자율적인 의사 결정 루프를 구축하는 데 도움이 되는 자체 개선 에이전트에도 유용합니다.

일반 사용 사례

- 블루 팀 에이전트와 비교한 레드 팀 에이전트
- 코드 또는 계획을 생성, 평가 및 수정하는 에이전트
- 품질 보증, 할루시네이션 감지 및 스타일 적용

기능

- 다양한 모델을 사용하여 분리된 생성 및 평가를 지원합니다(예: Claude for generation 및 Mistral for evaluation).
- 피드백은 구조화되어 있으며 수정된 출력을 표시하는 데 사용됩니다.
- 여러 반복 또는 수렴 임계값 지원

결론

LLMs 최신 소프트웨어 에이전트의 인지적 핵심을 제공하지만 원시 모델 호출만으로는 의도적이고 강력하며 제어 가능한 인텔리전스를 달성하기에 충분하지 않습니다. 출력 생성에서 구조화된 추론 및 목표 정렬 동작으로 이동하려면 모델이 입력을 처리하고 컨텍스트를 관리하며 작업을 조정하는 방법을 정의하는 의도적인 워크플로 패턴에 LLMs을 포함해야 합니다.

LLM 워크플로는 에이전트의 인지 모듈을 구축하기 위한 기반을 도입합니다.

- 프롬프트 체인은 복잡한 추론을 감사 가능한 모듈식 단계로 나눕니다.
- 라우팅을 통해 지능형 작업 분류 및 대상 위임이 가능합니다.
- 병렬화는 처리량을 가속화하고 다양한 추론을 촉진합니다.
- 에이전트 오케스트레이션은 작업 분해 및 역할 기반 실행을 통해 다중 에이전트 협업을 구성합니다.
- 평가자(반사 구체화 루프)를 사용하면 자체 개선, 품질 제어 및 정렬 검사를 수행할 수 있습니다.

각 워크플로는 에이전트의 요구 사항, 작업의 복잡성 및 사용자의 기대치에 맞게 조정할 수 있는 구성 가능한 패턴을 나타냅니다. 이러한 워크플로는 상호 배타적이지 않습니다. 이들은 종종 동적 추론, 다중 에이전트 조정 및 엔터프라이즈급 신뢰성을 지원하는 하이브리드 아키텍처에 결합되는 구성 요소입니다.

에이전트 워크플로 패턴에 대한 다음 장으로 전환하면 이러한 LLM 워크플로가 더 큰 시스템 내에 임베디드 구조로 다시 나타나 목표 위임, 도구 오케스트레이션, 의사 결정 루프 및 수명 주기 자율성을 지원합니다. 이러한 LLM 워크플로를 마스터하는 것은 텍스트뿐만 아니라 추론, 적응 및 의도적으로 행동하는 소프트웨어 에이전트를 설계하는 데 필수적입니다.

에이전트 워크플로 패턴

에이전트 워크플로 패턴은 모듈식 소프트웨어 에이전트를 구조화된 대규모 언어 모델(LLM) 워크플로와 통합하여 자율 추론 및 조치를 가능하게 합니다. 기존 서버리스 및 이벤트 기반 아키텍처에서 영감을 받은 이러한 패턴은 핵심 로직을 정적 코드에서 LLM 증강 에이전트로 전환하여 향상된 적응성과 상황별 의사 결정을 제공합니다. 이 진화는 기존 클라우드 아키텍처를 결정적 시스템에서 동적 해석 및 지능형 증강이 가능한 아키텍처로 변환하는 동시에 확장성 및 응답성의 기본 원칙을 유지합니다.

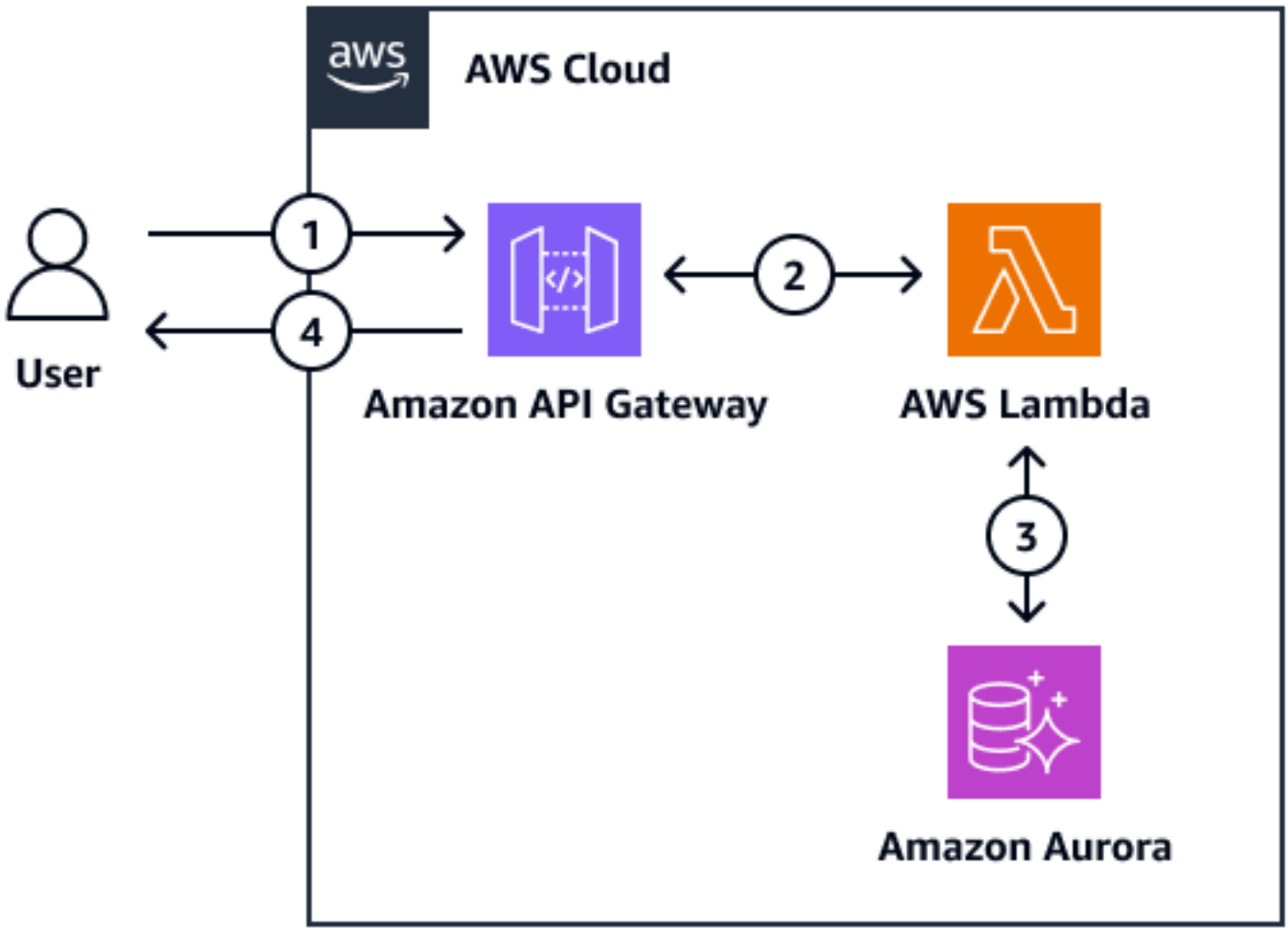
이 섹션의 내용

이벤트 기반 시스템에서 인지 증강 시스템으로

최신 클라우드 아키텍처, 특히 서버리스 및 이벤트 기반 원칙을 기반으로 구축된 아키텍처는 전통적으로 라우팅, 팬아웃 및 보강과 같은 패턴에 의존하여 응답성이 뛰어나고 확장 가능한 시스템을 만들었습니다. 에이전트 AI 시스템은 LLM 증강 추론 및 인지 유연성을 중심으로 리프레이밍하면서 이러한 기반을 기반으로 구축됩니다. 이 접근 방식을 사용하면 보다 정교한 문제 해결 및 자동화 기능을 통해 클라우드 환경에서 복잡한 작업을 처리하는 방법을 혁신할 수 있습니다.

이벤트 중심 아키텍처

다음 다이어그램은 일반적인 분산 시스템을 보여줍니다.

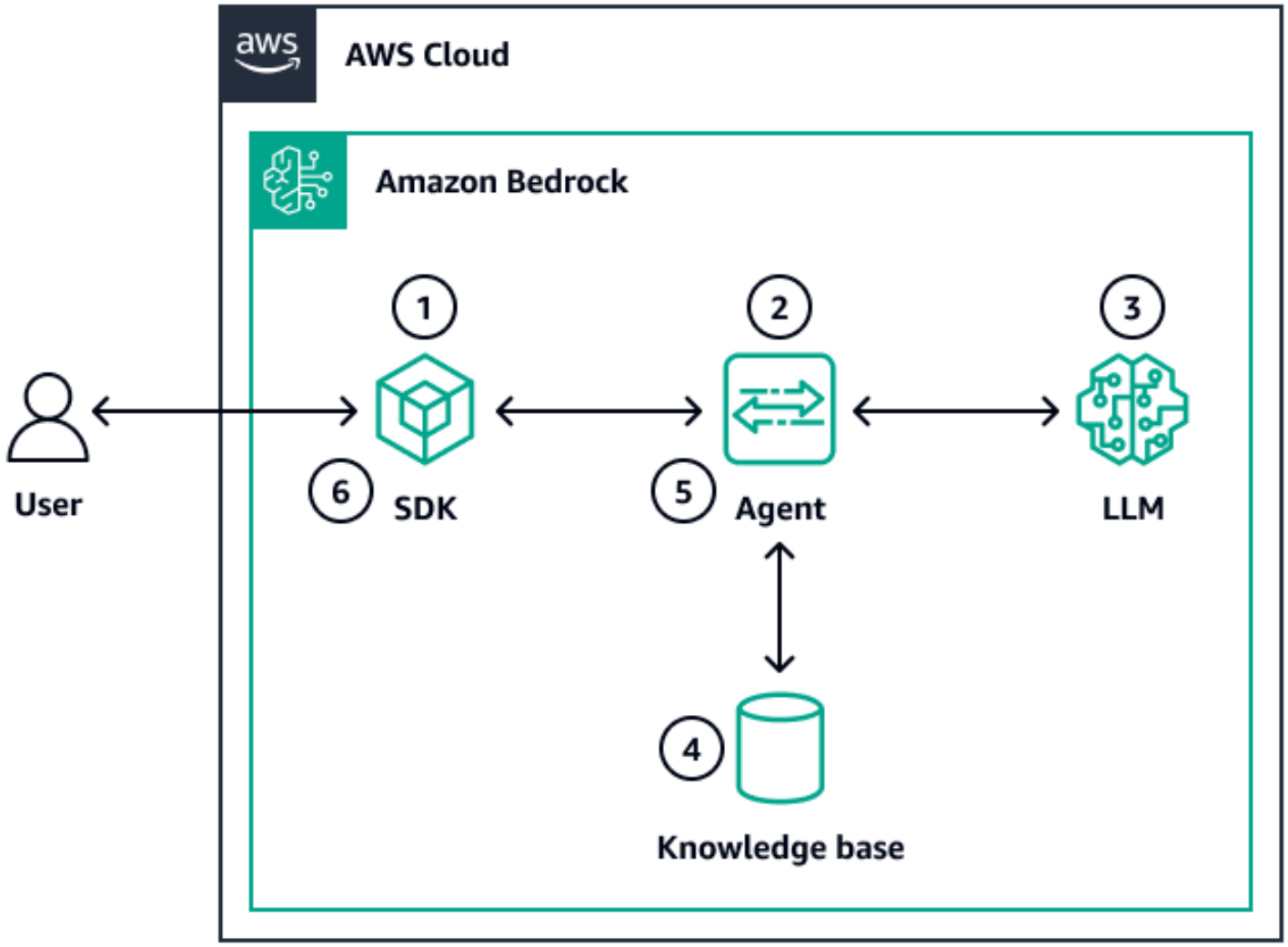


1. 사용자가 Amazon API Gateway에 요청을 제출합니다.
2. Amazon API Gateway는 요청을 AWS Lambda 함수로 라우팅합니다.
3. AWS Lambda 는 Amazon Aurora 데이터베이스를 쿼리하여 데이터 보강을 수행합니다.
4. Amazon API Gateway는 강화된 페이로드를 호출자에게 반환합니다.

이 구조는 안정적이고 확장 가능하지만 기본적으로 정적입니다. 비즈니스 규칙 및 로직 경로는 명시적으로 코딩되어야 하며 변화하는 컨텍스트 또는 불완전한 정보에 적응하는 것은 제한됩니다.

Cognition 증강 워크플로

에이전트 아키텍처는 이벤트 기반 시스템에 인지 증강을 추가합니다. 다음 다이어그램은 에이전트와 동등한 것을 보여줍니다.



1. 사용자가 SDK 또는 API 직접 호출을 통해 쿼리를 제출합니다.
2. Amazon Bedrock 에이전트가 쿼리를 수신합니다.
3. 에이전트는 LLM을 호출하여 쿼리를 해석합니다.
4. 에이전트는 Amazon Bedrock 지식 기반 또는 기타 외부 데이터 소스를 검색하여 의미 체계 보강을 수행합니다.
5. LLM은 컨텍스트가 풍부한 목표 정렬 응답을 합성합니다.
6. 시스템은 합성된 응답을 사용자에게 반환합니다.

이 흐름에서 LLM은 로직을 사용하고, 의도를 이해하고, 관련 컨텍스트를 검색 및 결합한 다음 가장 잘 대응하는 방법을 결정합니다. 이 패턴은 메시지가 추가 라우팅되기 전에 외부 데이터로 보강되는 기존 보강 패턴을 미러링합니다. 그러나 에이전트 시스템에서이 보강은 정적 조회가 아닙니다. 대신 보강은 동적이고 의미론적으로 안내되며 목적에 따라 구동됩니다.

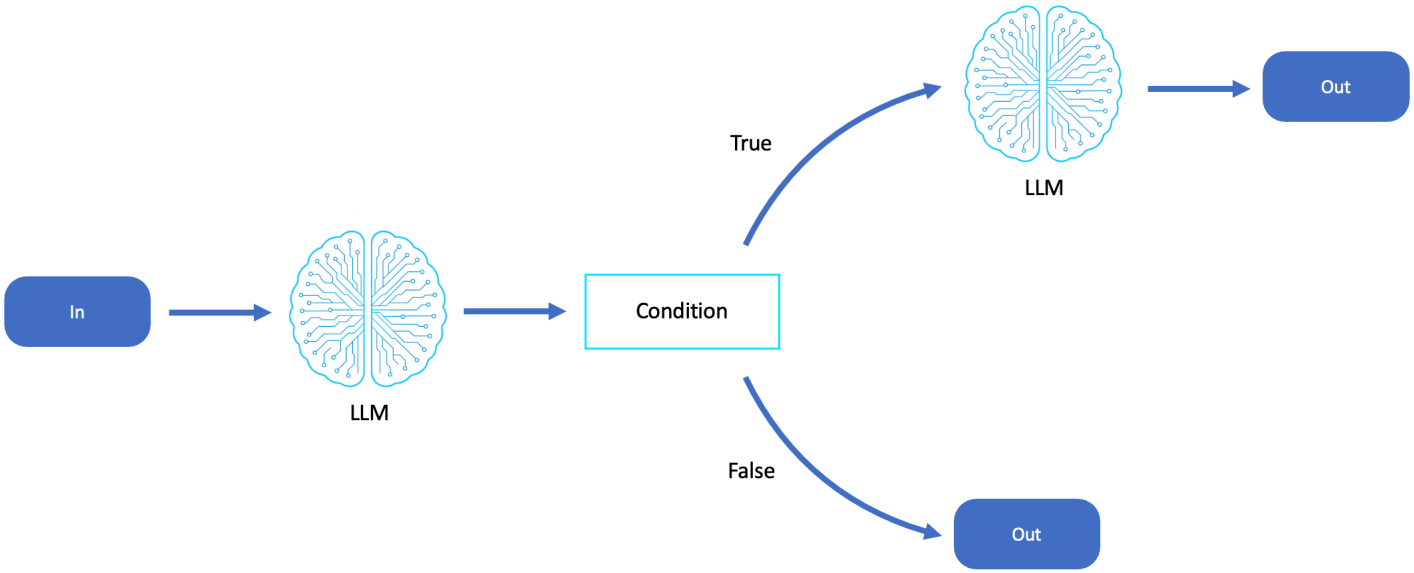
핵심 인사이트

각 LLM 워크플로는 기존의 이벤트 기반 아키텍처 스타일을 미러링하고 발전시키는 에이전트 워크플로 패턴에 매핑할 수 있습니다. 에이전트 워크플로의 기본 구성 요소는 데이터, 도구 및 메모리로 LLM의 컨텍스트를 보강하는 기능입니다. 이렇게 하면 정보에 입각하고 적응력이 뛰어나며 사용자 의도에 맞는 추론 루프가 생성됩니다. 기존 시스템이 조회 데이터로 메시지를 보강하는 경우 에이전트 시스템을 사용하면 소프트웨어가 스크립트처럼 작동하지 않고 지능형 공동 작업자처럼 작동할 수 있습니다.

프롬프트 체인 Saga 패턴

LLM 프롬프트 체인을 이벤트 기반 Saga로 재구상함으로써 워크플로가 자율 에이전트 간에 분산되고 복구 가능하며 의미론적으로 조정되는 새로운 운영 모델을 잠금 해제합니다. 각 프롬프트-응답 단계는 원자성 작업으로 재구성되고, 이벤트로 방출되며, 전용 에이전트가 사용하고, 컨텍스트 메타데이터로 보강됩니다.

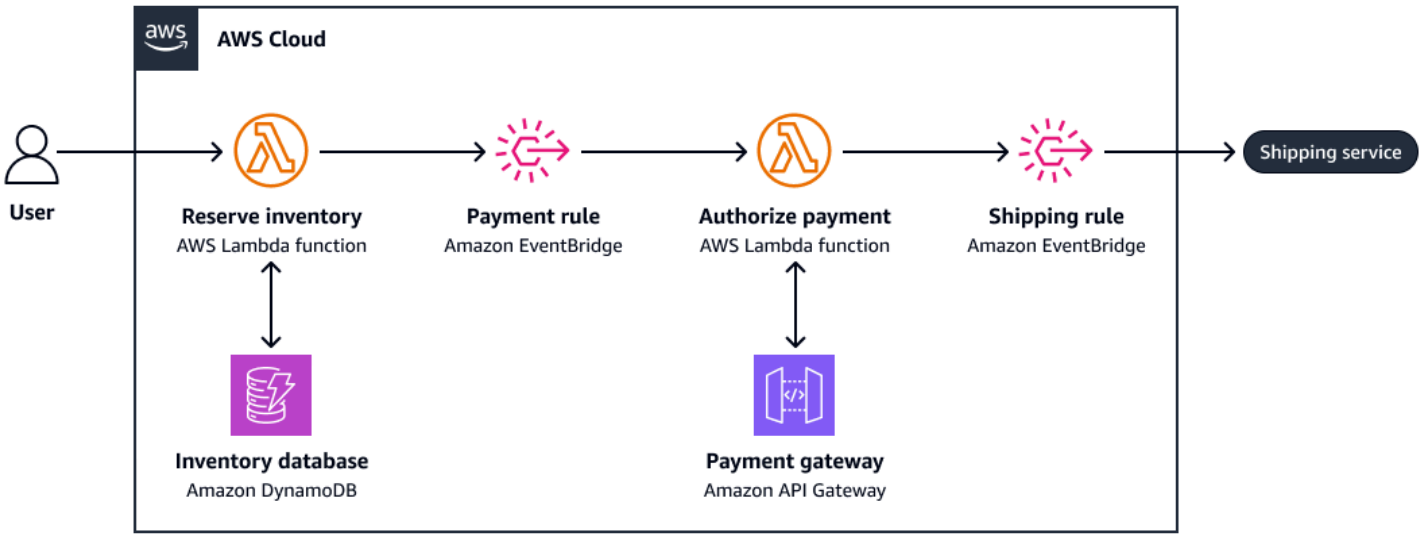
다음 다이어그램은 LLM 프롬프트 체인의 예입니다.



Saga 코레오그래피

Saga 코레오그래피 패턴은 중앙 조정자가 없는 분산 시스템의 구현 접근 방식입니다. 대신 각 서비스 또는 구성 요소는 다음 워크플로 작업을 트리거하는 이벤트를 게시합니다. 이 패턴은 분산 시스템에서 여러 서비스의 트랜잭션을 관리하는 데 널리 사용됩니다. Saga에서 시스템은 일련의 조정된 로컬 트랜잭션을 실행합니다. 실패하면 시스템은 일관성을 유지하기 위해 보정 작업을 트리거합니다.

다음 다이어그램은 Saga 코레오그래피의 예입니다.



1. 인벤토리 예약
2. 결제 권한 부여
3. 배송 주문 생성

3단계에 실패하면 시스템이 보상 작업을 호출합니다(예: 결제 취소 또는 인벤토리 릴리스).

이 패턴은 서비스가 느슨하게 결합되고 부분적으로 장애가 발생하더라도 시간이 지남에 따라 상태를 일관되게 해결해야 하는 이벤트 기반 아키텍처에서 특히 유용합니다.

프롬프트 체인 패턴

프롬프트 체인은 구조와 목적 모두에서 Saga 패턴과 유사합니다. 컨텍스트를 보존하고 롤백 및 개정을 허용하면서 순차적으로 빌드되는 일련의 추론 단계를 실행합니다.

에이전트 코레오그래피

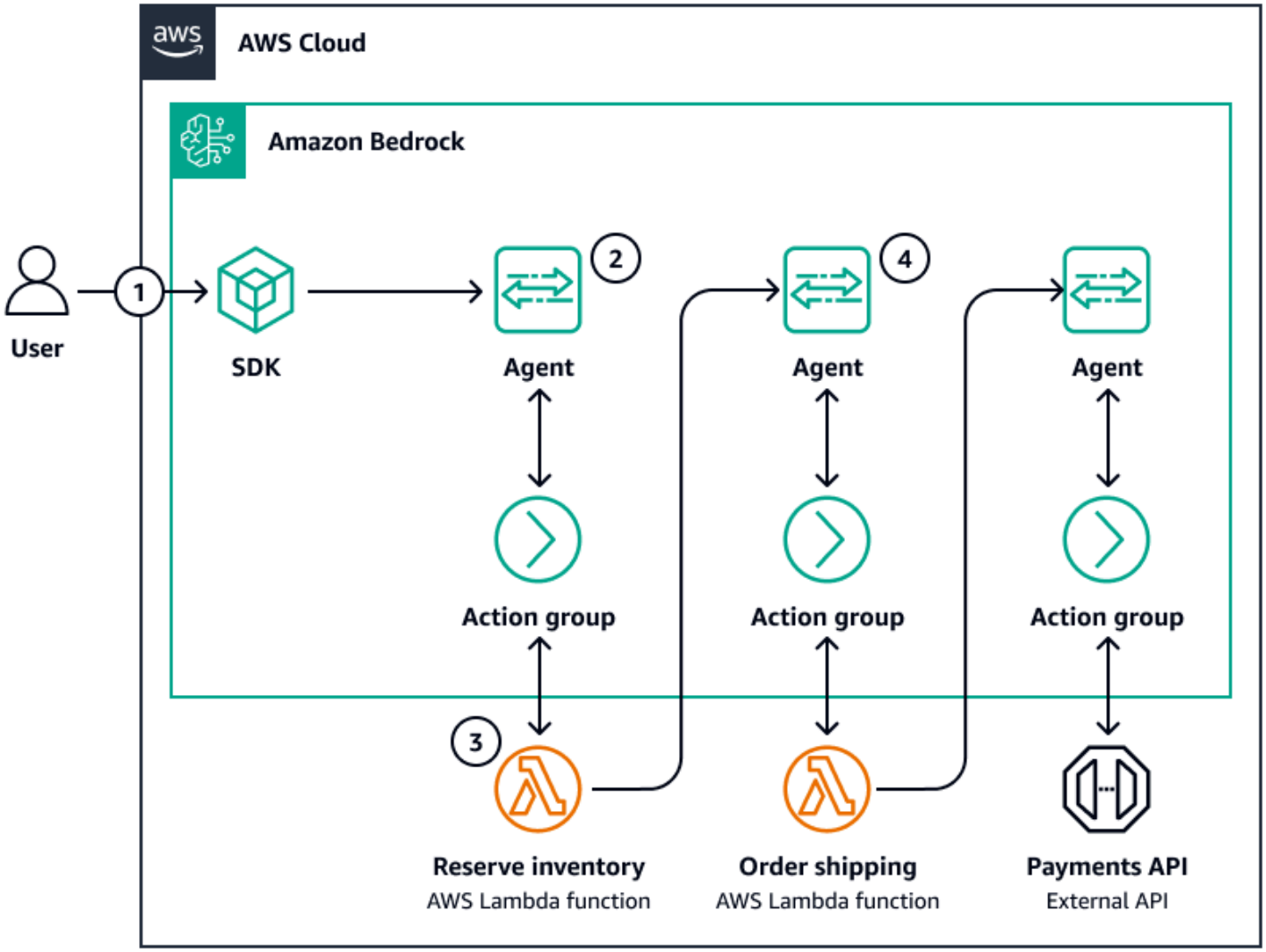
1. LLM은 복잡한 사용자 쿼리를 해석하고 가설을 생성합니다.
2. LLM은 작업을 해결하기 위한 계획을 자세히 설명합니다.
3. LLM은 하위 작업을 실행합니다(예: 도구 호출 사용 또는 지식 검색).
4. LLM은 결과를 만족스럽지 않다고 간주하는 경우 출력을 구체화하거나 이전 단계를 다시 검토합니다.

중간 결과에 결함이 있는 경우 시스템은 다음 중 하나를 수행할 수 있습니다.

- 다른 접근 방식을 사용하여 단계 재시도
- 이전 프롬프트로 되돌리고 다시 계획합니다.
- 평가자 루프(예: 평가자 최적화 패턴)를 사용하여 실패를 감지하고 수정합니다.

Saga 패턴과 마찬가지로 프롬프트 체인을 사용하면 부분 진행 및 롤백 메커니즘을 사용할 수 있습니다. 이는 데이터베이스 트랜잭션을 보정하는 대신 반복적인 개선 및 LLM 방향 수정을 통해 발생합니다.

다음 다이어그램은 에이전트 코레오그래피의 예입니다.



1. 사용자가 SDK를 통해 쿼리를 제출합니다.
2. Amazon Bedrock 에이전트는 다음을 통해 추론을 오케스트레이션합니다.
   - 해석(LLM)

- 계획(LLM)
- 도구 또는 지식 기반을 통한 실행
- 응답 구성

3. 도구가 실패하거나 불충분한 데이터를 반환하는 경우 에이전트는 작업을 동적으로 다시 계획하거나 다시 설명할 수 있습니다.
4. 메모리(예: 단기 벡터 스토어)는 여러 단계에서 상태를 유지할 수 있습니다.

요점

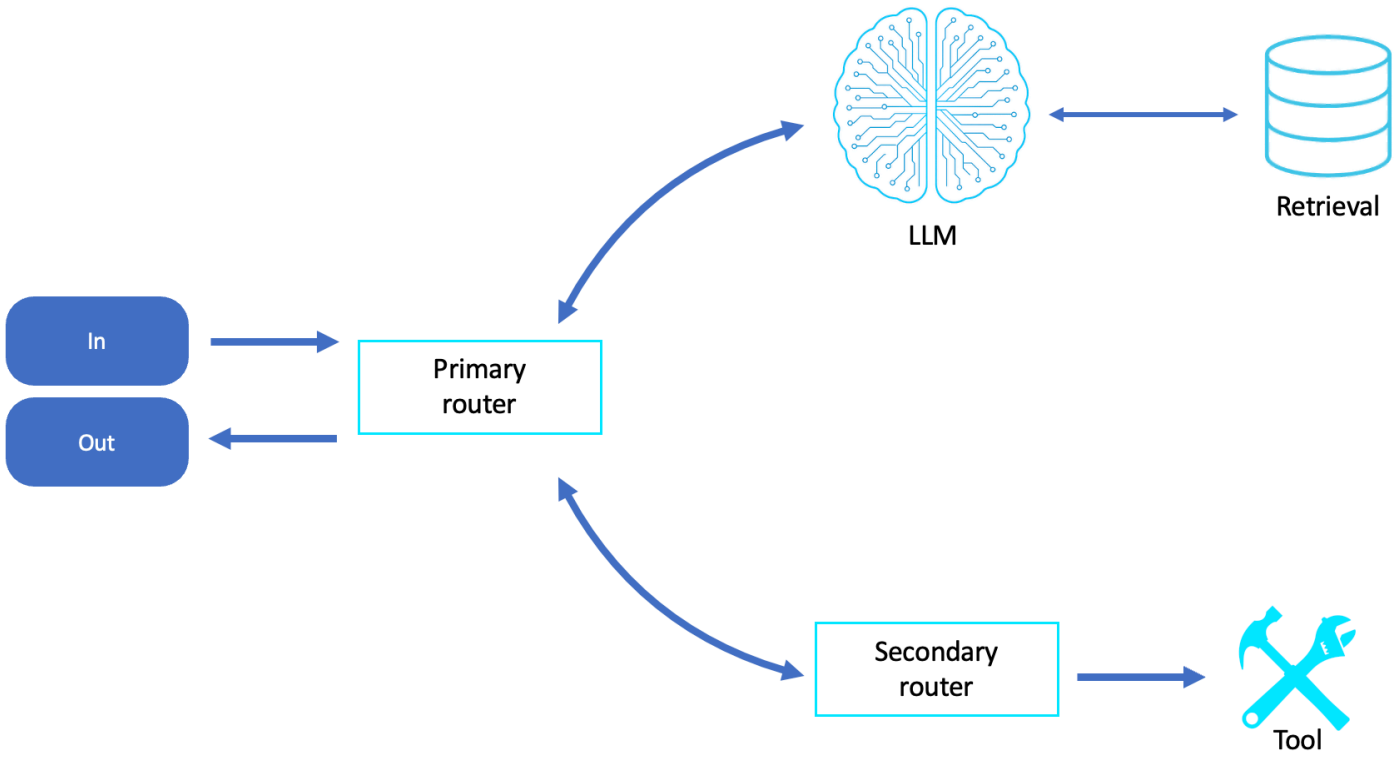
Saga 패턴이 보정 로직을 사용하여 분산 서비스 호출을 관리하는 경우 프롬프트 체인은 반사 시퀀싱 및 적응형 재계획을 통해 추론 작업을 관리합니다. 두 시스템 모두 점진적인 진행 상황, 분산된 결정 시점 및 장애 복구를 허용하며, 강체 롤백이 아닌 정보에 입각한 추론을 통해이 모든 것을 수행합니다.

프롬프트 체인은 사가스와 인지적으로 동등한 트랜잭션 추론을 도입합니다. 즉, 각 '사고'는 더 광범위한 목표 지향 대화의 일환으로 재평가, 수정 또는 중단됩니다.

동적 디스패치 패턴 라우팅

태스크가 문서 구문 분석부터 자율 소프트웨어 생성에 이르기까지 다양한 최신 에이전트 시스템에서는 요청을 가장 유능한 대규모 언어 모델(LLM) 또는 에이전트로 동적으로 라우팅하는 기능이 중요합니다. 오케스트레이션 스크립트 또는 API 계층에 포함되는 정적 라우팅 로직에는 실시간, 다중 모델, 다중 기능 환경에 필요한 적응성이 부족합니다. 이를 해결하기 위해 LLM 라우팅 워크플로를 동적 디스패치 패턴을 활용하는 이벤트 기반 아키텍처로 변환하여 LLM 호출을 지능적으로 라우팅되는 컨텍스트 인식 이벤트로 전환할 수 있습니다.

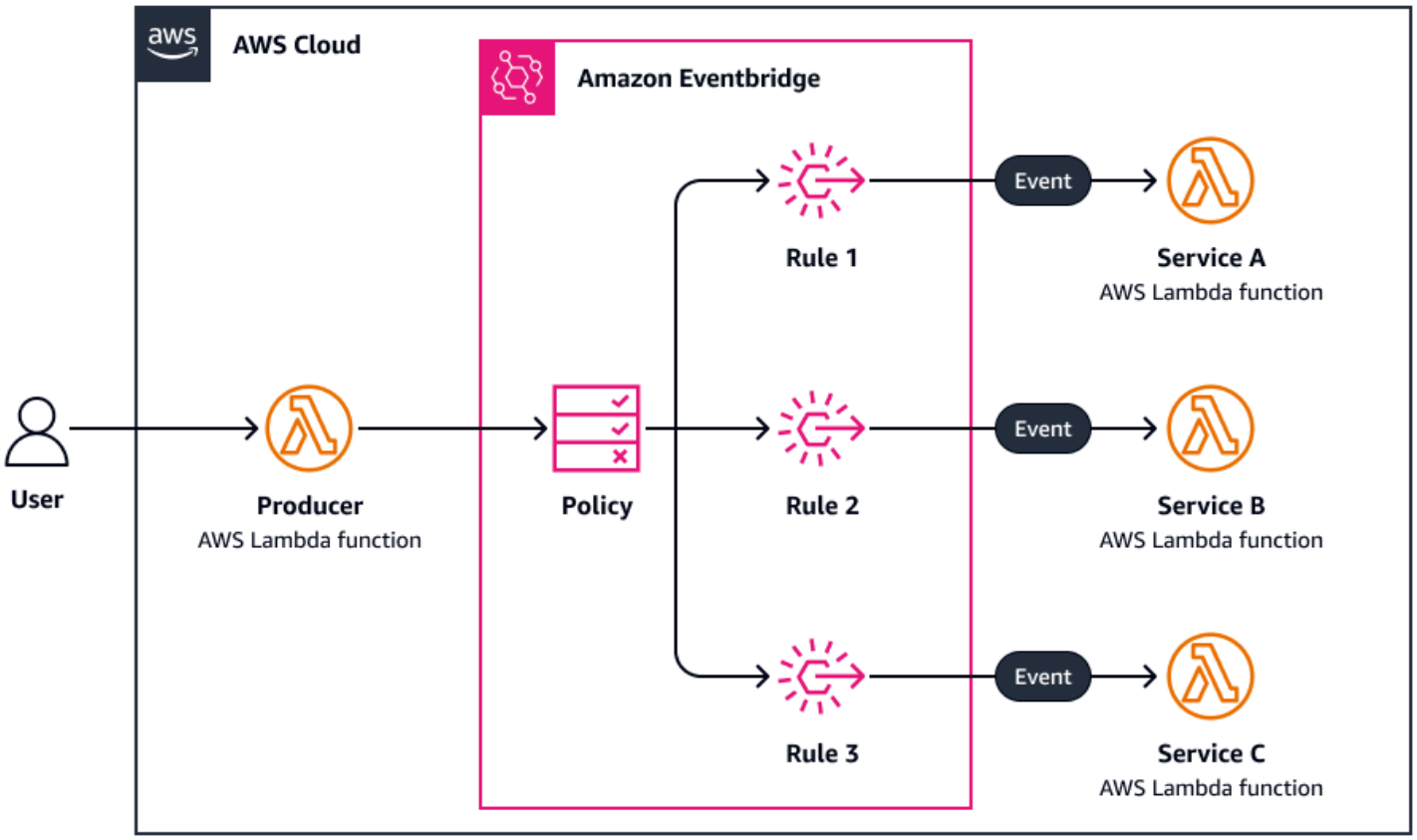
다음 다이어그램은 LLM 라우팅의 예입니다.



동적 디스패치

기존 분산 시스템에서 동적 디스패치 패턴은 이벤트 유형, 소스 및 페이로드와 같은 수신 이벤트 속성을 기반으로 런타임 시 특정 서비스를 선택하고 호출합니다. 이는 일반적으로 수신 이벤트를 평가하고 적절한 대상(예: AWS Lambda 함수 AWS Step Functions또는 Amazon Elastic Container Service 작업)으로 라우팅할 수 있는 Amazon EventBridge를 사용하여 구현됩니다.

다음 다이어그램은 동적 디스패치의 예입니다.



1. 애플리케이션은 이벤트(예: {"type": "orderCreated", "priority": "high"})를 내보냅니다.
2. Amazon EventBridge는 라우팅 규칙을 기준으로 이벤트를 평가합니다.
3. 이벤트의 속성에 따라 시스템은 다음과 같이 동적으로 디스패치합니다.
   - `HighPriorityOrderProcessor` (서비스 A)
   - `StandardOrderProcessor` (서비스 B)
   - `UpdateOrderProcessor` (서비스 C)

이 패턴은 느슨한 결합, 도메인 기반 전문화 및 런타임 확장성을 지원합니다. 이를 통해 시스템은 변화하는 요구 사항 및 이벤트 의미에 지능적으로 대응할 수 있습니다.

LLM 기반 라우팅

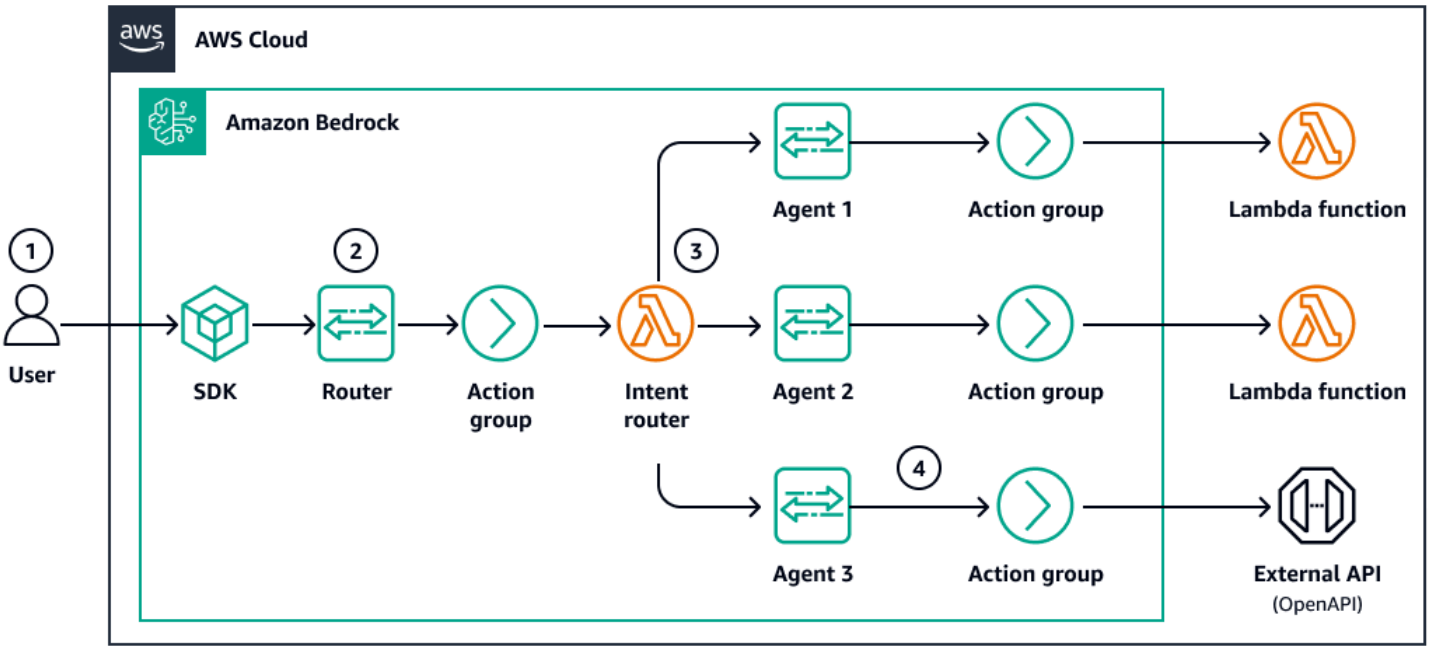
에이전트 시스템에서 라우팅은 동적 작업 위임도 수행하지만, Amazon EventBridge 규칙 또는 메타데이터 필터 대신 LLM은 자연어를 통해 사용자의 의도를 분류하고 해석합니다. 그 결과 유연하고 의미 체계적이며 적응형 형태의 디스패치가 이루어집니다.

에이전트 라우터

이 아키텍처를 사용하면 사전 정의된 스키마 또는 이벤트 유형 없이 풍부한 의도 기반 디스패치가 가능하므로 비정형 입력 및 복잡한 쿼리에 적합합니다.

1. 사용자가 "내 계약 조건을 검토할 수 있나요?" 요청을 제출합니다.
2. LLM은 이를 법적 문서 작업으로 해석합니다.
3. 에이전트는 작업을 다음 중 하나 이상으로 라우팅합니다.
   - 계약 검토 프롬프트 템플릿
   - 법적 추론 하위 에이전트
   - 문서 구문 분석 도구

다음 다이어그램은 에이전트 라우터의 예입니다.



1. 사용자가 SDK를 통해 자연어 요청을 제출합니다.
2. Amazon Bedrock 에이전트는 LLM을 사용하여 작업을 분류합니다(예: 법률, 기술 또는 일정).
3. 에이전트는 작업 그룹을 통해 작업을 동적으로 라우팅하여 필요한 에이전트를 호출합니다.
   - 도메인별 에이전트
   - 전문화된 도구 체인
   - 사용자 지정 프롬프트 구성

4. 선택한 핸들러가 작업을 처리하고 맞춤형 응답을 반환합니다.

요점

기존 동적 디스패치가 구조화된 이벤트 속성을 기반으로 라우팅하기 위해 Amazon EventBridge 규칙을 사용하는 경우 에이전트 라우팅은 LLMs 사용하여 의미와 의도를 기반으로 작업을 의미상 분류하고 라우팅합니다. 이렇게 하면 다음을 활성화하여 시스템의 유연성이 향상됩니다.

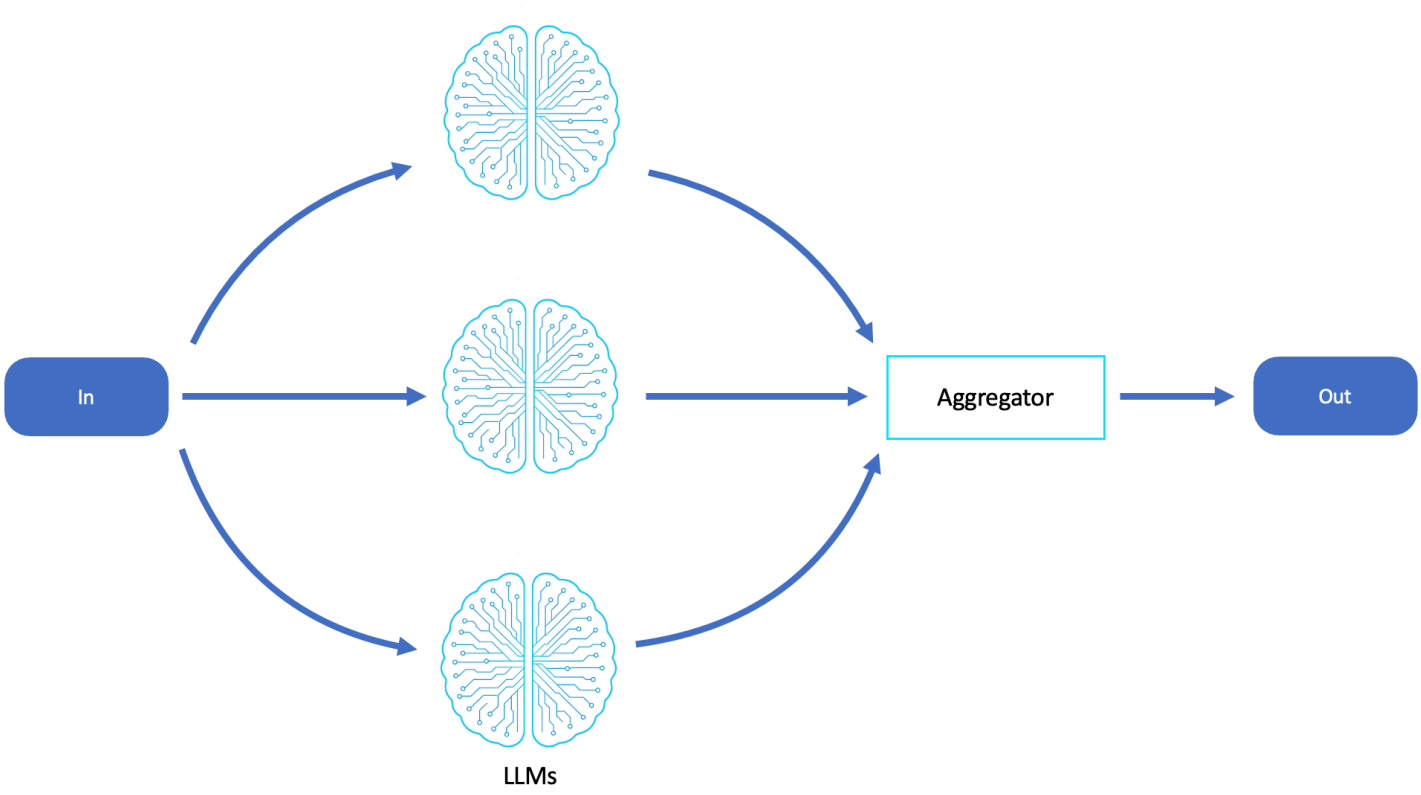
- 더 광범위한 입력 이해
- 지능형 대체 및 도구 선택
- 새로운 에이전트 역할 또는 프롬프트 스타일을 통한 자연스러운 확장성

에이전트 라우팅은 엄격한 규칙을 동적 인지 디스패치로 대체하므로 시스템이 코드가 아닌 언어에 따라 발전할 수 있습니다.

병렬화 및 산점도 수집 패턴

대규모 문서 요약, 여러 솔루션 경로 평가, 다양한 관점 비교와 같은 많은 고급 추론 및 생성 작업은 프롬프트의 병렬 실행의 이점을 누릴 수 있습니다. 확장성, 응답성 및 내결함성이 필요한 경우 기존 순차 워크플로는 부족합니다. 이를 해결하기 위해 이벤트 기반 산점 수집 패턴을 사용하여 LLM 기반 병렬화를 재구상할 수 있습니다.이 패턴에서는 작업이 자율 에이전트로 동적으로 팬아웃되고 결과가 지능적으로 합성됩니다.

다음 다이어그램은 LLM 병렬화 워크플로의 예입니다.



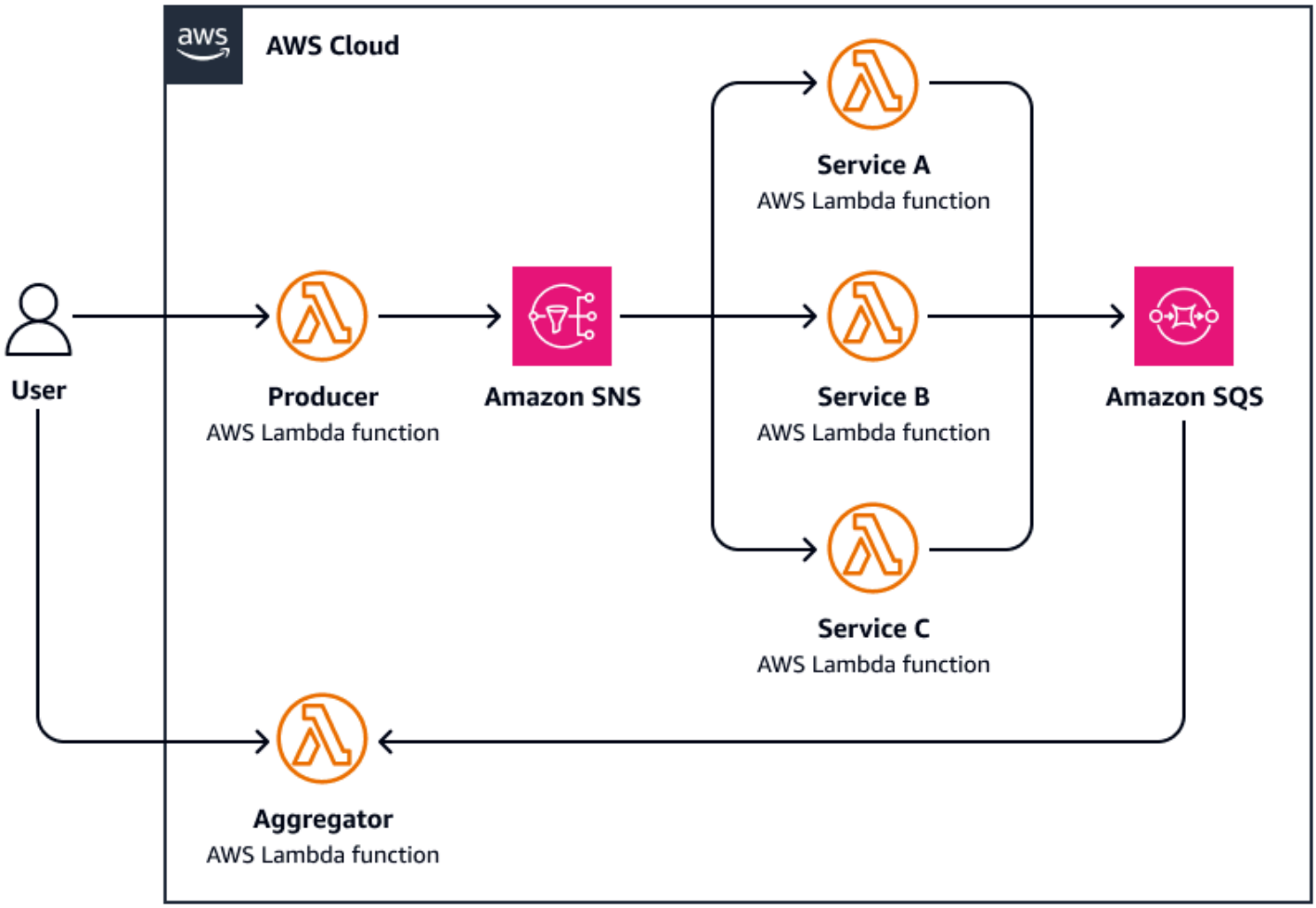
산점도 수집

분산 시스템에서 산점도 수집 패턴은 여러 서비스 또는 처리 유닛에 작업을 병렬로 전송하고 응답을 기다린 다음 결과를 통합 출력으로 집계합니다. 팬아웃과 달리 산점도 수집은 응답을 예상하고 일반적으로 결과를 결합, 비교 및 선택하는 로직을 적용하기 때문에 조정됩니다.

병렬화 및 산점도 수집을 위한 일반적인 구현은 다음과 같습니다.

- AWS Step Functions 병렬 작업 실행을 위한 상태 매핑
- AWS Lambda 동시성을 사용하여 여러 호출된 함수의 결과 조정
- 상관관계 IDs 및 집계 워크플로가 있는 Amazon EventBridge
- Amazon Simple Storage Service(Amazon S3), Amazon DynamoDB 또는 대기열을 사용하여 팬아웃을 관리하고 결과를 수집하는 사용자 지정 컨트롤러 패턴

다음 다이어그램은 산점도 수집의 예입니다.



1. 사용자는 Amazon Simple Notification Service(Amazon SNS) 주제에 병렬 메시지를 게시하여 작업을 분산시키는 요청을 중앙 조정자 함수로 보냅니다.
2. 각 메시지에는 작업 메타데이터가 포함되어 있으며 전문 작업자 에게 라우팅됩니다 AWS Lambda.
3. 각 작업자는 할당된 하위 작업(예: 외부 API 쿼리, 문서 처리, 데이터 분석)을 AWS Lambda 독립적으로 처리합니다.
4. 결과는 Amazon Simple Queue Service(Amazon SQS)와 같은 공통 스토리지 계층에 기록됩니다.
5. 애그리게이터 함수는 모든 응답이 완료될 때까지 기다린 다음 다음을 수행합니다.
   - 결과를 수집하고 집계합니다(예: 요약 병합, 가장 적합한 일치 항목 선택).
   - 최종 응답을 보내거나 다운스트림 워크플로를 트리거합니다.

산점도 수집 패턴의 일반적인 사용 사례는 다음과 같습니다.

- 페더레이션 검색

- 가격 비교 엔진
- 집계된 데이터 분석
- 다중 모델 추론

LLM 기반 병렬화(산점 수집 인식)

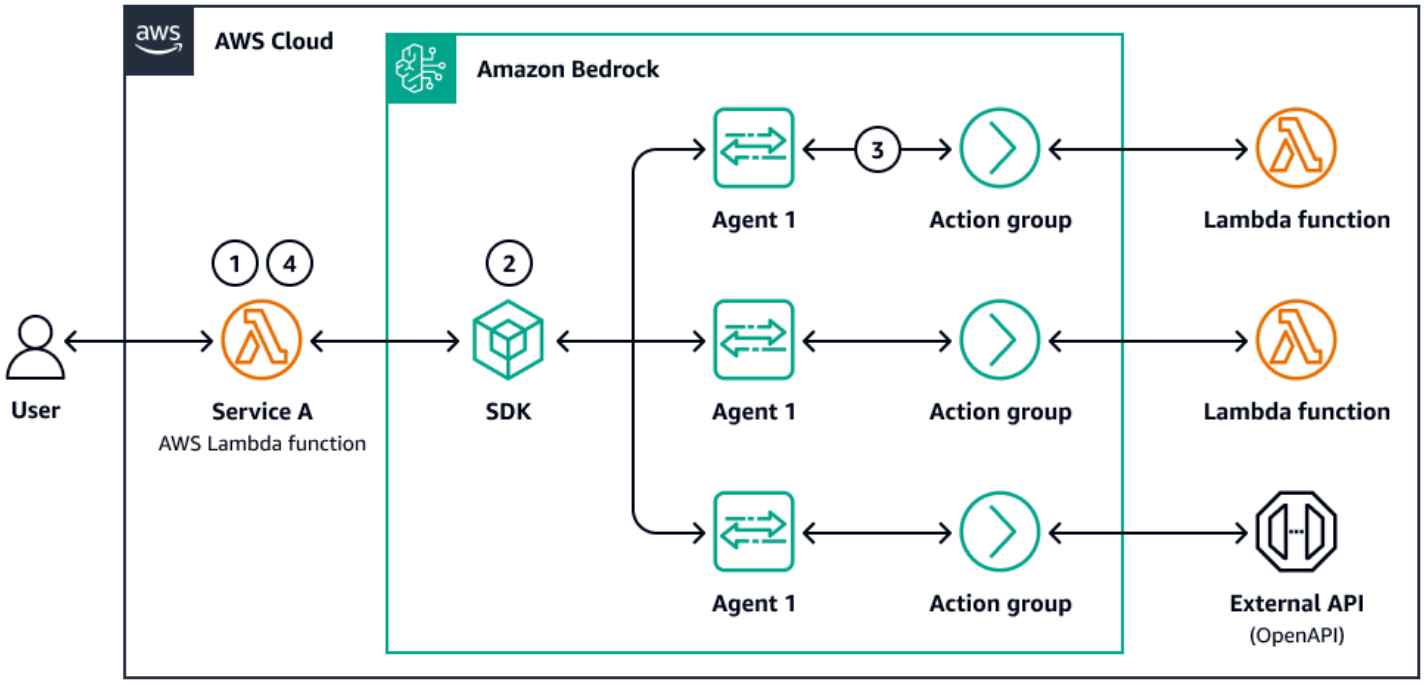
에이전트 시스템에서 병렬화는 하위 작업을 여러 LLM 호출 또는 에이전트에 분산하여 산란 수집을 면밀히 미러링하며, 각각 독립적으로 문제의 일부를 추론합니다. 반환된 결과는 종종 다른 LLM 또는 컨트롤러 에이전트인 집계 프로세스에 의해 수집되고 합성됩니다.

에이전트 병렬화

1. 에이전트가 "이 10개 보고서에 대한 인사이트 요약" 요청을 제출합니다.
2. 보고서를 10개의 병렬 LLM 요약 작업으로 분산합니다.
3. 모든 요약을 반환하면 에이전트는 다음을 수행합니다.
    - 요약을 통합 브리핑으로 집계합니다.
    - 테마 또는 모순 식별
    - 합성된 출력을 사용자에게 전송합니다.

이 에이전트 워크플로는 확장 가능한 모듈식 적응형 병렬 추론을 지원합니다. 이는 높은 인지 처리량이 필요한 사용 사례에 적합합니다.

다음 다이어그램은 에이전트 병렬화의 예입니다.



1. 사용자가 멀티파트 쿼리 또는 문서 세트를 제출합니다.
2. 컨트롤러 AWS Lambda 또는 단계 함수는 하위 작업을 분산합니다. 각 작업은 자체 프롬프트를 사용하여 Amazon Bedrock LLM 호출 또는 하위 에이전트를 호출합니다.
3. 호출 및 하위 작업이 완료되면 결과가 저장되고(예: Amazon S3 또는 메모리 스토어) 집계 단계에서 출력을 병합, 비교 또는 필터링합니다.
4. 시스템은 사용자 또는 다운스트림 에이전트에게 최종 응답을 반환합니다.

이 시스템에는 추적성, 내결함성 및 선택적 결과 가중치 또는 선택 로직이 포함된 분산 추론 루프가 있습니다.

요점

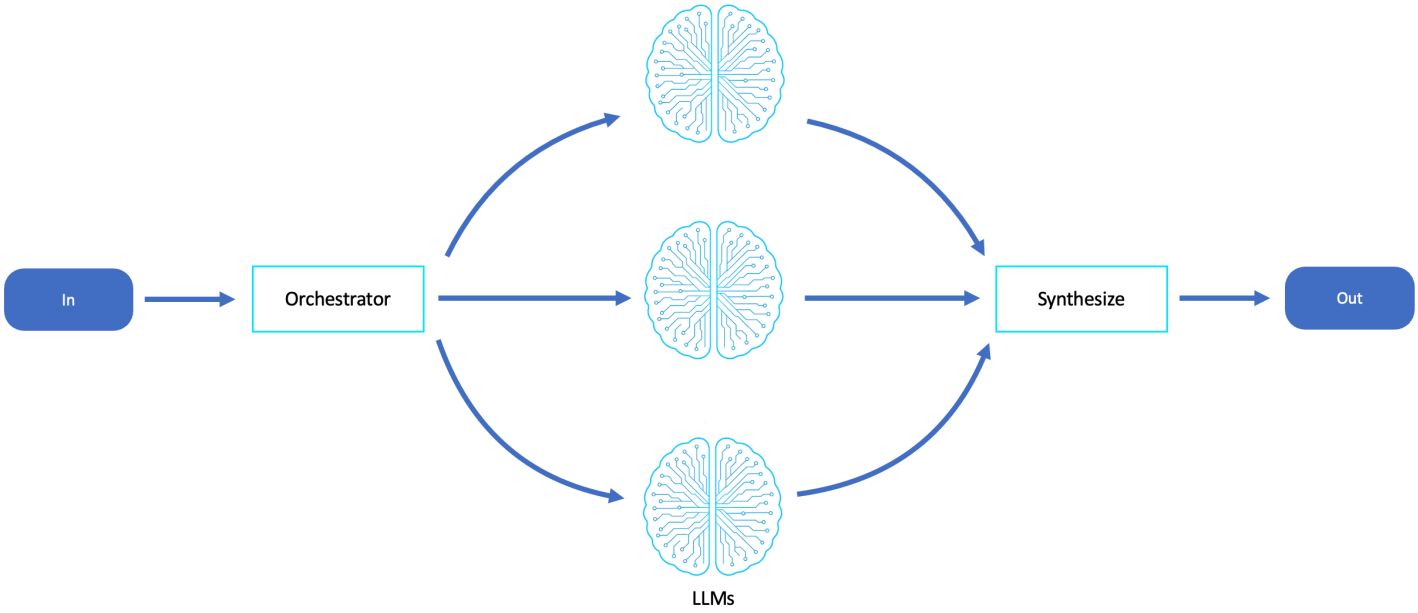
에이전트 병렬화는 산점도 수집 패턴을 사용하여 LLM 작업을 분산하므로 병렬 처리 및 지능형 결과 합성이 가능합니다.

Saga 오케스트레이션 패턴

프롬프트 체인, 데이터 처리 단계, 도구 호출 및 에이전트 협업에 걸쳐 LLMs으로 구동되는 워크플로가 점점 더 복잡해짐에 따라 지능형 오케스트레이션의 필요성이 필수적입니다. 긴밀하게 연결된 스크립트 또는 미리 결정된 정적 실행 흐름에 의존하는 대신 이러한 워크플로를 이벤트 기반 오케스트레이션

패턴으로 구현하여 LLM 기반 시스템이 자율 에이전트 간에 다단계 작업을 동적으로 조정, 모니터링 및 조정할 수 있습니다.

다음 다이어그램은 오케스트레이터의 예입니다.



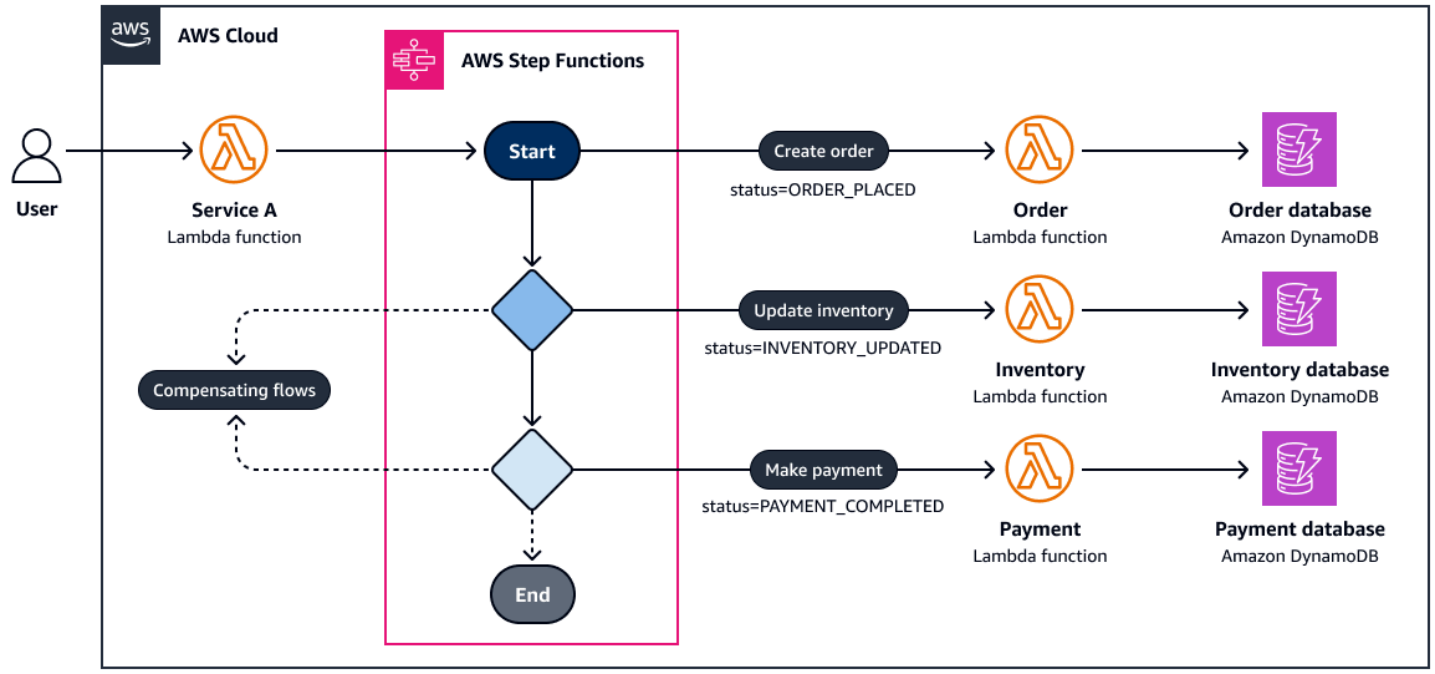
이벤트 오케스트레이션

기존 분산 시스템에서 이벤트 오케스트레이션은 중앙 코디네이터가 여러 서비스 또는 작업에 대한 제어 흐름을 명시적으로 지시하여 복잡한 워크플로를 관리하는 패턴을 말합니다. 이벤트 코레오그래피(각 서비스가 독립적으로 반응하는 경우)와 달리 오케스트레이션은 전체 프로세스에 대한 중앙 집중식 로직, 가시성 및 제어를 제공합니다.

이는 일반적으로 다음 도구를 사용하여 구현됩니다.

- AWS Step Functions - 상태 저장 워크플로 정의 및 실행
- AWS Lambda - 오케스트레이션된 흐름 내에서 개별 작업 수행
- Amazon SQS 또는 Amazon EventBridge - 비동기 단계 또는 응답을 트리거합니다.

다음 다이어그램은 Saga 오케스트레이션의 예입니다.



AWS Step Functions 워크플로는 고객 주문 프로세스를 관리합니다.

1. 주문 생성(AWS Lambda)
2. 인벤토리 업데이트(AWS Lambda)
3. 결제(AWS Lambda)

오케스트레이터는 재시도, 병렬 브랜치, 제한 시간 및 실패를 관리하여 각 단계를 조정합니다.

역할 기반 에이전트 시스템(오케스트레이터)

에이전트 시스템에서 오케스트레이터 패턴은 이벤트 오케스트레이션을 미러링하지만 각각 역할 또는 전문화가 정의된 여러 추론 에이전트에 로직을 분산합니다. 중앙 오케스트레이터 에이전트는 전체 작업을 해석하고, 하위 작업으로 분해하고, 작업자 에이전트에게 위임합니다. 각 에이전트는 특정 도메인(예: 연구, 코딩, 요약, 검토)에 최적화되어 있습니다.

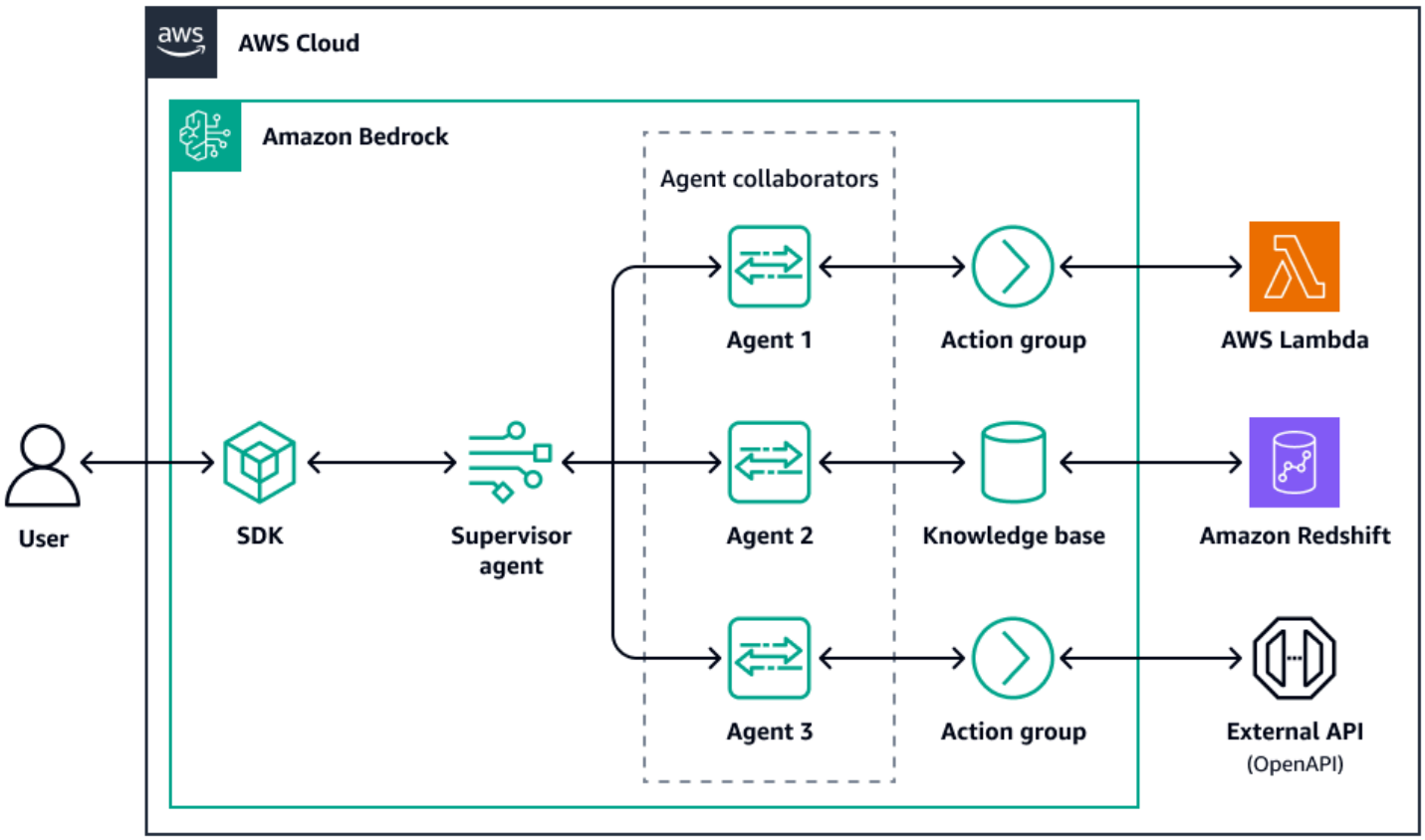
Supervisor

1. 사용자는 “프로젝트 개요 생성 및 상위 5개 경쟁자 요약” 쿼리를 제출합니다.
2. 오케스트레이터 에이전트는 다음을 수행합니다.
   - 연구 에이전트를 할당하여 경쟁자 데이터를 찾습니다.
   - 원시 조사 결과를 요약 에이전트로 보냅니다.

- 결과를 간이 라이터 에이전트에 전달합니다.
- 사용자의 최종 출력을 컴파일합니다.

각 에이전트는 독립적으로 작동하지만 오케스트레이터는 작업을 조정합니다. 이는 워크플로 작업을 처리하는 Lambda 함수와 같습니다.

다음 다이어그램은 감독자의 예입니다.



1. 사용자가 Amazon Bedrock 감독자 에이전트에게 작업을 제출합니다.
2. 감독자 에이전트는 요청을 각 에이전트 공동 작업자의 하위 작업으로 구문 분석합니다.
3. 각 하위 작업은 역할별 프롬프트 또는 도구 체인을 사용하여 공동 작업자 에이전트에 할당됩니다.
4. 작업자 에이전트는 작업 그룹을 통해 외부 APIs 또는 도구를 호출합니다.
5. 각 작업자 에이전트는 출력을 구조화된 형식으로 반환합니다.
6. 모든 작업자가 결과를 반환하면 감독자는 최종 응답을 평가, 합성 및 반환합니다.

이 구조를 사용하면 복잡한 다단계 에이전트 워크플로에서 모듈성, 적응성 및 내부 검사를 수행할 수 있습니다.

요점

이벤트 오케스트레이션이 중앙 집중식 제어(예: AWS Step Functions)를 사용하여 서비스 실행을 지시하는 경우 역할 기반 에이전트 시스템은 LLM 기반 오케스트레이터 에이전트를 사용하여 목표를 추론하고, 하위 작업을 작업자 에이전트에게 위임하고, 최종 출력을 합성합니다.

두 패러다임 모두에서 오케스트레이터는 다음을 수행합니다.

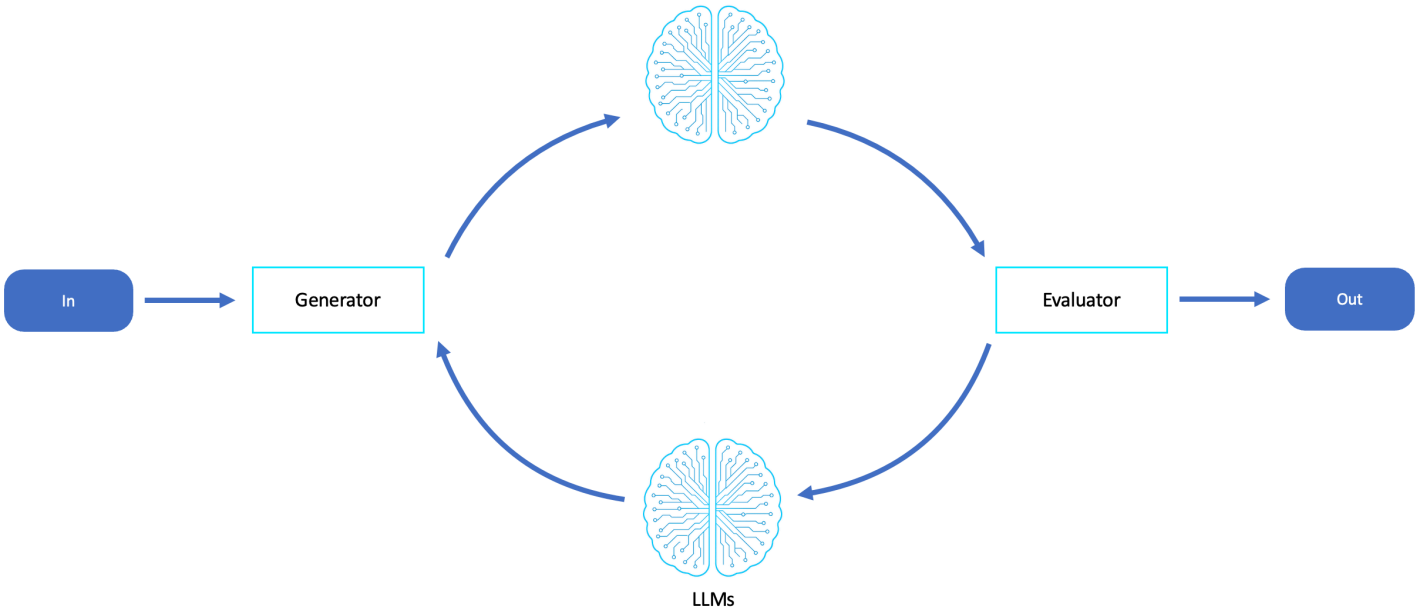
- 컨텍스트 및 실행 흐름을 유지합니다.
- 분기, 시퀀싱 및 오류 처리를 처리합니다.
- 분산 구성 요소에서 통합 결과를 생성합니다.

그러나 에이전트 오케스트레이션에는 추론, 적응성 및 의미 체계 위임이 추가됩니다. 따라서 개방형의 모호하고 진화하는 작업에 적합합니다.

평가자 반사 구체화 루프 패턴

코드 생성, 요약 또는 자율 의사 결정과 같은 작업은 런타임 피드백의 이점을 크게 활용하므로 시스템이 관찰 및 개선을 통해 발전할 수 있습니다. 이를 운영하기 위해 반사 세분화 주기를 자율적이고 지능적인 워크플로에 맞게 조정된 시스템 엔지니어링에서 영감을 받은 패턴인 이벤트 기반 피드백 제어 루프로 구현할 수 있습니다.

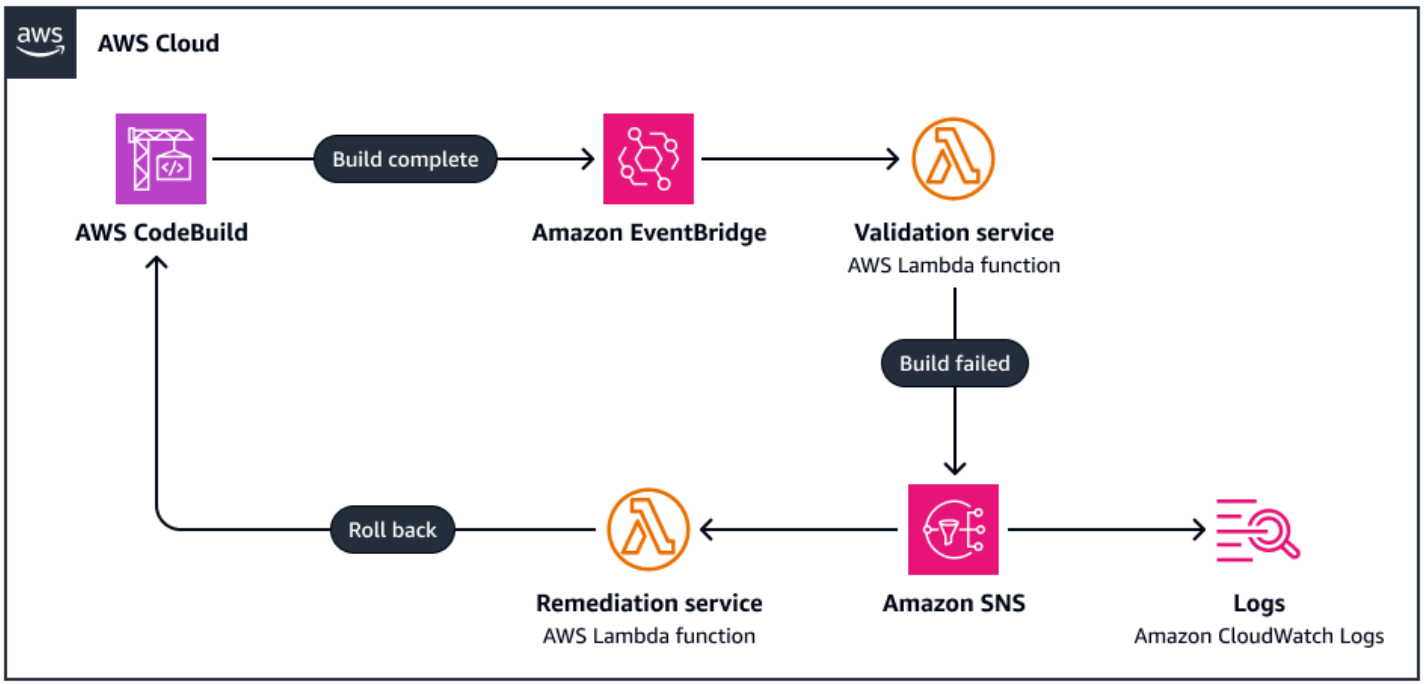
다음 다이어그램은 평가자 반영-구체화 피드백 루프의 예입니다.



피드백 제어 루프

피드백 제어 루프는 자체 출력 및 동작을 모니터링하고 정의된 기준 또는 원하는 상태를 기준으로 이를 평가한 다음 그에 따라 작업을 조정하는 패턴입니다. 이 아키텍처는 제어 이론에서 영감을 받았으며 자동화, 지속적 통합 및 지속적 전달(CI/CD) 파이프라인, 기계 학습 운영과 같은 도메인의 기본입니다.

다음 다이어그램은 피드백 제어 루프의 예입니다.



1. 배포 파이프라인은 buildComplete 이벤트를 내보냅니다.
2. 이벤트는 빌드를 검증하는 자동 테스트 또는 평가 작업을 트리거합니다.
3. 검증에 실패하면(예: 테스트 실패, 보안 문제 또는 정책 위반으로 인해) 시스템은 다음을 수행합니다.
   - buildComplete 이벤트를 내보냅니다.
   - 문제를 로깅하거나 알림을 보냅니다.
   - 롤백, 패치 적용 또는 재시도와 같은 수정 또는 수정 작업을 트리거합니다.

루프는 허용 가능한 결과 또는 에스컬레이션이 생성되거나 제한 시간이 발생할 때까지 계속됩니다. 이 패턴은 일반적으로 다음과 같은 경우에 사용됩니다.

- 이벤트를 평가 또는 수정 작업으로 라우팅하는 Amazon EventBridge 규칙
- AWS Step Functions 반복 재시도 로직 및 평가 결과에 대한 분기

- 피드백 트리거 및 알림에 대한 Amazon Simple Notification Service(Amazon SNS) 또는 Amazon CloudWatch 경보
- AWS Lambda 수정 조치를 적용하기 위한 함수 또는 컨테이너화된 작업자

피드백 제어 루프(평가자)

평가자 워크플로는 LLMs 또는 추론 에이전트로 구동되는 인지 피드백 루프입니다. 프로세스는 다음으로 구성됩니다.

1. 생성기 에이전트 또는 LLM은 출력(예: 계획, 답변 또는 초안)을 생성합니다.
2. 평가자 에이전트는 비평 프롬프트 또는 평가 마찰을 사용하여 결과를 검토합니다.
3. 피드백을 기반으로 원래 에이전트 또는 새 옵티마이저 에이전트가 출력을 수정합니다.

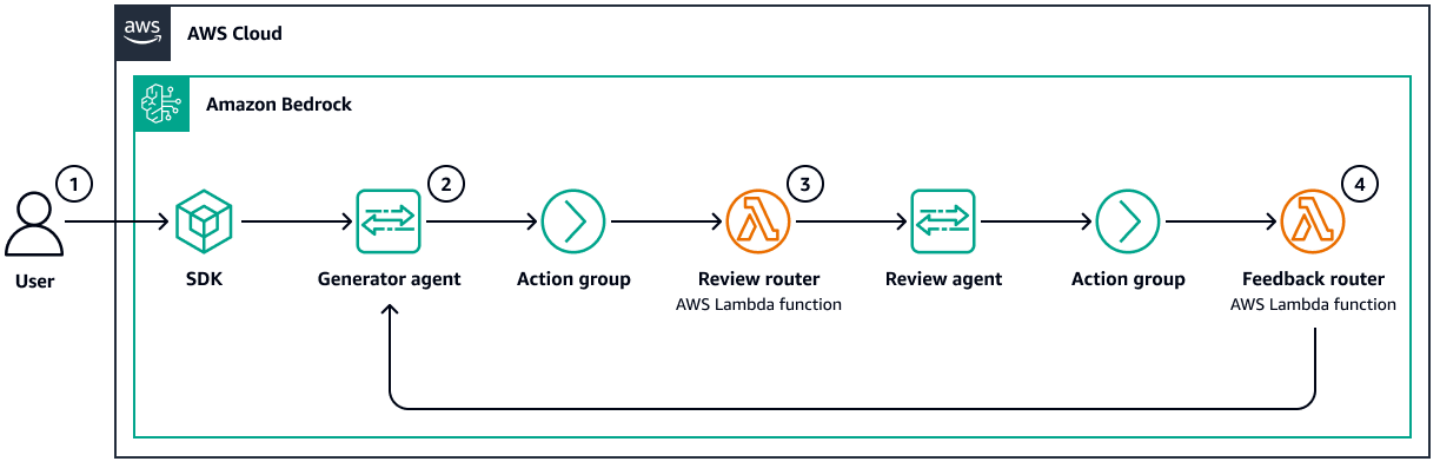
루프는 결과가 기준 집합을 충족하거나 승인되거나 재시도 제한에 도달할 때까지 반복됩니다.

평가자

1. 사용자가 에이전트에게 정책 요약을 작성하도록 요청합니다.
2. 생성기 에이전트가 초안을 작성합니다.
3. 평가자 에이전트는 적용 범위, 어조 및 법적 정확성을 확인합니다.
4. 응답이 부적절한 경우 피드백 루프가 수렴될 때까지 수정되고 다시 제출됩니다.

이를 통해 인적 입력 없이 자체 평가, 반복적 구체화 및 적응형 출력 제어가 가능합니다.

다음 다이어그램은 피드백 제어 루프(평가자)의 예입니다.



1. 사용자가 작업을 실행합니다(예: 비즈니스 전략 초안 작성).
2. Amazon Bedrock 에이전트는 LLM을 사용하여 초기 초안을 생성합니다.
3. 두 번째 에이전트(또는 후속 프롬프트)는 구조화된 평가를 수행합니다(예: "명확성, 완전성 및 어조로이 출력 평가").
4. 등급이 임계값 아래로 떨어지면 다음과 같이 응답이 수정됩니다.
   - 임베디드 비평으로 생성기 다시 호출
   - 특수 정련소 에이전트에게 피드백 전송
   - 허용 가능한 응답에 도달할 때까지 반복

AWS Lambda 컨트롤러 또는와 같은 선택적 구성 요소는 피드백 임계값, 재시도 및 대체 전략을 관리할 AWS Step Functions 수 있습니다.

요점

기존 피드백 제어 루프가 이벤트, 지표 및 수정 로직을 사용하여 시스템 동작을 검증하고 조정하는 경우 에이전트성 평가자 루프는 추론 에이전트를 사용하여 출력을 동적으로 평가, 반영 및 수정합니다.

두 패러다임 모두에서:

- 출력은 생성된 후 평가됩니다.
- 피드백을 기반으로 수정 또는 개선 작업이 트리거됩니다.
- 시스템이 대상 품질 또는 목표에 지속적으로 적응함

에이전트 버전은 정적 검증을 의미론적 반영으로 변환하여 자체 효과를 평가하는 자체 개선 에이전트를 지원합니다.

에서 에이전트 워크플로 설계 AWS

이 가이드의 각 패턴은를 사용하여 빌드할 수 있습니다 AWS 서비스. Amazon Bedrock 에이전트는 오케스트레이션, 데이터 액세스 및 상호 작용 채널을 제공합니다.

구성 요소	AWS 서비스	용도
LLM 추론	Amazon Bedrock	에이전트 로직, 계획, 도구 사용

도구 실행	AWS Lambda, Amazon ECS, Amazon SageMaker	에이전트를 위한 외부 도구 호스팅
메모리 및 RAG	Amazon Bedrock 지식 기반, Amazon S3, OpenSearch	영구 및 의미 체계 메모리
오케스트레이션	AWS Step Functions	다단계 작업 및 에이전트 조정
이벤트 라우팅	Amazon EventBridge, Amazon SQS	분리된 에이전트 간 메시징
사용자 인터페이스	Amazon API Gateway, AWS AppSync, SDK	애플리케이션 또는 시스템의 진입점
모니터링	Amazon CloudWatch, AWS X-Ray, AWS Distro for OpenTelemetry	관찰성 및 에이전트 내부 검사

결론

에이전트 워크플로 패턴은 이벤트 기반 아키텍처의 차세대 진화 단계로, 비즈니스 로직은 정적으로 정의되지 않지만 대규모 언어 모델(LLM) 강화 인식을 사용하여 동적으로 추론됩니다. 기존의 클라우드 네이티브 프리미티브와 LLM 워크플로 및 에이전트 설계 패턴을 결합하여 조직은 목적에 따라 대응하고 경험을 통해 학습하는 적응형 지능형 모듈식 시스템을 구축할 수 있습니다.

이러한 패턴에서 Amazon Bedrock은 에이전트 인식의 게이트웨이로, LLM 기반 에이전트가 이벤트 워크플로에 액세스하고 도구 및 메모리와 상호 작용하며 구조화되고 추적 가능하며 정렬된 결과를 제공할 수 있도록 합니다.

에이전트 시스템을 설계하고 배포할 때 이러한 워크플로 패턴은 구성 가능한 자율 AI 아키텍처를 구축하기 위한 청사진을 제공합니다. 이러한 시스템은 서버리스 모범 사례를 기반으로 하며 지능형 파운데이션 모델로 보강됩니다.

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. 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 문제 해결 작업의 모음입니다. YAML 템플릿을 사용하여 적합성 팩을 AWS 계정 및 리전 또는 조직 전체에 단일 엔터티로 배포할 수 있습니다. 자세한 내용은 AWS Config 설명서의 적합성 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 Framework에서 보안 원칙의 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 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 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 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.