구현 안내서

AWS의 생성형 AI 애플리케이션 빌더

AWS의 생성형 AI 애플리케이션 빌더: 구현 안내서

Table of Contents

이 솔루션은 생성형 인공 지능(AI) 애플리케이션의 개발, 신속한 실험 및 배포를 용이하게 합니다.

AWS의 생성형 AI Application Builder는 AI에 대한 심층적인 경험 없이 생성형 인공 지능(AI) 애플리케이션의 개발, 신속한 실험 및 배포를 용이하게 합니다. 이 AWS 솔루션은 다음을 지원하여 개발을 가속화하고 실험을 간소화합니다.

- 비즈니스별 데이터 및 문서 수집
- 대규모 언어 모델(LLMs)의 성능 평가 및 비교
- AI 에이전트를 사용하여 다단계 작업 및 워크플로 실행
- 확장 가능한 애플리케이션을 신속하게 구축하고 엔터프라이즈급 아키텍처를 사용하여 해당 애플리케이션을 배포합니다.

AWS의 생성형 AI Application Builder에는 다음과의 통합이 포함됩니다.

- Amazon Bedrock에서 사용 가능한 LLMs
- Amazon SageMaker AI에 배포한 LLMs
- 검색 증강 생성(RAG)을 위한 Amazon Bedrock 지식 기반
- Amazon Bedrock 가드레일을 통한 보호 구현 및 할루시네이션 감소
- Amazon Bedrock 에이전트 - 작업 오케스트레이션 및 완료를 수행할 수 있는 에이전트 워크플로 구축
- Amazon Bedrock AgentCore, 확장된 런타임 지원으로 프로덕션 지원 AI 에이전트 구축, 배포 및 관리
- 엔터프라이즈 데이터 및 도구 통합을 위한 모델 컨텍스트 프로토콜(MCP) 서버

또한이 솔루션을 사용하면 LangChain 커넥터를 사용하여 선택한 모델에 연결할 수 있습니다. 이러한 커넥터는 솔루션과 함께 배포되는 AWS Lambda 함수에서 사용할 수 있습니다. 노코드 배포 마법사로 시작하여 대화형 검색, AI 생성 챗봇, 텍스트 생성 및 텍스트 요약을 위한 생성형 AI 애플리케이션을 구축할 수 있습니다.

이 구현 가이드는 AWS의 생성형 AI Application Builder 솔루션, 참조 아키텍처 및 구성 요소, 배포 계획 고려 사항, Amazon Web Services(AWS) 클라우드에 솔루션을 배포하기 위한 구성 단계에 대한 개요를 제공합니다.

이 가이드는 환경에서 AWS에서 생성형 AI Application Builder를 구현하려는 솔루션 아키텍트, 비즈니스 의사 결정권자, DevOps 엔지니어, 데이터 과학자 및 클라우드 전문가를 대상으로 합니다.

이 탐색 테이블을 사용하여 다음 질문에 대한 답을 빠르게 찾을 수 있습니다.

다음을 수행하려는 경우 . . .	읽기 . . .
이 솔루션을 실행하는 데 드는 비용을 파악합니다. 이 솔루션을 실행하는 데 드는 예상 비용은 배포하는 구성 요소와 쿼리 수에 따라 달라집니다. 한 달 동안 미국 동부(버지니아 북부) 리전에서 기본 파라미터와 100명의 활성 사용자를 사용하여 배포 대시보드를 실행하는 데 드는 비용은 매월 약 20.12 USD입니다. LLM을 사용하여 하루에 100개의 쿼리를 수행하는 비즈니스 사용자 1명에 대해 RAG 없이 배포된 텍스트 사용 사례 비용은 매월 약 12.39 USD입니다. 하루에 8,000건의 상호 작용을 지원하는 Amazon Kendra 인덱스가 있는 RAG 지원 사용 사례의 비용은 매월 약 204.26 USD에 지식 기반 비용이 추가됩니다.	비용
이 솔루션의 보안 고려 사항을 이해합니다.	보안
이 솔루션의 할당량을 계획하는 방법을 파악합니다.	할당량
이 솔루션을 지원하는 AWS 리전을 파악합니다.	지원되는 AWS 리전
이 솔루션에 포함된 AWS CloudFormation 템플릿을 보거나 다운로드하여 이 솔루션의 인프라 리소스(“스택”)를 자동으로 배포합니다.	AWS CloudFormation 템플릿

다음을 수행하려는 경우 . . .	읽기 . . .
소스 코드에 액세스하고 선택적으로 AWS Cloud Development Kit(AWS CDK)를 사용하여 솔루션을 배포합니다.	GitHub 리포지토리

기능 및 이점

AWS의 생성형 AI Application Builder 솔루션은 다음과 같은 기능을 제공합니다.

빠른 실험

이 솔루션을 사용하면 구성이 다른 여러 인스턴스를 배포하고 출력과 성능을 비교하는 데 필요한 과도한 부담을 제거하여 빠르게 실험할 수 있습니다. 다양한 LLMs, 프롬프트 엔지니어링, 엔터프라이즈 지식 기반, 가드레일, AI 에이전트 및 기타 파라미터의 여러 구성을 실험합니다.

선택 및 구성 가능성

Amazon Bedrock을 통해 사용할 수 있는 모델과 같은 다양한 LLMs에 대한 사전 구축된 커넥터를 사용하는이 솔루션은 원하는 모델과 선호하는 AWS 및 주요 FM 서비스를 유연하게 배포할 수 있습니다. Amazon Bedrock Agents를 활성화하여 다양한 작업 및 워크플로를 수행할 수도 있습니다.

에이전트 빌더

전체 수명 주기 관리를 통해 프로덕션 지원 AI 에이전트를 구축하고 배포합니다. 시스템 프롬프트를 구성하고, 엔터프라이즈 도구 및 데이터 액세스를 위해 모델 컨텍스트 프로토콜(MCP) 서버를 통합하고, 대화 전반에 걸쳐 컨텍스트 보존을 위한 메모리 기능을 활성화합니다. 에이전트는 확장된 런타임 지원 및 실시간 스트리밍 응답을 통해 Amazon Bedrock AgentCore에 배포됩니다.

워크플로 빌더

계층적 위임을 사용하여 여러 Agent Builder 에이전트를 복잡한 워크플로로 오케스트레이션합니다. 다단계 작업을 처리하기 위해 특수 Agent Builder 에이전트를 자율적으로 선택하고 조정하는 감독자 에이전트를 생성합니다. 기존 Agent Builder 배포를 재사용하면서 에이전트 설명, 위임 전략 및 워크플로 수준 메모리를 구성합니다.

프로덕션 지원

AWS Well-Architected 설계 원칙을 기반으로 구축된이 솔루션은 고가용성과 짧은 지연 시간으로 엔터프라이즈급 보안 및 확장성을 제공하여 고성능 표준으로 애플리케이션에 원활하게 통합할 수 있습니다.

확장 가능한 모듈식 아키텍처

기존 프로젝트를 통합하거나 기본적으로 추가 AWS 서비스를 연결하여이 솔루션의 기능을 확장합니다. 이 애플리케이션은 오픈 소스 애플리케이션이므로 포함된 LangChain 오케스트레이션 계층 또는 Lambda 함수를 사용하여 원하는 서비스와 연결할 수 있습니다.

AWS Systems Manager의 기능인 Service Catalog AppRegistry 및 Application Manager와 통합 AWS Systems Manager

이 솔루션에는 솔루션의 CloudFormation 템플릿과 기본 리소스를 AWS Service Catalog AppRegistry 및 AWS Systems Manager Application Manager의 애플리케이션으로 등록하는 Service Catalog AppRegistry 리소스가 포함되어 있습니다. AWS Service Catalog 이 통합을 통해 솔루션의 리소스를 중앙에서 관리할 수 있습니다.

에이전트 빌더와 Bedrock 에이전트 사용 사례

이 솔루션은 AI 에이전트 작업을 위한 두 가지 고유한 접근 방식을 제공하며, 각각 다양한 사용 사례 및 요구 사항에 적합합니다.

기능	Bedrock Agent 사용 사례	에이전트 빌더
용도	사전 배포된 Amazon Bedrock 에이전트 호출	사용자 지정 에이전트 구축, 배포 및 관리
구성	에이전트 ID 및 별칭 ID만	전체 에이전트 구성: 시스템 프롬프트, 모델, MCP 서버, 메모리
배포	단순 호출 계층	AgentCore 런타임의 전체 에이전트 수명 주기
런타임	Amazon Bedrock Agents 서비스	Amazon Bedrock AgentCore와 Strands SDK
도구 통합	Bedrock Agents 콘솔에서 구성됨	MCP(모델 컨텍스트 프로토콜) 서버 및 내장 Strands 도구

기능	Bedrock Agent 사용 사례	에이전트 빌더
메모리	Bedrock Agents에서 관리(최대 30일)	구성 가능한 단기 및 장기 보존이 포함된 AgentCore 메모리
사용자 지정	사전 배포된 에이전트 설정으로 제한됨	프롬프트, 모델, 도구 및 동작을 완벽하게 제어
최적의 용도	기존 에이전트의 빠른 배포	사용자 지정 에이전트 개발 및 프로덕션 배포

> **Note**
>
> 두 옵션 모두 실시간 스트리밍, 대화 기록 및 엔터프라이즈급 보안을 지원합니다.

워크플로 빌더

Workflow Builder는 전문 에이전트 빌더 에이전트에게 작업을 위임하는 감독자 에이전트를 생성하여 다중 에이전트 오케스트레이션을 활성화합니다. 각 워크플로는 다음으로 구성됩니다.

- 감독자 에이전트: 사용자 요청을 수신하고 특수 에이전트를 조정하는 진입점 에이전트
- 전문 에이전트: 감독자가 작업을 위임할 수 있는 Agent Builder 사용 사례
- 에이전트를 도구 패턴으로: 감독자는 각 Agent Builder 에이전트를 도구로 등록하고 사용할 에이전트를 자율적으로 선택합니다.

기능	에이전트 빌더	워크플로 빌더
용도	단일 사용자 지정 에이전트 구축 및 배포	여러 Agent Builder 에이전트 오케스트레이션
에이전트 유형	MCP 도구를 사용하는 단일 에이전트	감독자 에이전트 + 여러 Agent Builder 에이전트
도구 통합	MCP 서버 및 Strands 도구	도구로 등록된 Agent Builder 에이전트

기능	에이전트 빌더	워크플로 빌더
위임	직접 도구 호출	자율 에이전트 선택 및 위임
복잡성	단일 에이전트 작업	다단계, 다중 에이전트 워크플로
에이전트 재사용	해당 사항 없음	기존 Agent Builder 배포를 재사용합니다.
최적의 용도	중점 단일 도메인 작업	여러 전문화가 필요한 복잡한 워크플로

Note

- 워크플로에는 특수 에이전트로서 최소 1개의 에이전트 빌더 사용 사례가 필요합니다.
- 모든 특수 에이전트는 GAAB에 배포된 에이전트 빌더 사용 사례여야 합니다.

사용 사례

엔터프라이즈 데이터에 대한 질문 답변

LLMs 및 기타 파운데이션 모델은 대규모 데이터 코퍼스에 대해 사전 훈련되어 많은 자연어 처리(NLP) 작업에서 잘 수행할 수 있습니다. 그러나 대부분의 파운데이션 모델 및 LLMs은 정적이며 사전 훈련되어 신규, 전문 또는 독점 주제에 대한 질문에 정확하게 답변하는 능력을 제한합니다. 프롬프트 기반 학습을 사용하면 LLM의 강력한 NLP 및 텍스트 생성 기능을 활용하여 엔터프라이즈 데이터에 대해 더 풍부한 고객 경험을 제공할 수 있습니다.

빠른 생성형 AI 프로토타이핑

기본적으로 솔루션은 다양한 모델 공급자 및 사용 사례와 함께 제공됩니다. 사용하기 쉬운 배포 마법사를 통해 고객은 사전 구축된 사용 사례를 배포하여 다양한 생성형 AI 프로토타입 및 워크로드를 신속하게 실험할 수 있습니다.

다중 LLM 비교 및 실험

LLMs 성능이 다르며 애플리케이션의 특정 요구 사항에 따라 한 LLM이 다른 LLM보다 애플리케이션에 더 적합할 수 있습니다. 성능, 정확성, 비용, 창의성 또는 기타 여러 요인과 관련된 이유 때문일 수 있습

니다. 이 솔루션을 사용하면 여러 사용 사례를 빠르게 배포하여 필요에 맞는 구성을 찾을 때까지 다양한 구성을 실험하고 비교할 수 있습니다.

개념 및 정의

이 섹션에서는 이 솔루션과 관련된 핵심 개념 및 용어에 대해 설명합니다.

관리자 사용자

이 가이드의 컨텍스트 내에서 관리자 사용자는 배포에 포함된 콘텐츠를 관리할 책임이 있습니다. 이 사용자는 배포 대시보드 UI에 액세스할 수 있으며 주로 비즈니스 사용자 경험을 큐레이션하는 역할을 합니다. 이는 기본 대상 고객입니다.

비즈니스 사용자

이 가이드의 컨텍스트 내에서 비즈니스 사용자는 사용 사례가 배포된 개인을 나타냅니다. 이들은 지식 기반의 소비자이며 LLMs.

배포 대시보드

배포 대시보드는 관리자 사용자가 사용 사례를 보고 관리하고 생성할 수 있는 관리 콘솔 역할을 하는 웹 인터페이스입니다. 이 대시보드를 통해 고객은 LLMs.

DevOps 사용자

이 가이드의 컨텍스트 내에서 DevOps 사용자는 AWS 계정 내에 솔루션을 배포하고 인프라 관리, 솔루션 업데이트, 성능 모니터링, 솔루션의 전반적인 상태 및 수명 주기 유지를 담당하는 사용자입니다.

사용 사례

사용 사례는 LLMs과 통합되어 신규 또는 기존 애플리케이션에 자연어 인터페이스를 추가하여 더 풍부한 고객 경험을 가능하게 하는 전체 솔루션에서 분리된 애플리케이션입니다. 사용 사례는 배포 대시보드를 통해 또는 자체적으로 배포할 수 있습니다.

> **Note**
>
> AWS 용어에 대한 일반 참조는 AWS 용어집을 참조하세요.

아키텍처 개요

이 섹션에서는이 솔루션과 함께 배포된 구성 요소에 대한 참조 구현 아키텍처 다이어그램을 제공합니다.

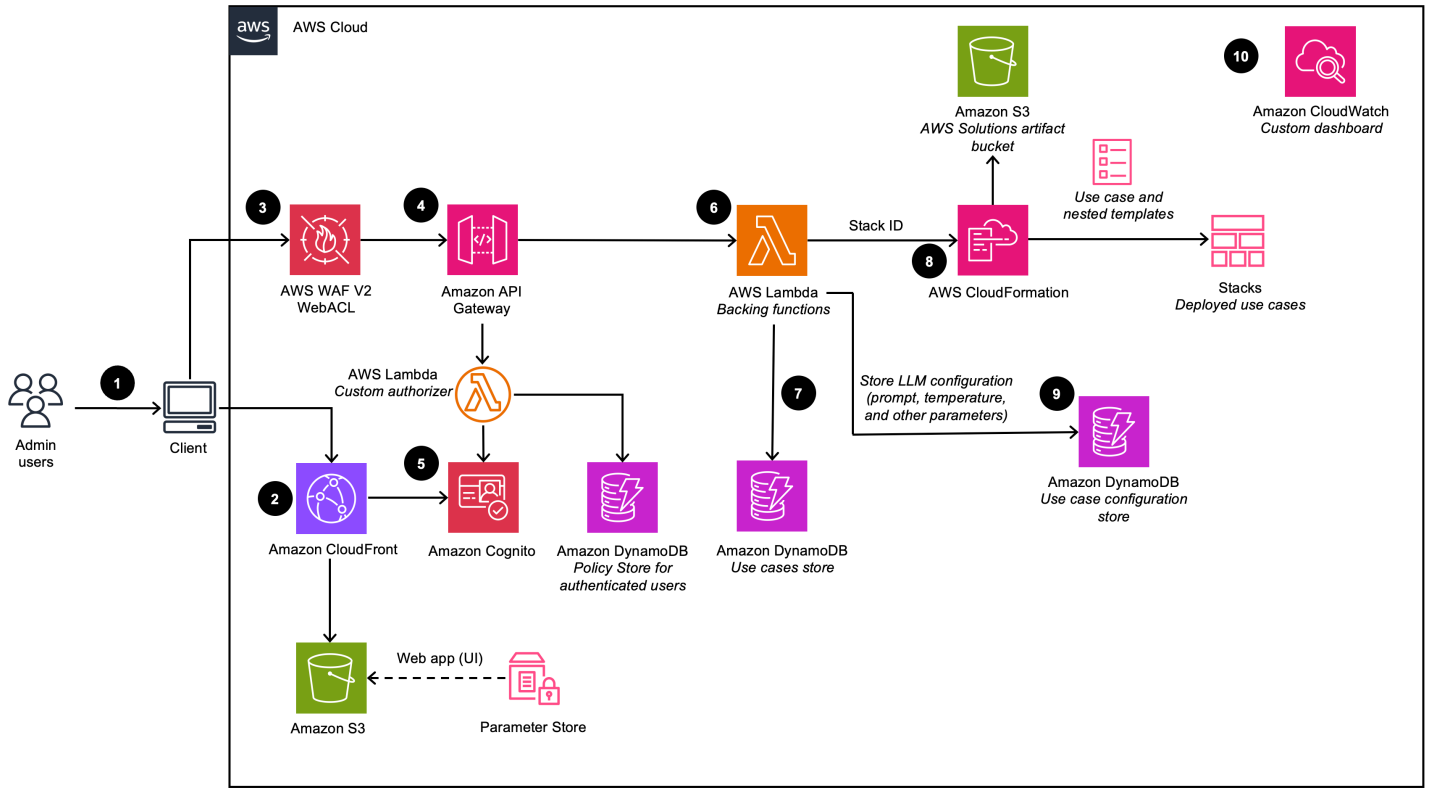
아키텍처 다이어그램

여러 사용 사례 및 비즈니스 요구 사항을 지원하기 위해이 솔루션은 6개의 AWS CloudFormation 템플릿을 제공합니다.

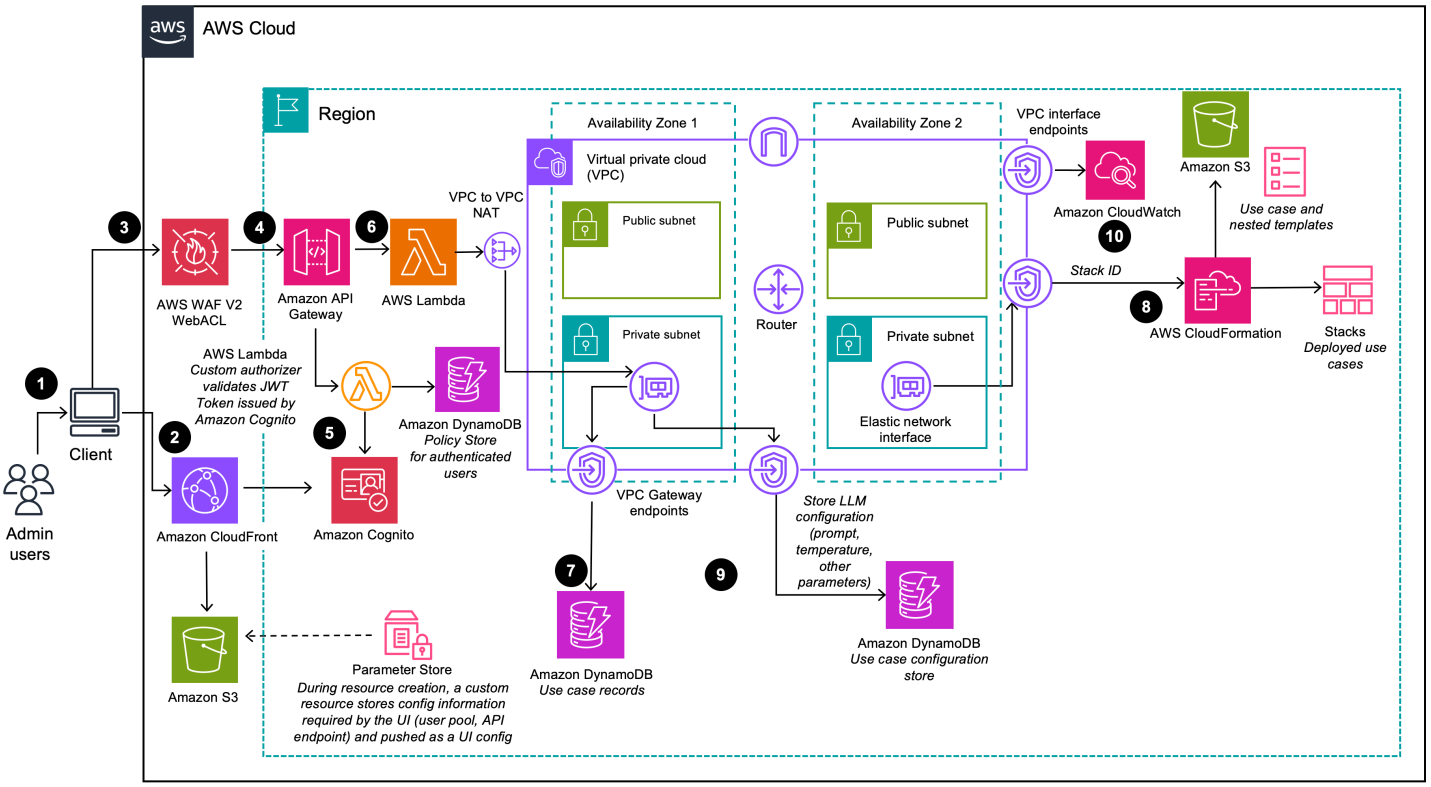
1. 배포 대시보드 - 배포 대시보드는 관리자 사용자가 사용 사례를 보고 관리하고 생성할 수 있는 관리 콘솔 역할을 하는 웹 인터페이스입니다. 이 대시보드를 통해 고객은 LLMs.
2. 텍스트 사용 사례 - 텍스트 사용 사례를 통해 사용자는 생성형 AI를 사용하여 자연어 인터페이스를 경험할 수 있습니다. 이 사용 사례는 신규 또는 기존 애플리케이션에 통합할 수 있으며 배포 대시보드를 통해 배포하거나 제공된 URL을 통해 독립적으로 배포할 수 있습니다.
3. Bedrock 에이전트 사용 사례 - Bedrock 에이전트 사용 사례를 사용하면 기존 Bedrock 에이전트를 사용하여 작업을 완료하거나 반복 워크플로를 자동화할 수 있습니다.
4. MCP 서버 - MCP 서버 사용 사례를 통해 AI 애플리케이션에 대한 표준화된 도구 및 리소스 액세스를 제공하는 모델 컨텍스트 프로토콜 서버를 배포하고 관리할 수 있습니다. 기존 Lambda 함수, APIs와 사용자 지정 컨테이너화된 MCP 서버를 배포하기 위한 런타임 메서드를 모두 지원합니다.
5. 에이전트 빌더 - 에이전트 빌더를 사용하면 전체 구성 제어, MCP 서버 통합 및 메모리 관리 기능을 통해 Amazon Bedrock AgentCore에서 프로덕션 지원 AI 에이전트를 생성하고 배포할 수 있습니다.
6. Workflow Builder - Workflow Builder를 사용하면 복잡한 다중 에이전트 워크플로에서 에이전트를 도구 위임 패턴으로 사용하여 여러 에이전트를 오케스트레이션하는 감독자 에이전트를 생성할 수 있습니다.

배포 대시보드

배포 대시보드 아키텍처를 보여줍니다(VPC 옵션이 비활성화된 상태로 배포된 경우).



배포 대시보드 아키텍처를 보여줍니다(VPC 옵션이 활성화된 상태로 배포된 경우).



> **Note**
>
> AWS CloudFormation 리소스는 AWS Cloud Development Kit(AWS CDK) 구문에서 생성됩니다.

AWS CloudFormation 템플릿과 함께 배포된 솔루션 구성 요소의 상위 수준 프로세스 흐름은 다음과 같습니다.

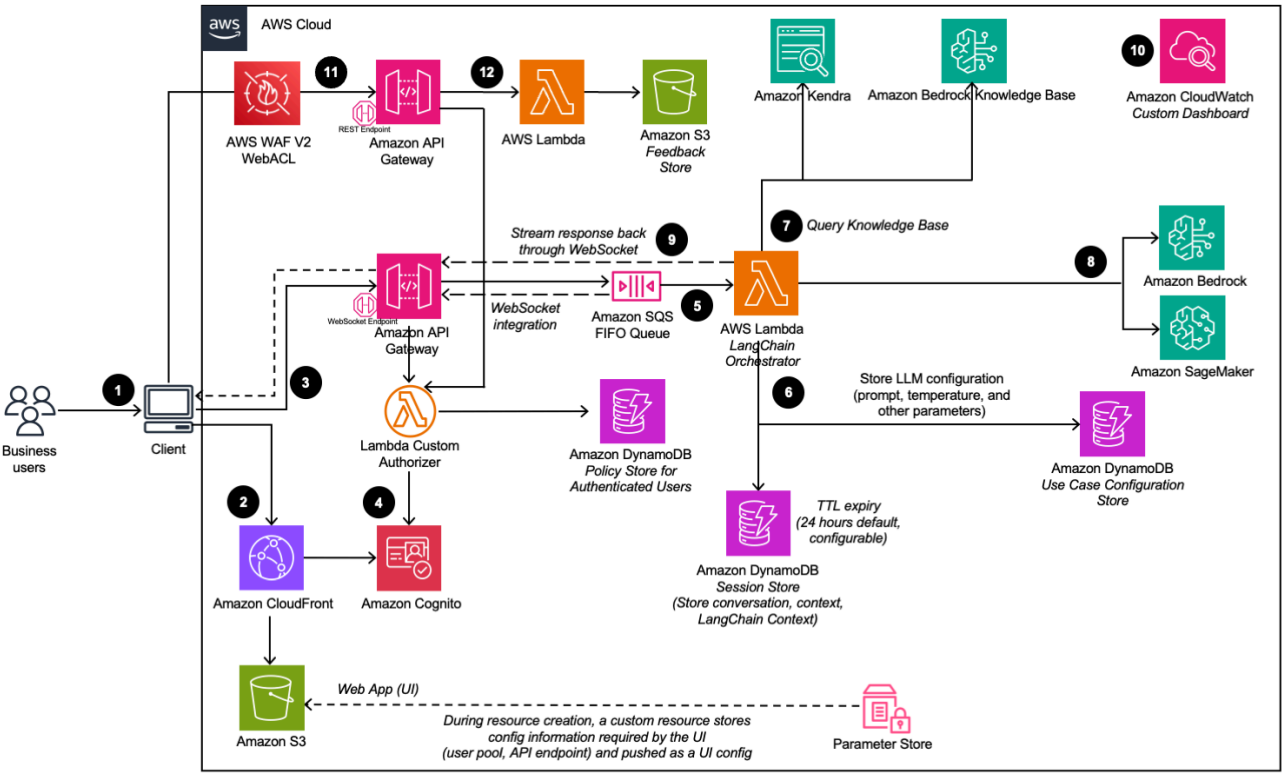
1. 관리자 사용자는 배포 대시보드 사용자 인터페이스(UI)에 로그인합니다.
2. Amazon CloudFront는 Amazon Simple Storage Service(Amazon S3) 버킷에서 호스팅되는 웹 UI를 제공합니다.
3. AWS WAF는 공격으로부터 APIs 보호합니다. 이 솔루션은 구성 가능한 사용자 정의 웹 보안 규칙 및 조건을 기반으로 웹 요청을 허용, 차단 또는 계산하는 웹 액세스 제어 목록(웹 ACL)이라는 규칙 세트를 구성합니다.
4. 웹 UI는 Amazon APIs Gateway를 사용하여 노출되는 REST API 세트를 활용합니다. Amazon API Gateway
5. Amazon Cognito는 사용자를 인증하고 CloudFront 웹 UI와 API Gateway를 모두 지원합니다.
6. AWS Lambda는 REST 엔드포인트에 대한 비즈니스 로직을 제공합니다. 이 백업 Lambda 함수는 AWS CloudFormation을 사용하여 사용 사례 배포를 수행하는 데 필요한 리소스를 관리하고 생성합니다.
7. Amazon DynamoDB는 배포 목록을 저장합니다.
8. 관리자 사용자가 새 사용 사례를 생성하면 백업 Lambda 함수가 요청된 사용 사례에 대한 CloudFormation 스택 생성 이벤트를 시작합니다.
9. 배포 마법사에서 관리자 사용자가 제공하는 모든 LLM 구성 옵션은 DynamoDB에 저장됩니다. 배포는이 DynamoDB 테이블을 사용하여 런타임 시 LLM을 구성합니다.
10. 이 솔루션은 Amazon CloudWatch를 사용하여 다양한 서비스에서 운영 지표를 수집하여 솔루션의 성능 및 운영 상태를 모니터링할 수 있는 사용자 지정 대시보드를 생성합니다.

> **Note**
>
> - Amazon VPC에이 솔루션을 배포하도록 선택하면 프라이빗 네트워크 내에서 데이터가 라우팅됩니다.

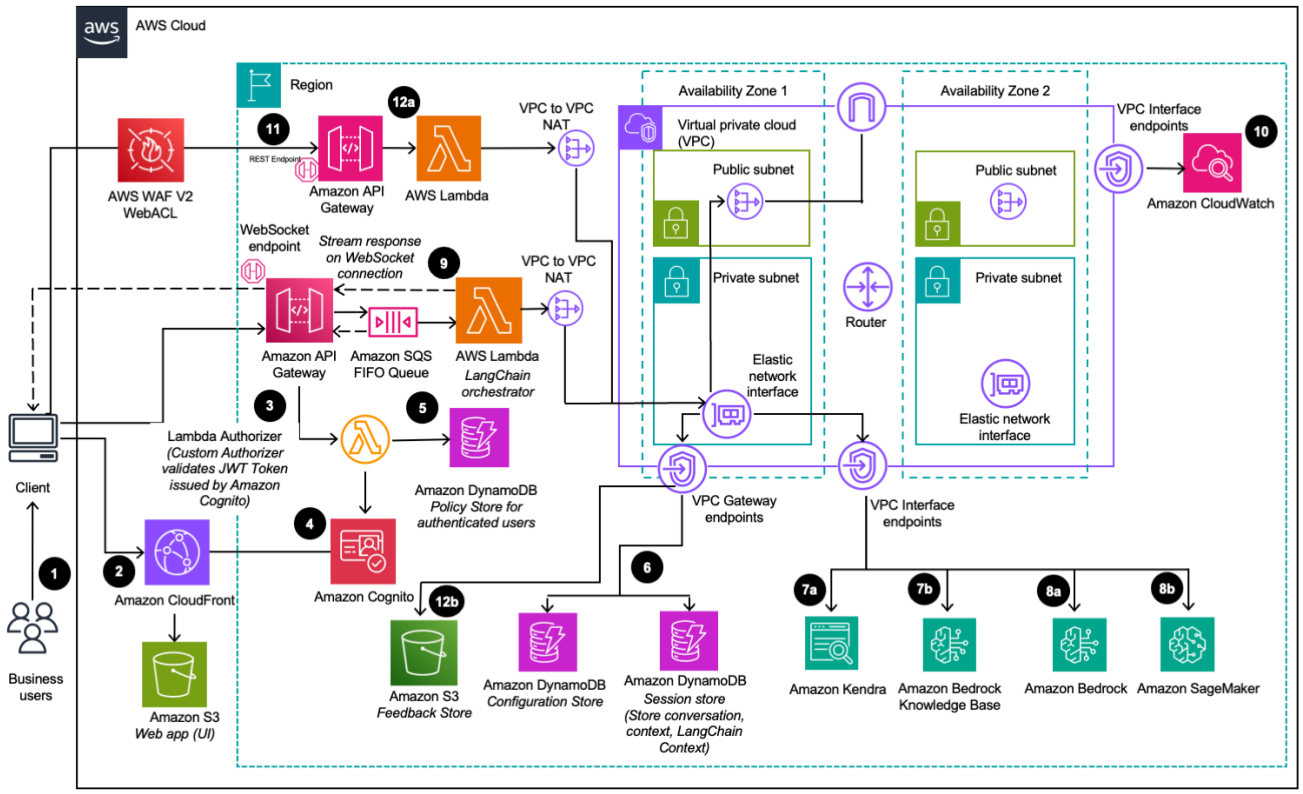
- 배포 대시보드는 대부분의 AWS 리전에서 시작할 수 있지만 배포된 사용 사례에는 서비스 가용성에 따라 특정 제한이 있습니다. 자세한 내용은 지원되는 AWS 리전을 참조하세요.

텍스트 사용 사례

텍스트 사용 사례 아키텍처를 보여줍니다(VPC 옵션이 비활성화된 상태로 배포된 경우).



텍스트 사용 사례 아키텍처를 보여줍니다(VPC 옵션이 활성화된 상태로 배포된 경우).



AWS CloudFormation 템플릿과 함께 배포된 솔루션 구성 요소의 상위 수준 프로세스 흐름은 다음과 같습니다.

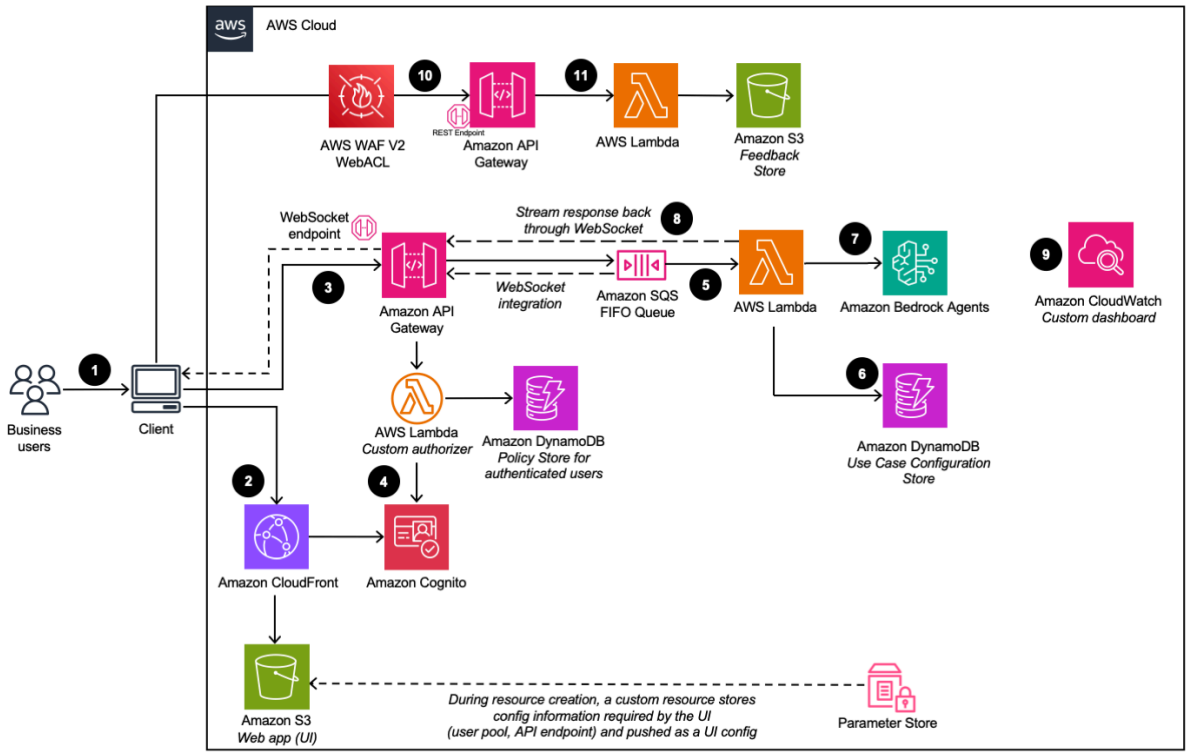
1. 관리자 사용자는 배포 대시보드를 사용하여 사용 사례를 배포합니다. 비즈니스 사용자는 사용 사례 UI에 로그인합니다.
2. CloudFront는 S3 버킷에서 호스팅되는 웹 UI를 제공합니다.
3. 웹 UI는 API Gateway를 사용하여 구축된 WebSocket 통합을 활용합니다. API Gateway는 인증 사용자가 속한 Amazon Cognito 그룹을 기반으로 적절한 AWS Identity and Access Management(IAM) 정책을 반환하는 사용자 지정 Lambda 권한 부여자 함수의 지원을 받습니다. 정책은 DynamoDB에 저장됩니다.
4. Amazon Cognito는 사용자를 인증하고 CloudFront 웹 UI와 API Gateway를 모두 지원합니다.
5. 비즈니스 사용자의 수신 요청은 API Gateway에서 Amazon SQS 대기열로 전달된 다음 LangChain Orchestrator로 전달됩니다. LangChain 오케스트레이터는 비즈니스 사용자의 요청을 이행하기 위한 비즈니스 로직을 제공하는 Lambda 함수 및 계층의 모음입니다. 대기열은 API Gateway와 Lambda 통합의 비동기 작업을 활성화합니다. 대기열은 연결 정보를 Lambda 함수에 전달한 다음 결과를 API Gateway 웹 소켓 연결에 직접 게시하여 장기 실행 추론 호출을 지원합니다.
6. LangChain Orchestrator는 Amazon DynamoDB를 사용하여 구성된 LLM 옵션과 필요한 세션 정보(예: 채팅 기록)를 가져옵니다.

7. 배포에 지식 기반이 활성화된 경우 LangChain Orchestrator는 Amazon Kendra 또는 Amazon Bedrock용 지식 기반을 활용하여 검색 쿼리를 실행하여 문서 발췌문을 검색합니다.
8. LangChain Orchestrator는 지식 기반의 채팅 기록, 쿼리 및 컨텍스트를 사용하여 최종 프롬프트를 생성하고 Amazon Bedrock 또는 Amazon Amazon SageMaker AI에서 호스팅되는 LLM으로 요청을 보냅니다.
9. LLM에서 응답이 반환되면 LangChain Orchestrator는 클라이언트 애플리케이션에서 사용할 API Gateway WebSocket을 통해 응답을 다시 스트리밍합니다.
10. 이 솔루션은 Amazon CloudWatch를 사용하여 다양한 서비스에서 운영 지표를 수집하여 배포의 성능 및 운영 상태를 모니터링할 수 있는 사용자 지정 대시보드를 생성합니다.
11. 피드백 수집이 활성화된 경우 Amazon API Gateway를 활용하는 REST API 엔드포인트를 사용자 피드백 수집에 사용할 수 있습니다.
12. 피드백 지원 Lambda는 추가 사용 사례별 메타데이터(예: 사용된 모델)로 제출된 피드백을 보강하고 이후 DevOps 사용자의 분석 및 보고를 위해 Amazon S3에 데이터를 저장합니다.

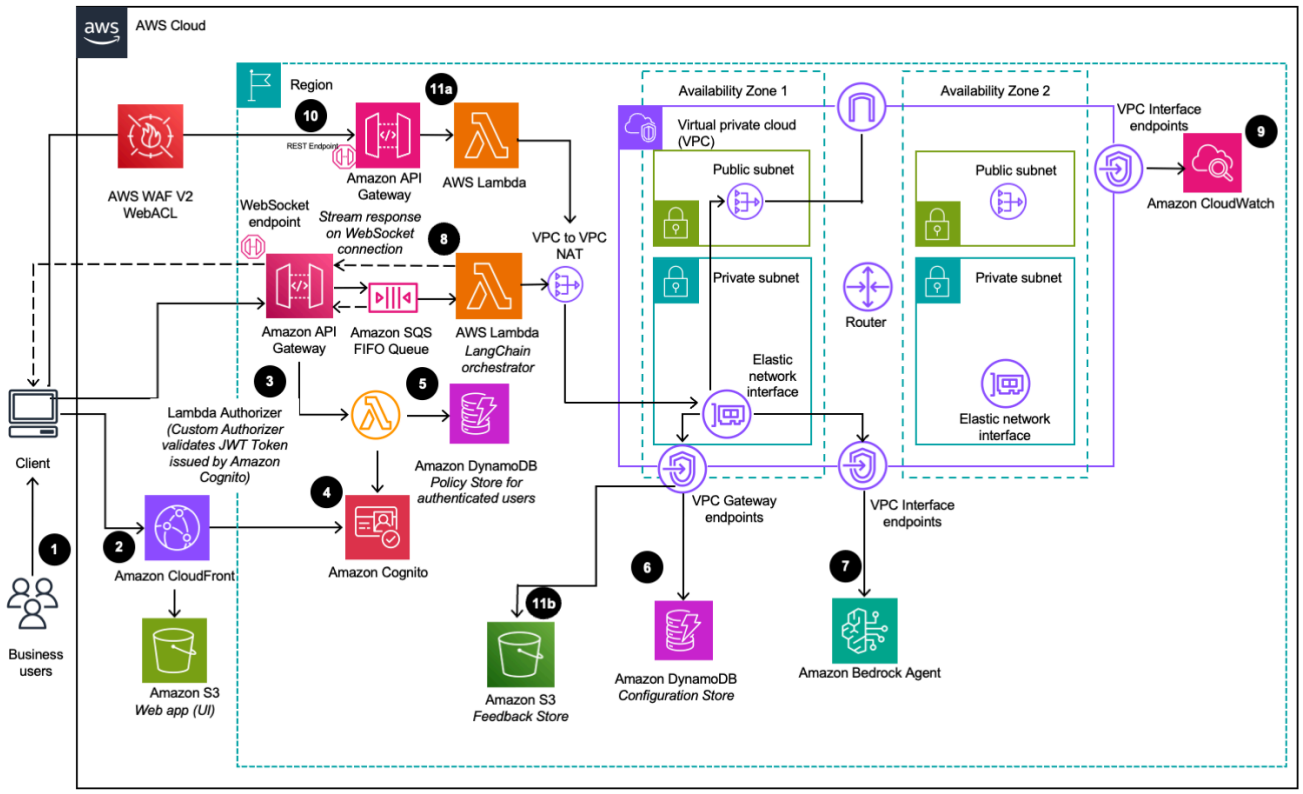
> **Note**
>
> 이 솔루션을 Amazon VPC에 배포하도록 선택하면 데이터가 프라이빗 네트워크로 라우팅됩니다.

Bedrock Agent 사용 사례

Bedrock 에이전트 사용 사례 아키텍처를 보여줍니다(VPC 옵션이 비활성화된 상태로 배포된 경우).



Bedrock 에이전트 사용 사례 아키텍처를 보여줍니다(VPC 옵션이 활성화된 상태로 배포된 경우).



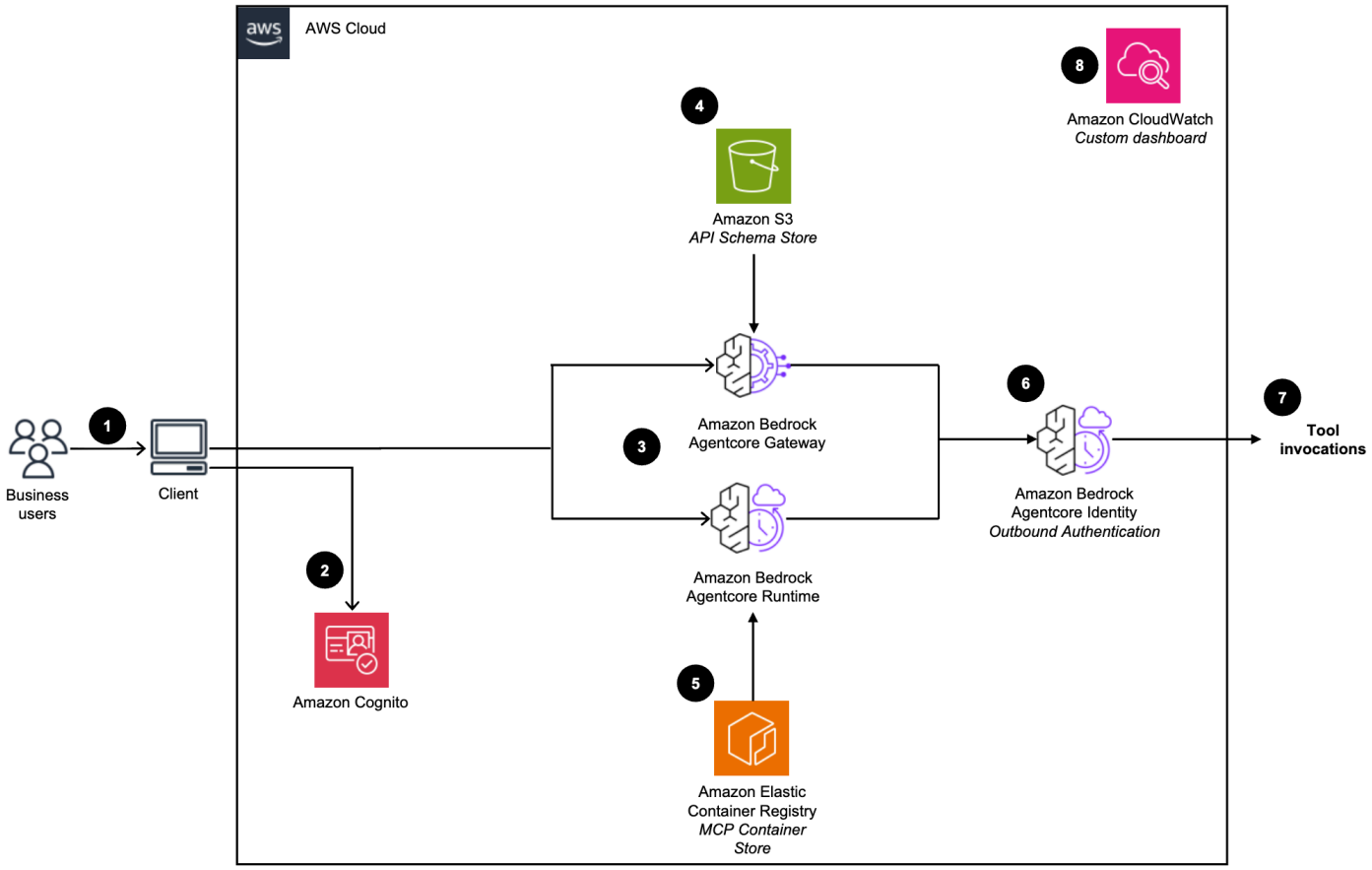
AWS CloudFormation 템플릿과 함께 배포된 솔루션 구성 요소의 상위 수준 프로세스 흐름은 다음과 같습니다.

1. 관리자 사용자는 배포 대시보드를 사용하여 사용 사례를 배포합니다. 비즈니스 사용자는 사용 사례 UI에 로그인합니다.
2. CloudFront는 S3 버킷에서 호스팅되는 웹 UI를 제공합니다.
3. 웹 UI는 API Gateway를 사용하여 구축된 WebSocket 통합을 활용합니다. API Gateway는 인증 사용자가 속한 Amazon Cognito 그룹을 기반으로 적절한 AWS Identity and Access Management(IAM) 정책을 반환하는 사용자 지정 Lambda 권한 부여자 함수의 지원을 받습니다. 정책은 DynamoDB에 저장됩니다.
4. Amazon Cognito는 사용자를 인증하고 CloudFront 웹 UI와 API Gateway를 모두 지원합니다.
5. 비즈니스 사용자의 수신 요청은 API Gateway에서 Amazon SQS 대기열로 전달된 다음 AWS Lambda 함수로 전달됩니다. 대기열은 API Gateway와 Lambda 통합의 비동기 작업을 활성화합니다. 대기열은 연결 정보를 Lambda 함수에 전달한 다음 결과를 API Gateway 웹 소켓 연결에 직접 게시하여 장기 실행 추론 호출을 지원합니다.
6. AWS Lambda 함수는 Amazon DynamoDB를 사용하여 필요에 따라 사용 사례 구성을 가져옵니다.
7. AWS Lambda 함수는 사용자 입력 및 관련 사용 사례 구성을 사용하여 구성된 Amazon Bedrock Agent에 요청 페이로드를 빌드하고 전송하여 사용자 의도를 이행합니다.
8. Amazon Bedrock Agent에서 응답이 반환되면 Lambda 함수는 클라이언트 애플리케이션에서 사용할 API Gateway WebSocket을 통해 응답을 다시 스트리밍합니다.
9. 이 솔루션은 Amazon CloudWatch를 사용하여 다양한 서비스에서 운영 지표를 수집하여 배포의 성능 및 운영 상태를 모니터링할 수 있는 사용자 지정 대시보드를 생성합니다.
10. 피드백 수집이 활성화된 경우 Amazon API Gateway를 활용하는 REST API 엔드포인트를 사용자 피드백 수집에 사용할 수 있습니다.
11. 피드백 지원 Lambda는 추가 사용 사례별 메타데이터로 제출된 피드백을 보강하고 이후 DevOps 사용자의 분석 및 보고를 위해 Amazon S3에 데이터를 저장합니다.

> **Note**
>
> Amazon VPC에이 솔루션을 배포하도록 선택하면 프라이빗 네트워크 내에서 데이터가 라우팅됩니다.

MCP 서버 사용 사례

MCP 서버 사용 사례 아키텍처를 보여줍니다.



MCP 서버 사용 사례를 통해 Amazon Bedrock AgentCore에서 모델 컨텍스트 프로토콜 서버를 배포하고 관리할 수 있습니다. MCP 서버는 AI 애플리케이션이 도구, 리소스 및 엔터프라이즈 데이터 소스에 액세스할 수 있는 표준화된 인터페이스를 제공합니다.

솔루션은 두 가지 배포 방법을 지원합니다.

- 게이트웨이 메서드: 기존 Lambda 함수, REST APIs 또는 외부 MCP 서버를 MCP 도구로 래핑하여 프로토콜 번역을 자동으로 처리합니다.
- 런타임 방법: Amazon ECR 이미지에서 사용자 지정 컨테이너화된 MCP 서버를 배포합니다.

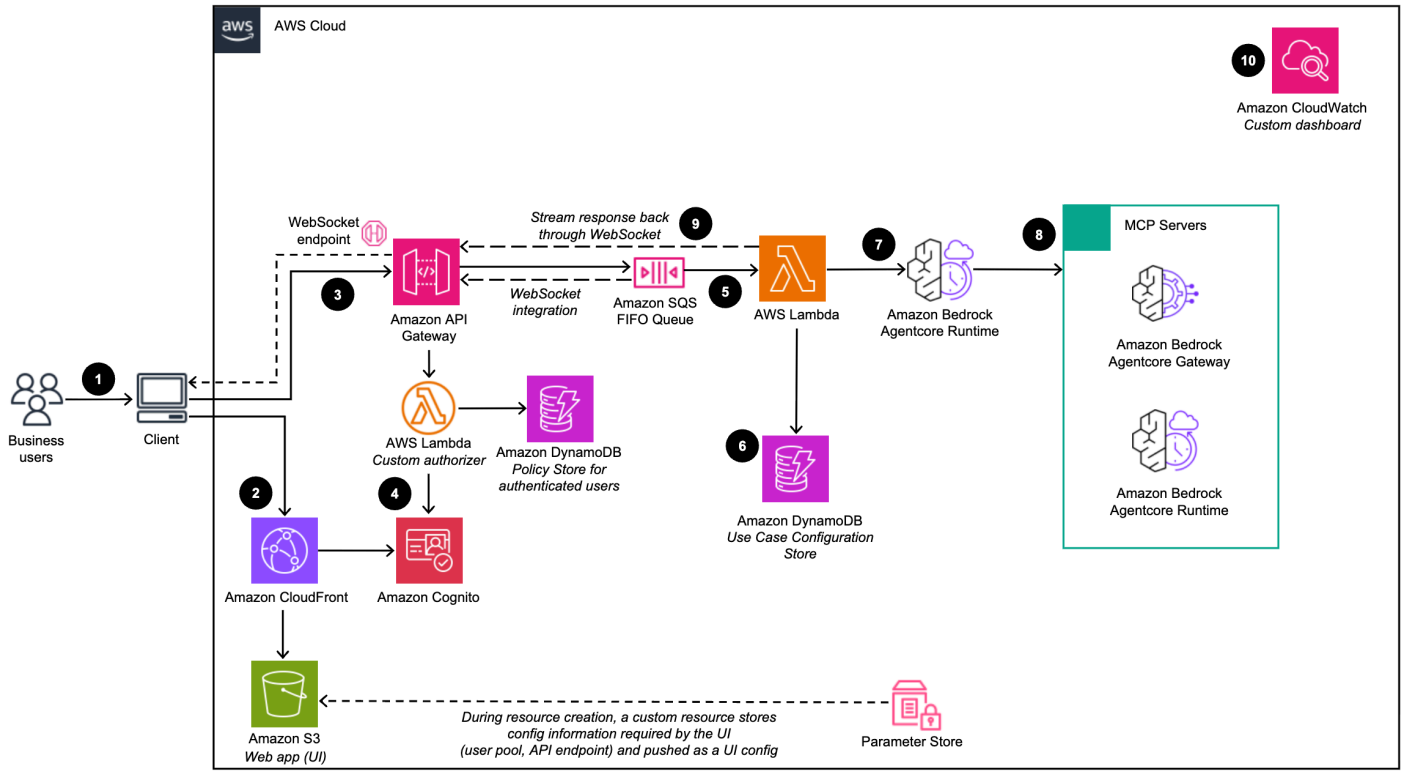
MCP Server 배포를 위한 상위 수준 프로세스 흐름은 다음과 같습니다.

1. 관리자 사용자는 배포 대시보드를 사용하여 게이트웨이 또는 런타임 배포 방법을 선택하여 MCP 서버 사용 사례를 배포합니다.

2. 이 작업은 Amazon Cognito로 인증됩니다.
3. 게이트웨이 배포의 경우 솔루션은 기존 Lambda 함수, APIs 또는 외부 MCP 서버를 MCP 호환 도구로 변환하는 Amazon Bedrock AgentCore Gateway를 생성합니다. 런타임 배포의 경우 솔루션은 제공된 ECR 이미지를 사용하여 Amazon Bedrock AgentCore 런타임에 컨테이너화된 MCP 서버를 배포합니다.
4. 게이트웨이 배포는 Amazon S3에 업로드된 위치에서 필요한 API/Lambda/Smithy 스키마를 검색하거나 MCP 서버 URL 엔드포인트에 직접 연결합니다.
5. 런타임 배포는 Amazon Elastic Container Registry(ECR)에서 사용자가 제공한 컨테이너화된 MCP 서버를 검색합니다.
6. MCP 서버는 Amazon Bedrock AgentCore Identity OAuth 클라이언트로 계측됩니다.
7. MCP 서버를 사용하면 에이전트가 검색할 수 있도록 /mcp 엔드포인트에서 관련 도구를 사용할 수 있습니다.
8. Amazon CloudWatch는 모니터링 및 문제 해결을 위해 MCP 서버 배포에서 운영 지표와 로그를 수집합니다.

Agent Builder 사용 사례

에이전트 빌더 아키텍처를 보여줍니다.



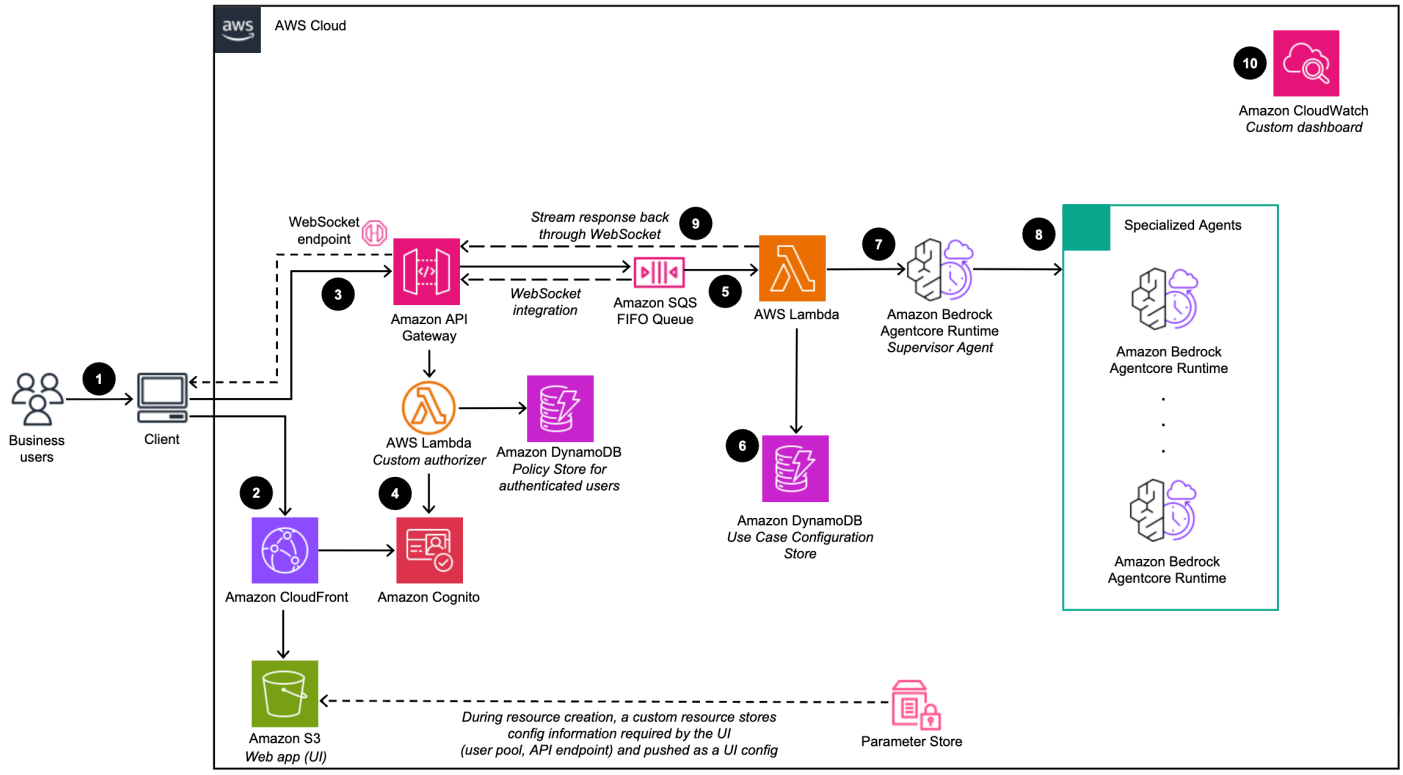
AWS CloudFormation 템플릿과 함께 배포된 Agent Builder 구성 요소의 상위 수준 프로세스 흐름은 다음과 같습니다.

1. 관리자 사용자는 배포 대시보드를 사용하여 사용 사례를 배포합니다. 비즈니스 사용자는 사용 사례 UI에 로그인합니다.
2. CloudFront는 S3 버킷에서 호스팅되는 웹 UI를 제공합니다.
3. 웹 UI는 API Gateway를 사용하여 구축된 WebSocket 통합을 활용합니다. API Gateway는 인증 사용자가 속한 Amazon Cognito 그룹을 기반으로 적절한 AWS Identity and Access Management(IAM) 정책을 반환하는 사용자 지정 Lambda 권한 부여자 함수의 지원을 받습니다. 정책은 DynamoDB에 저장됩니다.
4. Amazon Cognito는 사용자를 인증하고 CloudFront 웹 UI와 API Gateway를 모두 지원합니다.
5. 비즈니스 사용자의 수신 요청은 API Gateway에서 Amazon SQS 대기열로 전달된 다음 AWS Lambda 함수로 전달됩니다. 대기열은 API Gateway와 Lambda 통합의 비동기 작업을 활성화합니다. 대기열은 연결 정보를 Lambda 함수에 전달한 다음 결과를 API Gateway 웹 소켓 연결에 직접 게시하여 장기 실행 추론 호출을 지원합니다.
6. AWS Lambda 함수는 DynamoDB에서 에이전트 구성을 검색합니다.
7. AWS Lambda 함수는 사용자 입력 및 관련 사용 사례 구성을 사용하여 Amazon Bedrock AgentCore 런타임에서 실행되는 요청 페이로드를 빌드하고 에이전트에 전송합니다.
8. 에이전트는 연결된 MCP 서버에 연결하고 도구를 스트랜드 에이전트 인스턴스에 등록합니다. 그런 다음 에이전트는 도구 설명 및 작업 요구 사항에 따라 작업을 자율적으로 선택하고 수행합니다.
9. Amazon Bedrock AgentCore 런타임에서 응답이 반환되면 Lambda 함수는 클라이언트 애플리케이션에서 사용할 API Gateway WebSocket을 통해 응답을 다시 스트리밍합니다.

> **Note**
>
> - 에이전트 처리는 Lambda 실행 제한 시간(15분)으로 제한됩니다.

Workflow Builder 사용 사례

워크플로 빌더 아키텍처를 보여줍니다.



AWS CloudFormation 템플릿과 함께 배포된 Workflow Builder 구성 요소의 상위 수준 프로세스 흐름은 다음과 같습니다.

1. 관리자 사용자는 배포 대시보드를 사용하여 워크플로를 배포하고 특수 에이전트로 포함할 에이전트 빌더 에이전트를 선택합니다.
2. CloudFront는 S3 버킷에서 호스팅되는 웹 UI를 제공합니다.
3. 웹 UI는 API Gateway를 사용하여 구축된 WebSocket 통합을 활용합니다. API Gateway는 인증 사용자가 속한 Amazon Cognito 그룹을 기반으로 적절한 AWS Identity and Access Management(IAM) 정책을 반환하는 사용자 지정 Lambda 권한 부여자 함수의 지원을 받습니다. 정책은 DynamoDB에 저장됩니다.
4. Amazon Cognito는 사용자를 인증하고 CloudFront 웹 UI와 API Gateway를 모두 지원합니다.
5. 비즈니스 사용자의 수신 요청은 API Gateway에서 Amazon SQS 대기열로 전달된 다음 AWS Lambda 함수로 전달됩니다. 대기열은 API Gateway와 Lambda 통합의 비동기 작업을 활성화합니다.
6. AWS Lambda 함수는 특수 에이전트 빌더 에이전트 목록을 포함하여 DynamoDB에서 워크플로 구성을 검색합니다.
7. Lambda는 사용자 입력 및 워크플로 구성을 사용하여 감독자 에이전트를 호스팅하는 Amazon Bedrock AgentCore 런타임에 요청을 보냅니다.

8. 감독자 에이전트는 AgentCore 런타임 환경 내에 모든 특수 에이전트 빌더 에이전트의 로컬 인스턴스를 생성합니다. 이러한 특수 에이전트는 에이전트를 도구 패턴으로 사용하여 도구로 등록됩니다. 그런 다음 감독자는 에이전트 설명 및 작업 요구 사항에 따라 작업을 자율적으로 선택하고 전문 에이전트에게 위임합니다.
9. 감독자 에이전트는 특수 에이전트의 결과를 집계하고 최종 응답을 공식화하여 Lambda로 반환하고 API Gateway Websocket을 통해 클라이언트 애플리케이션으로 다시 스트리밍합니다.

> **Note**
>
> - 워크플로 처리는 Lambda 실행 제한 시간(15분)으로 제한됩니다.

AWS Well-Architected 설계 고려 사항

이 솔루션은 고객이 클라우드에서 안정적이고 안전하며 효율적이고 비용 효율적인 워크로드를 설계하고 운영하는 데 도움이 되는 AWS Well-Architected Framework의 모범 사례로 설계되었습니다.

이 섹션에서는 이 솔루션을 구축할 때 Well-Architected Framework의 설계 원칙과 모범 사례가 어떻게 적용되었는지 설명합니다.

운영 우수성

이 섹션에서는 운영 우수성 요소의 원칙과 모범 사례를 사용하여 이 솔루션을 설계한 방법을 설명합니다.

- Amazon CloudFormation을 사용하여 솔루션을 infrastructure-as-code로 구축했습니다.
- Lambda 함수는 사용자 지정 지표를 CloudWatch 및 사용자 지정 CloudWatch 대시보드로 푸시하여 솔루션 상태를 모니터링합니다.
- 솔루션 구성 요소는 고도로 모듈화되어 있어 배포할 구성 요소를 유연하게 선택할 수 있습니다.

보안

이 섹션에서는 보안 요소의 원칙과 모범 사례를 사용하여 이 솔루션을 설계한 방법을 설명합니다.

- 배포 대시보드 및 모든 사용 사례는 Amazon Cognito에서 인증되고 승인됩니다.
- 모든 서비스 간 통신은 AWS IAM 역할을 사용합니다.

- 모든 솔루션 역할은 최소 권한 액세스를 따릅니다. 즉, 필요한 최소 권한만 부여됩니다.
- S3 버킷, DynamoDB 및 Amazon Kendra를 포함한 모든 데이터 스토리지에는 유휴 암호화가 있습니다.

신뢰성

이 섹션에서는 신뢰성 요소의 원칙과 모범 사례를 사용하여 이 솔루션을 설계한 방법을 설명합니다.

- 서버리스 패러다임을 기반으로 하는 아키텍처입니다.
- 기본 인프라의 장애로부터 온디맨드, 수평 확장성 및 자동 복구를 위한 아키텍처를 구축했습니다.
- 아키텍처에는 기본 엔드포인트를 압도하지 않도록 하는 버퍼링 및 제한 요청이 포함되어 있습니다.

성능 효율성

이 섹션에서는 성능 효율성 요소의 원칙과 모범 사례를 사용하여 이 솔루션을 설계한 방법을 설명합니다.

- 이 솔루션은 온디맨드 규모 조정이 가능한 완전 관리형 서버리스 NoSQL 데이터베이스인 DynamoDB를 사용합니다.
- 이 솔루션은 객체 스토리지에 Amazon S3를 사용하고 웹 사이트(CloudFront를 통해)를 호스팅하여 11 9s의 내구성과 함께 저렴하고 확장 가능한 웹 사이트를 제공합니다.

비용 최적화

이 섹션에서는 비용 최적화 요소의 원칙과 모범 사례를 사용하여 이 솔루션을 설계한 방법을 설명합니다.

- 가능한 경우 서버리스 아키텍처를 사용하는 솔루션을 구축했으므로 사용한 만큼만 비용을 지불하면 됩니다.

지속 가능성

이 섹션에서는 지속 가능성 요소의 원칙과 모범 사례를 사용하여 이 솔루션을 설계한 방법을 설명합니다.

- 이 솔루션의 모듈식 구성 아키텍처는 개별 사용 사례에 맞게 프로비저닝할 리소스를 유연하게 사용자 지정할 수 있습니다.
- 아키텍처는 리소스 사용률을 최적화하는 서버리스 컴퓨팅 및 스토리지를 사용합니다.
- 클라우드 기반 솔루션인이 솔루션은 공유 리소스, 네트워킹, 전력 냉각 및 물리적 시설의 이점을 활용합니다.

아키텍처 세부 정보

이 섹션에서는이 솔루션을 구성하는 구성 요소 및 AWS 서비스와 이러한 구성 요소가 함께 작동하는 방식에 대한 아키텍처 세부 정보를 설명합니다.

이 솔루션의 AWS 서비스

AWS 서비스	설명
Amazon API Gateway	Core. 이 서비스는 배포 대시보드용 REST APIs와 사용 사례에 대한 WebSocket API를 제공합니다.
AWS CloudFormation	Core. 이 솔루션은 CloudFormation 템플릿으로 배포되며 CloudFormation은 솔루션의 AWS 리소스를 배포합니다.
Amazon CloudFront	Core. CloudFront는 Amazon S3에서 호스팅되는 웹 콘텐츠를 제공합니다.
Amazon Cognito	Core. 이 서비스는 API에 대한 사용자 관리 및 인증을 처리합니다.
Amazon DynamoDB	Core. DynamoDB는 배포 대시보드에 대한 배포 정보 및 구성 세부 정보를 저장합니다. 채팅 기록 및 대화 IDs 텍스트 사용 사례에 저장하여 대화 기록 및 쿼리 모호화를 활성화합니다.
Lambda	Core. 솔루션은 Lambda 함수를 사용하여 다음을 수행합니다. * REST 및 WebSocket API 엔드포인트 지원 * 각 사용 사례 오케스트레이터의 코어 로직 처리 * CloudFormation 배포 중에 사용자 지정 리소스 구현
Amazon S3	Core. Amazon S3는 정적 웹 콘텐츠를 호스팅합니다.

AWS 서비스	설명
Amazon CloudWatch	지원. 이 솔루션은 솔루션 리소스의 로그를 CloudWatch Logs에 게시하고 지표를 CloudWatch 지표에 게시합니다. 또한 솔루션은 이 데이터를 볼 수 있는 CloudWatch 대시보드를 생성합니다.
AWS Systems Manager	지원. Systems Manager는 리소스 운영 및 비용 데이터에 대한 애플리케이션 수준의 리소스 모니터링 및 시각화를 제공합니다. Parameter Store에 구성 데이터를 저장하는 데도 사용됩니다.
AWS WAF	지원. AWS WAF는 API Gateway 배포 앞에 배포되어 이를 보호합니다.
Amazon Bedrock	선택 사항. 이 솔루션은 Amazon Bedrock을 활용하여 파운데이션 또는 사용자 지정 모델, Amazon Bedrock 에이전트, Amazon Bedrock 지식 기반에 액세스합니다. Amazon Bedrock은 데이터가 AWS 네트워크를 벗어나지 않도록 하기 위해 권장되는 통합입니다.
Amazon Bedrock AgentCore	선택 사항 솔루션은 Amazon Bedrock AgentCore를 활용하여 MCP 서버 연결과 Agent Builder 및 워크플로 사용 사례를 실행하고 지원합니다.
Amazon Elastic Container Registry(Amazon ECR)	선택 사항. Agent Builder 배포의 경우 ECR은 에이전트 컨테이너 이미지를 저장하고 배포합니다. 이 솔루션은 ECR 풀스루 캐시를 사용하여 GAAB 팀의 퍼블릭 ECR 리포지토리에서 사전 구축된 에이전트 이미지를 자동으로 검색합니다.

AWS 서비스	설명
OpenTelemetry용 AWS Distro(ADOT)	선택 사항. 에이전트 빌더 배포의 경우 ADOT는 에이전트 관찰성을 위한 자동 계측을 제공하여 에이전트 작업에 대한 분산 추적 및 구조화된 로깅을 활성화합니다.
Amazon Kendra	선택 사항. 텍스트 사용 사례에서 관리자는 선택적으로 Amazon Kendra 인덱스를 연결하여 LLM과의 대화에 대한 지식 기반으로 사용하기로 결정할 수 있습니다. 이를 사용하여 LLM에 새 정보를 주입하여 응답에 해당 정보를 사용할 수 있습니다.
Amazon SageMaker AI	선택 사항. 이 솔루션은 Amazon SageMaker AI 추론 엔드포인트와 통합하여 AWS 계정 및 리전 내에서 호스팅되는 FMs에 액세스할 수 있으며, 데이터가 AWS 네트워크를 벗어나지 않도록 하는 기본 통합입니다. **Note** 추론 엔드포인트를 사용할 수 있는 리전과 동일한 리전에 솔루션을 배포해야 합니다.
Amazon Virtual Private Cloud	선택 사항. 솔루션은 VPC 지원 구성으로 구성 요소를 배포하는 옵션을 제공합니다. VPC 지원 구성으로 솔루션을 배포하는 동안 솔루션이 VPC를 생성하도록 하거나 솔루션이 배포될 동일한 계정 및 리전에 있는 기존 VPC를 사용할 수 있습니다(자체 VPC 가져오기). 솔루션이 VPC를 생성하는 경우 서브넷, 보안 그룹 및 해당 규칙, 라우팅 테이블, 네트워크 ACLs, NAT 게이트웨이, 인터넷 게이트웨이, VPC 엔드포인트 및 해당 정책을 포함하는 필요한 네트워크 구성 요소를 생성합니다.

배포 대시보드

API Gateway 사용자 지정 권한 부여자

표면 아래에서 API Gateway에 대한 Lambda 사용자 지정 권한 부여자는 모든 API 호출(RESTful 및 WebSocket 기반 모두)에 사용되어 지정된 사용자에게 자신이 속한 그룹(들)을 기반으로 작업을 수행할 권한이 있는지 확인합니다. 이 사용자 지정 권한 부여자는 각 그룹의 정책이 포함된 DynamoDB 테이블에 의해 지원됩니다. API 호출 시 API Gateway는 사용자 지정 권한 부여자 Lambda 함수를 호출합니다.이 함수는 제공된 Amazon Cognito 액세스 토큰을 디코딩하여 사용자가 속한 사용자 그룹을 결정합니다. 그런 다음 그룹 이름으로 정책 테이블을 쿼리하여 해당 그룹에 대한 관련 정책을 반환합니다.

새로운 사용 사례 배포마다 관리자 정책이 업데이트되어 해당 사용 사례의 API에서 execute-api:Invoke 작업을 허용하는 새 문이 저장됩니다. 사용 사례가 삭제되면 해당 문이 정책에서 제거됩니다.

개별 사용 사례에 대해 생성된 그룹의 경우 정책에 단일 문만 있으므로 해당 사용 사례의 API에서만 execute-api:Invoke 작업을 허용합니다.

이 구조로 인해 사용 사례의 그룹에 속한 모든 사용자는 해당 사용 사례의 API에 액세스할 수 있습니다. 단일 사용자를 여러 그룹에 수동으로 추가하여 해당 사용자가 여러 사용 사례를 사용할 수 있도록 할 수도 있습니다.

> **Warning**
>
> 기존 사용자 그룹에 새 사용 사례에 대한 액세스 권한을 부여하려면 정책 테이블에서 지정된 그룹의 정책을 수동으로 편집할 수도 있습니다. 사용 사례가 삭제되면 사용 사례 그룹이 삭제되므로(수동으로 편집한 경우에도) 사용 사례를 삭제할 때는 주의해야 합니다.

사용 사례 스택이 독립 실행형으로 배포되는 경우(배포 대시보드 사용 안 함) 해당 사용 사례의 API에 액세스할 수 있는 단일 사용자를 포함하는 해당 배포에 대해 Amazon Cognito 사용자 풀이 생성됩니다. 이 사용자 풀은이 사용 사례에만 속하며 다른 독립 실행형 배포에서는 공유되지 않습니다.

텍스트 사용 사례

스트리밍 지원

채팅 애플리케이션에서 지연 시간은 응답형 사용자 경험을 활성화하는 데 중요한 지표입니다. LLM 추론에 몇 초에서 몇 분 정도 걸릴 가능성은 고객에게 콘텐츠를 가장 잘 제공하는 방법에 어려움을 줍니

다. 이러한 이유로 여러 LLM 공급자가 응답을 호출자에게 다시 스트리밍할 수 있습니다. 응답을 반환하기 전에 전체 추론이 완료될 때까지 기다리는 대신 사용 가능한 경우 각 토큰을 반환할 수 있습니다.

이 기능의 사용을 지원하기 위해 텍스트 사용 사례는 WebSocket API를 사용하여 채팅 경험을 지원하도록 설계되었습니다. 이 WebSocket은 API Gateway를 통해 배포됩니다. WebSocket API를 사용하면 채팅 세션 시작 시 연결을 생성하고 해당 소켓을 통해 응답을 스트리밍할 수 있습니다. 이를 통해 프런트엔드 애플리케이션은 더 나은 사용자 경험을 제공할 수 있습니다.

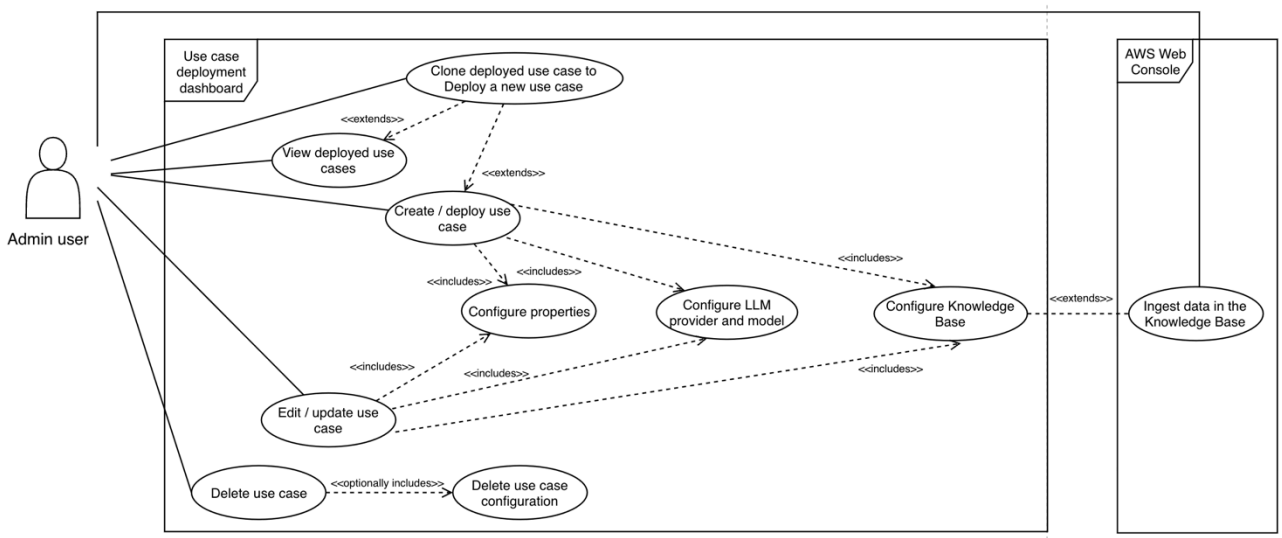
> **Note**
>
> 모델이 스트리밍 지원을 제공하더라도 솔루션이 WebSocket API를 통해 응답을 다시 스트리밍할 수 있다는 의미는 아닙니다. 솔루션이 각 모델 공급자에 대한 스트리밍을 지원하는 사용자 지정 로직을 활성화해야 합니다. 스트리밍을 사용할 수 있는 경우 관리자는 배포 시이 기능을 활성화/비활성화할 수 있습니다.

AWS의 생성형 AI Application Builder 솔루션 작동 방식

관리자 사용자는 주로 배포 대시보드와 인터페이스하여 신규 및 기존 사용 사례 배포를 보고, 생성하고, 관리합니다. 이 대시보드를 통해 관리자 사용자는 다음 작업에 액세스할 수 있습니다.

- 배포 목록 보기
- 새 배포 생성
- 기존 배포 편집
- 배포 구성을 복제하여 새 배포 생성
- 배포 삭제(CloudFormation 삭제를 통해 리소스 프로비저닝 해제)
- 배포의 구성 세부 정보 영구 삭제

배포 대시보드의 관리자 사용자를 위한 사용 사례 다이어그램을 보여줍니다.

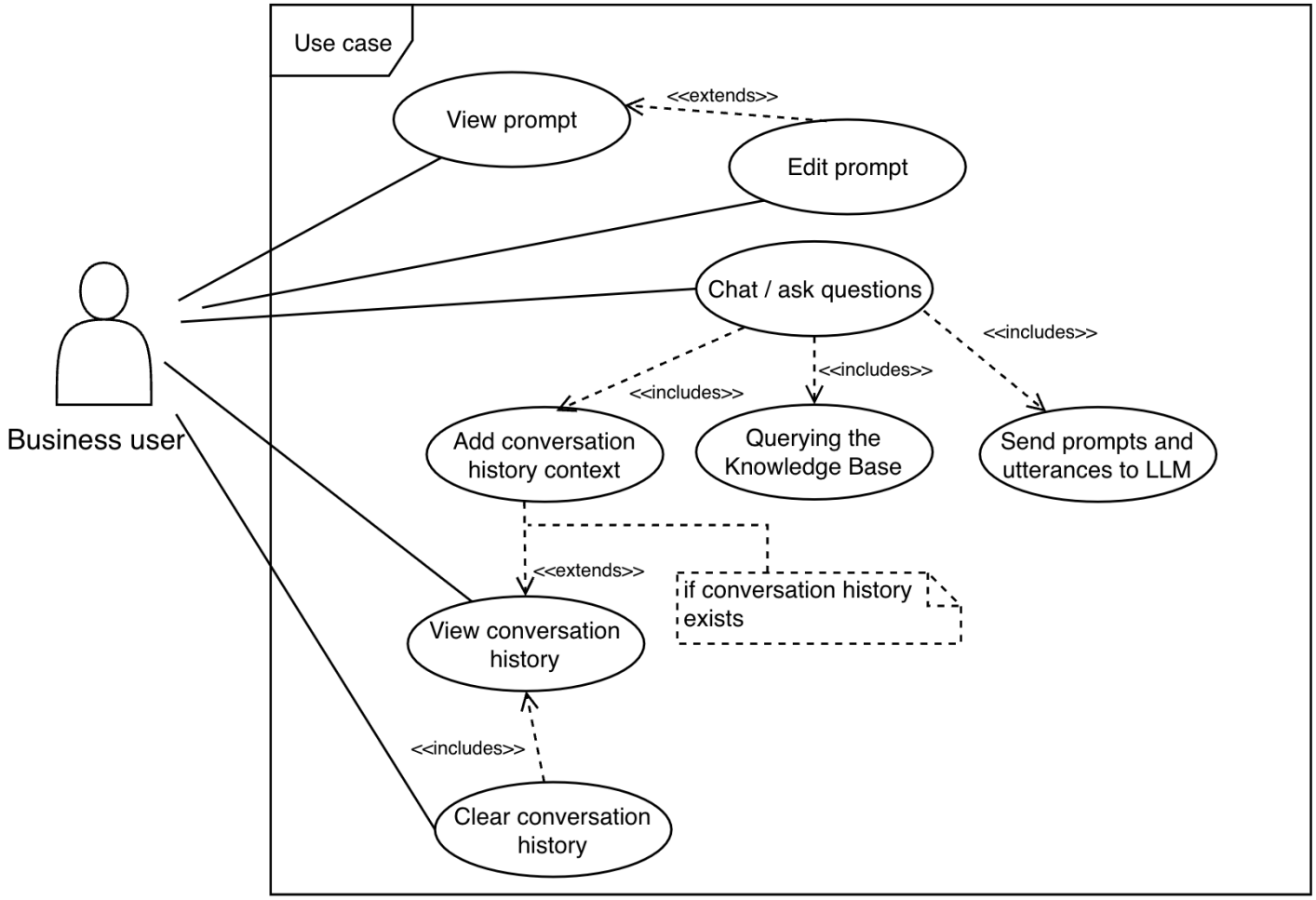


> **Note**
>
> 관리자 사용자는 AWS 콘솔에 직접 액세스할 수 없습니다. 이 경우 관리자는 DevOps 사용자와 협력하여 Kendra 지식 기반에 데이터를 수집하는 등의 작업을 지원해야 합니다.

텍스트 사용 사례의 경우 비즈니스 사용자는 사용자 인터페이스에 액세스하여 LLM과 채팅할 수 있습니다. 이 구성의 세부 정보는 관리자 사용자가 구성한 배포 설정에 의해 제어됩니다. 텍스트 사용 사례에서 비즈니스 사용자는 다음 작업에 액세스할 수 있습니다.

- 채팅 인터페이스를 통해 메시지 전송
- 대화 기록 보기
- 대화 기록 지우기
- 프롬프트 보기
- 프롬프트 편집

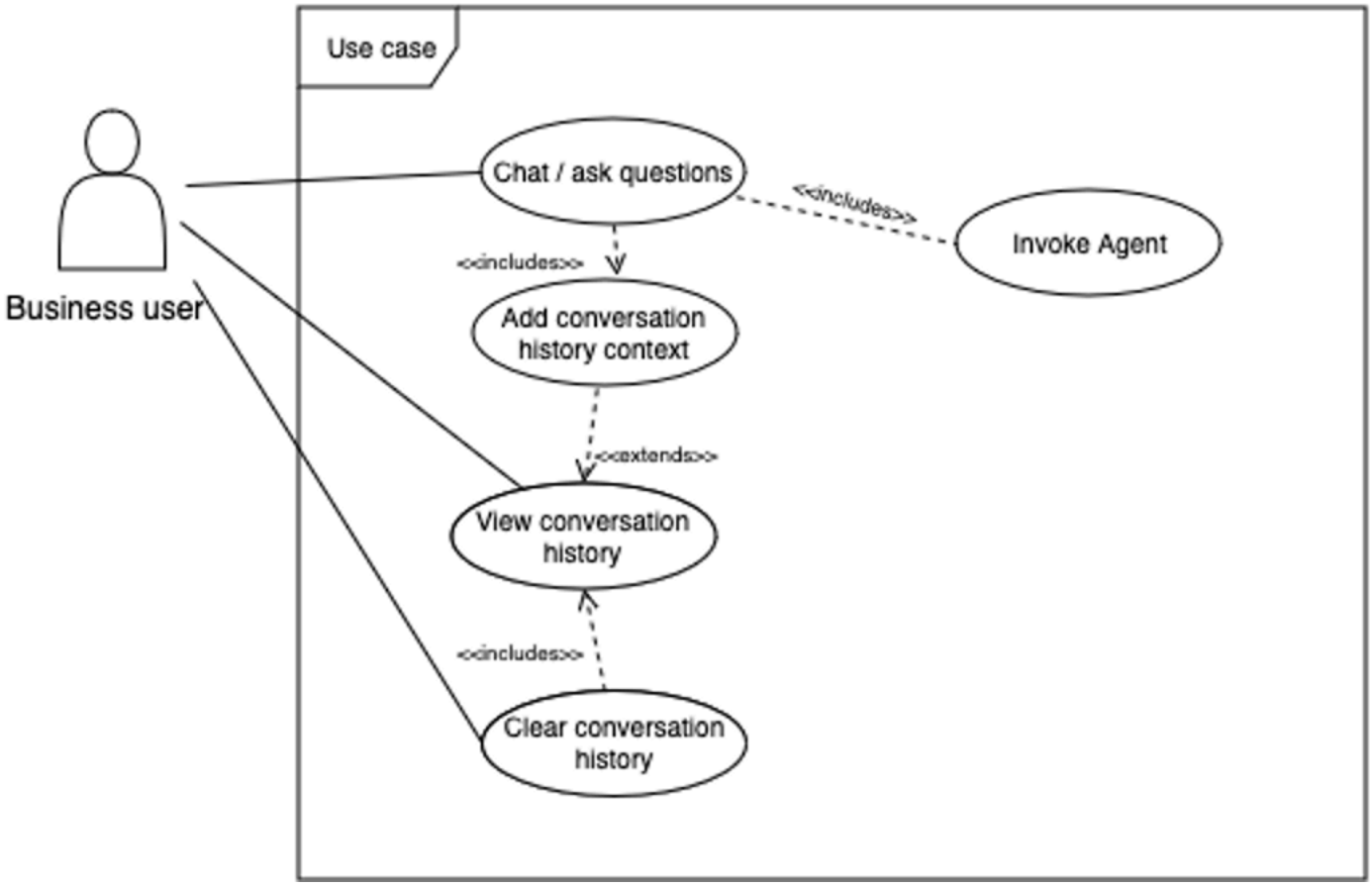
텍스트 사용 사례의 비즈니스 사용자에 대한 사용 사례 다이어그램을 보여줍니다.



Bedrock Agent 사용 사례에서는 비즈니스 사용자가 구성된 Amazon Bedrock Agent와 채팅하기 위한 UI에 액세스할 수 있습니다. 관리자 사용자는 배포 설정에서 이러한 세부 정보를 구성할 수 있습니다. Bedrock Agent 사용 사례에서 비즈니스 사용자는 다음 작업에 액세스할 수 있습니다.

- 채팅 인터페이스를 통해 메시지 전송
- 대화 기록 보기
- 대화 기록 지우기

Bedrock Agent 사용 사례의 비즈니스 사용자를 위한 사용 사례 다이어그램을 보여줍니다.



에이전트 빌더

Agent Builder는 Amazon Bedrock AgentCore에서 프로덕션 지원 AI 에이전트를 생성, 배포 및 관리하기 위한 플랫폼을 제공합니다. 이 섹션에서는 기술 구성 요소와 구현 세부 정보를 설명합니다.

AgentCore 통합

Agent Builder는 사전 구축된 에이전트 이미지와 함께 구성 기반 배포 접근 방식을 사용하여 빠르고 안전하며 확장 가능한 에이전트 배포를 지원합니다.

사전 구축된 에이전트 이미지

에이전트 컨테이너 이미지는 CI/CD 파이프라인 중에 GAAB 팀이 빌드하고 퍼블릭 ECR 리포지토리에 게시합니다. 각 이미지 버전은 GAAB 솔루션 버전(예: v4.0.0 → gaab-strands-agent:v4.0.0)과 연결됩니다. 이미지는 Strands SDK를 기반으로 하며 다음을 포함합니다.

- 에이전트 런타임 환경
- MCP 클라이언트 통합
- 메모리 관리 기능
- OpenTelemetry 계측

ECR 풀스루 캐시

이 솔루션은 ECR 풀스루 캐시를 사용하여 퍼블릭 ECR 리포지토리의 에이전트 이미지를 고객의 프라이빗 ECR에 자동으로 배포합니다. 이 AWS 관리형 서비스는 다음과 같습니다.

- 첫 번째 풀 시 이미지 캐싱(2~5분 지연)
- 사용자 지정 이미지 복사 로직 제거
- 후속 배포를 위한 로컬 이미지 가용성 제공
- 충돌을 방지하기 위해 배포당 고유한 캐시 규칙을 생성합니다.

구성 스토리지

에이전트 구성은 기존 사용 사례 구성과 함께 DynamoDB에 저장됩니다. 각 구성에는 다음이 포함됩니다.

- 시스템 프롬프트 템플릿
- 모델 공급자 및 모델 ID
- 모델 파라미터(온도, max_tokens)
- MCP 서버 참조 및 엔드포인트
- 메모리 설정(장기 메모리 토글)
- 배포 메타데이터

이미지 버전 레지스트리

DynamoDB 테이블은 사용 가능한 에이전트 이미지 버전과 캐시 URIs를 추적하여 버전 관리 및 이전 버전과의 호환성을 지원합니다.

에이전트 구성

시스템 프롬프트

시스템 프롬프트는 에이전트 동작, 성격 및 기능을 정의합니다. 관리자 사용자는 다음을 수행할 수 있습니다.

- Agent Builder UI를 통해 기본 템플릿 편집
- 도구 사용 및 응답 형식 지정에 대한 지침 포함
- 언제든지 기본 템플릿으로 재설정

모델 선택

Agent Builder는 v4.0.0에서 Amazon Bedrock 모델을 지원합니다.

- 모델 공급자: Amazon Bedrock(v4.0.0의 옵션만 해당)
- 모델 선택: Claude, Nova 및 기타 Bedrock 모델
- 모델 파라미터: 온도, max_tokens, top_p 및 모델별 설정

MCP 서버 통합

모델 컨텍스트 프로토콜 서버는 에이전트에게 엔터프라이즈 도구 및 데이터에 대한 액세스 권한을 제공합니다.

- GET /mcp API 엔드포인트를 통한 서버 검색
- 코드 변경이 없는 동적 구성
- 인증 및 엔드포인트 관리
- 에이전트에 대한 도구 기능 노출

스트리밍 및 처리

실시간 스트리밍

Agent Builder는 실시간 응답 스트리밍을 위해 WebSocket에 연결된 AgentCore의 Server-Sent Events(SSE)를 사용합니다.

- Lambda 함수는 AgentCore 런타임에 대한 SSE 연결을 설정합니다.
- 스트림은 API Gateway WebSocket에 브리지됩니다.
- 클라이언트에 token-by-token 응답 전송을 활성화합니다.
- 장기 실행 요청에 대한 연결을 유지합니다.

처리 제약 조건

v4.0.0의 에이전트 처리는 Lambda 실행 제한 시간으로 제한됩니다.

- 최대 처리 시간: 15분
- 동기식 처리 모델
- 대화형 에이전트 및 중간 워크플로에 적합
- v4.1+에 대한 확장된 비동기 지원 계획

메모리 관리

단기 메모리

사용자 지정 MemoryHookProvider를 사용하는 모든 에이전트에 대해 기본적으로 활성화됩니다.

- Strands 콜백 핸들러를 통해 대화 이벤트 캡처
- 컨텍스트 격리를 위해 actorId 및 sessionId별로 구성
- 세션 내에서 대화 컨텍스트를 유지합니다.
- AgentCore 메모리와 자동 통합

장기 메모리

strands_tools의 AgentCore 메모리 도구를 사용하는 선택적 기능:

- Agent Builder UI의 단순 토글
- 기본 설정을 사용한 의미 체계 메모리 전략
- 자연 도구 호출을 통한 에이전트 제어 액세스
- 세션 간에 추출된 인사이트를 저장합니다.
- conversationId를 sessionId로 사용

관찰성

AWS OpenTelemetry Distro(ADOT)

에이전트는 컨테이너 빌드 중에 자동으로 계측됩니다.

- 에이전트 작업을 위한 자동 트레이스 생성

- 서비스 경계를 넘어 분산 추적
- 상관관계 IDs 사용한 구조화된 로깅
- CloudWatch 트랜잭션 검색과 통합

인증 흐름

사용자는 사용자 그룹을 기반으로 DynamoDB에서 IAM 정책을 검색하는 사용자 지정 Lambda 권한 부여자가 검증한 JWT 토큰으로 Amazon Cognito를 통해 인증합니다.

워크플로 빌더

Workflow Builder는 에이전트를 도구 위임 패턴으로 사용하여 여러 에이전트를 조정하는 감독자 에이전트를 생성하여 다중 에이전트 오케스트레이션을 활성화합니다.

워크플로 아키텍처

주요 구성 요소

- 감독자 에이전트: 사용자 요청을 수신하고 특수 에이전트에게 위임하는 진입점 에이전트
- 전문 에이전트: 감독자를 위한 도구로 등록된 Agent Builder 사용 사례
- 에이전트 레지스트리: 에이전트 구성 및 메타데이터를 저장하는 DynamoDB 테이블
- 오케스트레이션 계층: 에이전트의 SDK 구현을 도구 패턴으로 스트랜드합니다.

에이전트 인스턴스화

로컬 에이전트 생성

모든 특수 에이전트는 동일한 AgentCore 런타임 내에서 로컬로 인스턴스화됩니다.

1. DynamoDB에서 에이전트 구성을 검색합니다.
2. 각 Agent Builder 에이전트의 로컬 인스턴스를 생성합니다.
3. 각 에이전트는 자체 MCP 서버 연결을 유지합니다.
4. 감독자 에이전트가 특수 에이전트를 도구로 등록
5. Strands SDK는 에이전트 선택 및 위임을 관리합니다.

배포 계획

이 섹션에서는 배포 계획을 위한 비용, 보안, 리전 및 할당량 고려 사항에 대해 설명합니다.

> **Important**
>
> 이 솔루션은 AI 생성 모델에 액세스하기 위한 기본 서비스로 Amazon Bedrock을 활용합니다. 먼저 모델에 대한 액세스를 요청해야 솔루션 내에서 모델을 사용할 수 있습니다. 자세한 내용은 Amazon Bedrock 사용 설명서의 모델 액세스를 참조하세요.

지원되는 AWS 리전

> **Important**
>
> 이 솔루션은 일부 AWS 리전에서 현재 사용할 수 없는 Amazon Bedrock 및 Amazon Kendra 서비스를 선택적으로 사용합니다. 이러한 서비스를 사용할 수 있는 AWS 리전에서이 솔루션을 시작해야 합니다. 리전별 AWS 서비스의 최신 가용성은 AWS 리전 서비스 목록을 참조하세요.

AWS의 생성형 AI Application Builder는 다음 AWS 리전에서 지원됩니다.

리전 이름	
미국 동부(오하이오)	캐나다(중부)
미국 동부(버지니아 북부)	유럽(프랑크푸르트)
미국 서부(캘리포니아 북부)	유럽(아일랜드)
미국 서부(오리건)	유럽(런던)
아시아 태평양(뭄바이)	유럽(밀라노)
아시아 태평양(서울)	유럽(파리)
아시아 태평양(싱가포르)	유럽(스톡홀름)

리전 이름	
아시아 태평양(시드니)	Middle East (Bahrain)
아시아 태평양(도쿄)	남아메리카(상파울루)

> **Note**
>
> 배포에서 AWS 외부에서 액세스한 파운데이션 모델을 사용하는 경우 모델 공급자에게 APIs를 사용할 수 있는 리전을 확인하세요. 특정 리전에서만 APIs 사용할 수 있는 경우 지연 시간이 길거나 시간 초과가 발생할 수 있습니다. 또한 조직의 법률 및 규정 준수 팀에 문의하여 리전 경계를 넘어가는 데이터의 고려 사항을 평가하는 것이 중요합니다.

비용

이 AWS 솔루션을 사용하면 사용하는 리소스에 대해서만 비용을 지불하며 최소 요금이나 설정 요금은 없습니다. 사용자는 생성형 AI 사용 사례 및를 시작하는 데 사용되는 대시보드와 배포된 사용 사례에 대한 비용을 지불합니다. 배포된 사용 사례의 비용은 구성에 따라 다릅니다. 구성 예:

1. 매월 약 20 USD의 비용이 드는 간단한 배포 대시보드입니다.
2. 문서에 액세스하지 않고 Amazon Bedrock으로 구동되는 미국 동부(버지니아 북부)에서 실행되는 기본 설정으로 배포된 간단한 프로덕션 지원 챗봇 사용 사례로, 매월 약 200 USD의 비용도 듭니다.
3. Amazon VPC 사용 사례에서 수십만 개 이상의 문서에 대해 하루에 8,000개의 쿼리를 지원하는 확장 시스템으로, 요금은 매월 약 1,500 USD입니다. 사용 사례 비용은 모델 공급자가 다른 텍스트 사용 사례, 검색 증강 생성(RAG)이 활성화된 상태 또는 활성화되지 않은 상태 등과 같은 구성에 따라 달라집니다.

워크로드 설명	예상 비용(USD/월)
배포 대시보드의 샘플 비용	$20/월
텍스트 기반 개념 증명에 대한 샘플 비용 (배포 대시보드 및 텍스트 사용 사례 1개, 하루에 약 100건의 상호 작용 포함)	$40/월

워크로드 설명	예상 비용(USD/월)
확장성이 뛰어난 생성형 AI 쿼리 엔진의 샘플 비용 (배포 대시보드, 텍스트 사용 사례 1개, VPC가 활성화된 상태에서 하루에 ~8K 쿼리가 있는 최대 100K 개의 RAG용 Amazon Kendra 인덱스 포함	$1,500/월
에이전트 기반 개념 증명에 대한 샘플 비용 (배포 대시보드, Amazon Bedrock 지식 기반 및 Amazon Bedrock 가드레일이 활성화된 Bedrock 에이전트 사용 사례 1개, 하루에 최대 100개의 상호 작용 포함)	$840/월
MCP 서버의 샘플 비용 (배포 대시보드, Lambda 통합을 위한 게이트웨이 메서드가 포함된 MCP 서버 사용 사례 1개, 하루에 최대 100개의 도구 호출 포함)	$22/월
에이전트 빌더의 샘플 비용 (배포 대시보드, MCP 통합 및 장기 메모리가 활성화된 Agent Builder 사용 사례 1개, 하루에 최대 100개의 상호 작용 포함)	$55/월
Workflow Builder의 샘플 비용 (배포 대시보드, 에이전트 빌더 에이전트 3명이 있는 워크플로 1개, 하루에 최대 100개의 상호 작용 포함)	$109/월

Important

이 예제는 특정 워크로드에 대한 비용을 추정하는 데 도움을 주기 위한 것입니다. 다양한 LLMs, 구성 또는 AWS 서비스를 사용하면 비용이 변경될 수 있습니다(예: 서버리스/온디맨드

결제 vs. 프로비저닝된/시간 청구). 비용을 관리하려면 AWS Cost Explorer를 통해 예산을 생성하는 것이 좋습니다. 요금은 변경될 수 있습니다. 자세한 내용은이 솔루션에 사용되는 각 AWS 서비스의 요금 웹 페이지를 참조하세요.

배포 대시보드 실행을 위한 샘플 비용

다음 표에는 기본 파라미터와 한 달 동안 미국 동부(버지니아 북부) 리전의 활성 사용자 100명이 포함된 배포 대시보드의 비용 내역이 나와 있습니다.이 비용은 월 약 20 USD입니다.

AWS 서비스	측정 기준	비용[USD]
API Gateway, DynamoDB, CloudFront, Amazon S3, Lambda, Systems Manager 파라미터 스토어	캐싱이 활성화되지 않은 상태에서 매월 5,000건의 512KB REST API 호출	1.97 USD
Amazon Cognito	고급 보안 기능이 활성화되어 있고 SAML 또는 OIDC 페더레이션을 통해 로그인하는 사용자가 없는 월 100명의 활성 사용자	5.55 USD
AWS WAF	규칙 그룹이 없는 1개의 웹 ACL 및 7개의 정의된 규칙에서 10,000개의 웹 요청	12.60 USD
총 배포 대시보드 비용		20.12 USD

텍스트 기반 개념 증명에 대한 샘플 비용

배포 대시보드에는 지정된 시간에 많은 사용 사례가 배포될 수 있습니다. 다음 표는 LLM을 사용하여 하루에 100개의 쿼리를 수행하는 비즈니스 사용자 1명에 대해 RAG 없이 배포된 사용 사례의 비용 분석을 보여줍니다. 쿼리는 WebSocket에서 문자 메시지로 전송되고 응답은 스트리밍이 활성화되어 있다는 가정하에 토큰으로 다시 스트리밍됩니다. Amazon Bedrock Nova Pro 모델을 사용하면이 사용 사례를 실행하는 데 드는 비용은 월 약 20 USD입니다.

AWS 서비스	측정 기준	비용[USD]
API Gateway(WebSocket), CloudFront, Lambda, Amazon S3, AWS Systems Manager 파라미터 스토어	하루에 100건의 채팅 상호 작용. 평균 메시지 크기는 메시지당 32KB, 연결당 5분입니다.	0.61 USD
CloudWatch	실험을 위해 상세 정보 모드가 켜져 있는 1.5GB CloudWatch 로그	7.23 USD
Amazon DynamoDB	대화 기록 테이블, 1GB 스토리지 LLM 구성 테이블, 1GB 스토리지	3.05 USD
사용 사례 비용의 소계(LLMs 제외)		10.89 USD
Amazon Bedrock(Nova Pro)	하루에 100건의 상호 작용에 대한 가정: * 일일 190K000개의 입력 토큰에 대한 월별 비용 = 0.152 USD × 30 * 일일 16K000개의 출력 토큰에 대한 월별 비용 = 0.0512 USD × 30	6.10 USD
Amazon Bedrock(Nova Pro)을 사용한 총 애플리케이션 비용	10.89 USD(사용 사례 비용) + 6.10 USD(Amazon Bedrock 비용)	17.00 USD

> **Note**
>
> AWS 네트워크 외부의 서비스에 대한 추론 호출 비용은 이러한 추정치에 포함되지 않습니다. AWS 모델 공급자를 사용하지 않는 경우 LLM 공급자의 요금 안내서를 참조하세요.

AWS 서비스에 대한 요금 가이드는 Amazon Bedrock 요금 및 Amazon SageMaker AI 요금에서 확인할 수 있습니다.

확장성이 뛰어난 생성형 AI 쿼리 엔진의 샘플 비용

다음 표에는 Amazon Bedrock의 Nova Pro 모델을 LLM으로 사용하는 RAG 지원 사용 사례의 비용 내역이 나와 있습니다. Bedrock 지식 기반이 추가되면이 사용 사례 비용은 월 약 1,300 USD입니다.

AWS 서비스	측정 기준	비용[USD]
API Gateway(WebSocket)	하루에 8,000건의 채팅 상호 작용. 평균 메시지 크기는 메시지당 32KB, 연결당 5분입니다.	38.89 USD
CloudFront	인터넷으로 전송된 100GB 데이터와 오리진으로 전송된 1GB 데이터가 포함된 월별 요청 240,000개	8.76 USD
Amazon Bedrock(Nova Pro)	가정: 입력 토큰 = promptTemplate(400) + 컨텍스트(400)+ chatHistory(1080) + 쿼리 입력 토큰(20)= 1,900 출력 토큰 = 160(평균) 하루에 8,000개의 트랜잭션을 사용하는 경우 일일 입력 토큰 비용(1,900 x 8,000 = 토큰 15,200,000개 x 토큰당 가격 0.0008/1000) 일일 출력 토큰 비용(160 x 8,000 = 토큰 1,280,000개 x 토큰당 가격 0.0032/1000)	487.80 USD

AWS 서비스	측정 기준	비용[USD]
	월별 비용((12.16 USD + 4.10 USD) x 30)	
CloudWatch	로그에 대해 수집된 5GB 데이터를 사용하는 지표 24개와 대시보드 1개	9.72 USD
DynamoDB	DynamoDB 테이블은 최대 1KB 데이터, 하루에 8,000회 읽기 및 쓰기의 각 레코드에 대한 대화 기록을 추적합니다.	11.70 USD
Lambda	컨테이너 크기 - 128MB, 임시 512MB 스토리지, 권한 부여에 사용되는 Lambda 함수 2개 컨테이너 크기 - 256MB, 512MB 임시 스토리지, 초당 요청 5개, 평균 컴퓨팅 시간 20초	20.89 USD
총 사용 사례 비용		$577.76/월 + 지식 기반 비용 (아래 참조)

> **Note**
>
> AWS 네트워크 외부의 서비스에 대한 API 호출 비용은 이러한 추정치에 포함되지 않습니다. Amazon Bedrock을 사용하지 않는 경우 LLM 공급자의 요금 안내서를 참조하세요.

지식 기반 추가 비용

지식 기반 비용은 사용되는 지식 기반 유형과 지식 기반에서 사용하는 지원 벡터 스토어(Bedrock의 경우)에 따라 달라집니다. 지식 기반을 프로비저닝하고 관리하는 것은 솔루션 범위를 벗어납니다.

Amazon Bedrock 지식 기반

이 솔루션은 Amazon Bedrock 지식 기반과 관련된 리소스를 관리하거나 프로비저닝하지 않습니다. Amazon Bedrock은 지식 기반 기능 자체를 사용하는 데 비용이 발생하지 않지만 각 쿼리에서 사용 사례에 사용되는 임베딩 모델의 사용에 대해서는 요금이 부과됩니다. 또한 지식 기반(예: Amazon OpenSearch Service의 인덱스 또는 Amazon Relational Database Service 내의 데이터베이스)의 백업 벡터 스토어에는 여기에서 제공하거나 계산할 수 없는 관련 비용이 발생합니다.

위의 확장성이 뛰어난 생성형 AI 쿼리 엔진 시나리오의 경우 Amazon Bedrock 임베딩 모델을 호출하기 위해이 서비스로 인해 발생하는 비용은 다음과 같습니다.

AWS 서비스	측정 기준	비용[USD]
Amazon Bedrock(Amazon Titan Text Embeddings V2)	쿼리당 1,900개의 입력 토큰이 있는 일일 쿼리 8,000개 = 토큰 15,200,000개 = 일일 0.30 USD. 일일 비용 x 30일 = $9.00 USD 월별 비용	9.00 USD
Amazon OpenSearch Service(서버리스) 샘플 사용량	4 x OpenSearch 컴퓨팅 유닛(OCU)(청구 가능한 최소 금액) = 일일 23.04 USD를 사용하는 기본 서버리스 구성 일일 비용 x 30일 = 691.20 USD **Note** 이렇게 하면 일부 워크로드에는 더 많은 OCUs 필요하지만 기존 프로비저닝된 OpenSearch 리소스가 있는 고객은 여기에서 비용이 적게 들기 때문	691.20 USD

AWS 서비스	측정 기준	비용[USD]
	에 대략적인 추정치를 얻을 수 있습니다.	
총 추가 비용		700.20 USD

Amazon Kendra

솔루션은 Kendra 인덱스를 프로비저닝하거나 직접 가져올 수 있습니다. 위의 확장성이 뛰어난 생성형 AI 쿼리 엔진에 적합한 구성을 실행하는 데 드는 비용은 다음과 같습니다.

AWS 서비스	측정 기준	비용[USD]
Amazon Kendra	0~50개의 데이터 소스가 있는 Amazon Kendra Enterprise Edition을 사용하여 하루에 0~8,000개의 쿼리와 최대 100,000개의 문서	1,008.00 USD

> **Note**
>
> 사용 사례 간에 Amazon Kendra 인덱스를 공유할 수 있지만 이로 인해 인덱스당 쿼리 수가 증가할 수 있습니다. Amazon Kendra Enterprise 에디션을 벗어나는 경우 추가 요금이 적용됩니다.

사용 사례에 대해 Amazon VPC를 활성화하는 데 드는 증분 비용

다음 표에는 두 AZs.

AWS 서비스	측정 기준	비용[USD]
Amazon NAT 게이트웨이	가정: 각 AZ에 NAT 게이트웨이가 있는 2개의 AZ 배포. NAT Gateway를 통해 처리되는 데	74.70 USD

AWS 서비스	측정 기준	비용[USD]
	이터 100GB 730시간, 매월 처리되는 데이터 100GB	
AWS PrivateLink(VPC 엔드포인트)	가정: AZ 배포 2개, 각 AZ에 프라이빗 서브넷 1개, 탄력적 네트워크 인터페이스(ENIs. 6개의 VPC 엔드포인트, VPC 엔드포인트당 2ENIs, 한 달에 1,024GB 데이터가 처리되는 730시간	97.84 USD
퍼블릭 IPv4 주소	가정: AZ 배포 2개, 각 퍼블릭 서브넷에 NAT 게이트웨이가 있는 각 AZ에 퍼블릭 서브넷 1개. 활성 퍼블릭 IPv4 1개로 구성된 각 NAT 게이트웨이. 활성 퍼블릭 IPv4 주소 2개 x 한 달에 730시간 x 시간당 0.005 USD 요금 = 7.3 USD	7.30 USD
추가 요금 (Amazon VPC의 경우)		179.93 USD

프로비저닝된 처리량 사용 시 비용에 미치는 영향

프로비저닝된 처리량 비용은 프로비저닝한 모델 유형과 약정 기간 및 약정 기간 동안 선택한 모델 단위에 따라 달라집니다. 프로비저닝된 처리량 사용과 관련된 추가 비용이 있습니다.

자세한 내용과 up-to-date 요금은 Bedrock 요금을 참조하세요.

교차 리전 추론 사용 비용

교차 리전 추론을 사용하기 위한 라우팅 또는 데이터 전송에는 추가 비용이 들지 않습니다. 소스 또는 기본 리전과 동일한 가격의 모델 토큰당 요금을 지불합니다.

에이전트 기반 개념 증명에 대한 샘플 비용

Amazon Bedrock Agents를 사용하면 추가 기능과 함께 지원 모델 및 지식 기반(RAG가 활성화된 경우)과 같이 에이전트로 구성된 구성 요소를 기준으로 요금이 청구됩니다. 다음 표는 온디맨드 Claude 3.5 Sonnet 모델, Amazon Bedrock 지식 기반 및 Amazon Bedrock 가드레일로 구성된 Bedrock 에이전트 사용 사례의 비용 분석을 보여줍니다.

Amazon Bedrock 지식 기반을 추가하는 비용과 마찬가지로이 솔루션은 Amazon Bedrock 에이전트와 관련된 리소스를 관리하거나 프로비저닝하지 않습니다. 또한이 솔루션은 Amazon Bedrock 지식 기반을 사용하는 데 비용이 발생하지 않지만 다음에 대한 비용이 발생합니다.

- 전송되는 각 쿼리에 임베딩 모델 사용
- 지식 기반의 백업 벡터 스토어(예: Amazon OpenSearch Service의 인덱스 또는 Amazon RDS 내의 데이터베이스)

다음 표에서는 쿼리당 1,900개의 입력 토큰과 160개의 출력 토큰을 사용하여 하루에 100개의 상호 작용을 가정합니다.

> **Note**
>
> 이 샘플 Bedrock Agent 사용 사례의 경우 외부 API를 사용하도록 구성된 작업 그룹이 있는 경우 해당 비용이 추가됩니다. 이 테이블의 계산 범위를 벗어납니다.

AWS 서비스	측정 기준	비용[USD]
API Gateway(WebSocket), CloudFront, Lambda, Amazon S3, Systems Manager 파라미터 스토어	일일 채팅 상호 작용 100건, 메시지당 평균 메시지 크기 32KB, 연결당 5분	0.61 USD
CloudWatch	실험을 위해 상세 정보 모드가 켜져 있는 1.5GB CloudWatch Logs	7.23 USD
DynamoDB	1KB 레코드 크기 및 1GB 스토리지에 대한 LLM 구성 테이블	0.25 USD

AWS 서비스	측정 기준	비용[USD]
비용 소계(LLMs 제외)		8.09 USD
Anthropic Claude 3.5 Sonnet	* 일일 190K 개의 입력 토큰에 대한 일일 비용(0.003/1,000개의 토큰) = 0.57 USD + 일일 비용 × 30일 = 17.10 USD * 일일 16K 출력 토큰의 일일 비용(0.015/1,000 토큰) = 0.24 USD + 일일 비용 × 30일 = 7.20 USD	24.30 USD
Amazon Bedrock 지식 기반용 Amazon Bedrock(Amazon Titan Text Embeddings V2)	일일 190K 개의 입력 토큰에 대한 일일 비용(0.00002/1000 토큰) = 0.004 일일 비용 × 30일 = 0.12 USD	0.12 USD
Amazon OpenSearch Service(서버리스) 샘플 사용량	4 × OpenSearch 컴퓨팅 유닛(OCU)(청구 가능한 최소 금액) = 일일 23.04 USD를 사용한 기본 서버리스 구성 일일 비용 × 30일 = 691.20 USD	691.20 USD

AWS 서비스	측정 기준	비용[USD]
Amazon Bedrock Guardrails	190K 토큰은 약 760K(190,000 × 4) 문자 및 3,800 텍스트 단위(760K 문자/200)와 동일합니다. 콘텐츠 필터, 개인 식별 정보(PII) 필터, 민감한 정보 필터(정규 표현식) 및 단어 필터로 구성된 가드레일을 고려합니다. 일일 콘텐츠 필터 비용(0.75/1000 텍스트 단위) + PII 필터 비용(0.1/1000 텍스트 단위) + 민감한 정보 필터(정규) + 단어 필터 = 2.85 + 0.38 + 0 USD + 0 USD 월별 비용 = 일일 비용 × 30일 = 96.90 USD	96.90 USD
Anthropic Claude 3.5 Sonnet에서 지원하는 에이전트의 총 애플리케이션 비용	8.09 USD(사용 사례 비용) + 812.52 USD(기타 에이전트 구성)	820.61 USD

> **Note**
>
> AWS 모델 공급자를 사용하지 않는 경우 LLM 공급자의 요금 안내서를 참조하세요. AWS 서비스에 대한 요금 가이드는 Amazon Bedrock 요금 및 Amazon SageMaker AI 요금에서 확인할 수 있습니다.

MCP 서버의 샘플 비용

MCP 서버 사용 사례를 통해 Amazon Bedrock AgentCore에서 모델 컨텍스트 프로토콜 서버를 배포하고 관리할 수 있습니다. 다음 표는 게이트웨이 메서드를 사용하여 기존 Lambda 함수를 래핑하는 MCP 서버 사용 사례의 비용 분석을 보여줍니다.

솔루션은 AgentCore Gateway 배포 및 구성을 관리합니다. 다음과 같은 요금이 부과됩니다.

- 인프라 비용(API Gateway, Lambda, DynamoDB, CloudWatch, S3)
- AgentCore Gateway 사용량(도구 간접 호출당)
- Lambda 함수 실행 비용(Lambda 대상이 있는 게이트웨이 메서드의 경우)
- 외부 API 비용(해당하는 경우 API 또는 MCP Server 대상이 있는 게이트웨이 메서드의 경우)

항목	계산	비용
Amazon API Gateway(REST API)	하루에 100개의 도구 호출 × 30일 = 매월 3,000개의 요청	0.05 USD
AWS Lambda(오케스트레이션)	하루에 100회 호출 × 30일 × 1초 평균 × 512MB = 매월 3,000GB-초	0.05 USD
Amazon DynamoDB	매월 3,000개의 읽기/쓰기 요청 + 1GB 스토리지	0.15 USD
Amazon CloudWatch	3,000회 호출에 대한 표준 모니터링 및 로깅	1.00 USD
Amazon S3	구성 스토리지 및 로그(최소 사용량)	0.25 USD
Amazon Bedrock AgentCore Gateway	매월 3,000개의 도구 호출	0.05 USD
대상 Lambda 함수	하루에 100회 호출 × 30일 × 0.5초 × 128MB = 매월 1,500GB-초	0.25 USD

항목	계산	비용
총 월별 비용	1.75 USD(인프라) + 0.05 USD(AgentCore Gateway)	1.80 USD

> **Note**
>
> 비용은 배포 방법(게이트웨이 대 런타임), 대상 유형 및 사용 패턴에 따라 달라집니다. 런타임 메서드 배포에는 게이트웨이 요금 대신 AgentCore 런타임 요금이 발생합니다. 외부 API 비용 및 사용자 지정 컨테이너 호스팅 비용은 추가 비용입니다.

Agent Builder의 샘플 비용

Agent Builder를 사용하면 Amazon Bedrock AgentCore에서 사용자 지정 에이전트를 생성하고 배포할 수 있습니다. 다음 표는 Claude 3.5 Sonnet, MCP 서버 통합 및 장기 메모리가 활성화된 상태로 구성된 Agent Builder 사용 사례의 비용 분석을 보여줍니다.

솔루션은 AgentCore 런타임 배포 및 구성을 관리합니다. 다음과 같은 요금이 부과됩니다.

- 인프라 비용(API Gateway, Lambda, DynamoDB, CloudWatch, S3)
- AgentCore 런타임 소비(실제 에이전트 실행 시간을 기준으로 한 CPU 및 메모리 시간)
- 파운데이션 모델 추론(입력 및 출력 토큰)
- AgentCore 메모리(단기 이벤트 및 장기 스토리지/검색)

다음 표에서는 하루에 100개의 상호 작용을 쿼리당 1,900개의 입력 토큰과 160개의 출력 토큰으로 가정하고, 평균 에이전트 실행 시간은 상호 작용당 5초입니다.

AWS 서비스	측정 기준	비용[USD]
API Gateway(WebSocket), CloudFront, Lambda, Amazon S3, Systems Manager 파라미터 스토어	일일 채팅 상호 작용 100건, 메시지당 평균 메시지 크기 32KB, 연결당 5분	0.61 USD

AWS 서비스	측정 기준	비용[USD]
CloudWatch	실험을 위해 상세 정보 모드가 켜져 있는 1.5GB CloudWatch Logs	7.23 USD
DynamoDB	1KB 레코드 크기 및 1GB 스토리지에 대한 LLM 구성 테이블	0.25 USD
인프라 비용 소계		8.09 USD
Amazon Bedrock AgentCore 런타임	* CPU: vCPU 1개 × 5초 × 상호 작용 100개 = vCPU 125초/일 = vCPU 0.140시간/일 + 일일 비용: 0.140 × 0.0895 USD = 0.013 USD + 월별 비용: 0.013 USD × 30 = 0.38 USD * 메모리: 512MB(0.5GB) × 5초 × 100 상호 작용 = 250GB-초/일 = 0.069GB-시간/일 + 일일 비용: 0.069 × 0.00945 = 0.0007 USD + 월별 비용: 0.0007 USD × 30 = 0.02 USD	0.40 USD
Anthropic Claude 3.5 Sonnet	* 일일 190K 개의 입력 토큰에 대한 일일 비용(0.003/1,000개의 토큰) = 0.57 USD + 일일 비용 × 30일 = 17.10 USD * 일일 16K 출력 토큰의 일일 비용(0.015/1,000 토큰) = 0.24 USD + 일일 비용 × 30일 = 7.20 USD	24.30 USD

AWS 서비스	측정 기준	비용[USD]
Amazon Bedrock AgentCore 메모리	* 단기 메모리: 100개의 새 이벤트/일 × 0.25 USD/1,000개의 이벤트 = 0.025 USD/일 + 월별 비용: 0.025 USD × 30 = 0.75 USD * 장기 메모리 스토리지(기본 제공 전략): 레코드 100개 × 레코드 0.75/1,000 USD/월 = 0.075 USD/월 * 장기 메모리 검색: 하루에 100회 검색 × $0.50/1,000회 검색 = $0.05/일 + 월별 비용: $0.05 × 30 = $1.50	2.33 USD
Claude 3.5 Sonnet을 사용하는 Agent Builder의 총 애플리케이션 비용	$8.09(인프라) + $0.40(AgentCore 런타임) + $24.30(모델) + $2.33(메모리)	35.12 USD

Note

AgentCore 런타임 요금은 소비 기반입니다. 실제 비용은 다음에 따라 달라집니다.

- 에이전트 실행 시간(활성 처리 중 CPU 및 메모리 사용량)
- 상호 작용 수 및 복잡성
- MCP 도구 사용(도구 실행을 위한 추가 CPU/메모리)
- 메모리 구성(단기 및 장기 메모리 활성화됨)

자세한 AgentCore 요금은 Amazon Bedrock 요금을 참조하세요.

> **Note**
>
> 외부 APIs 또는 서비스를 호출하는 MCP 서버를 사용하는 경우 해당 비용은 추가 비용이며이 계산 범위를 벗어납니다. 마찬가지로 AgentCore 브라우저 또는 코드 해석기 도구를 사용하는 경우 vCPU 시간당 0.0895 USD, GB 시간당 0.00945 USD의 소비 기반 요금이 적용됩니다.

Workflow Builder의 샘플 비용

Workflow Builder는 여러 Agent Builder 에이전트를 오케스트레이션하는 감독자 에이전트를 생성합니다. 다음 표에는 Claude 3.5 Sonnet 및 장기 메모리가 활성화된 상태로 구성된 감독자 에이전트 1명과 특수 에이전트 빌더 에이전트 3명이 있는 워크플로의 비용 내역이 나와 있습니다.

가정: 하루에 100건의 상호 작용, 상호 작용당 평균 2건의 에이전트 위임, 에이전트당 5초의 실행 시간.

AWS 서비스	측정 기준	비용[USD]
API Gateway(WebSocket), CloudFront, Lambda, Amazon S3, Systems Manager 파라미터 스토어	일일 채팅 상호 작용 100건, 메시지당 평균 메시지 크기 32KB, 연결당 5분	0.61 USD
CloudWatch	실험을 위해 상세 정보 모드가 켜져 있는 1.5GB CloudWatch Logs	7.23 USD
DynamoDB	1KB 레코드 크기 및 1GB 스토리지에 대한 LLM 구성 테이블	0.25 USD
인프라 비용 소계		8.09 USD
Amazon Bedrock AgentCore 런타임(관리자 에이전트)	* CPU: 1 vCPU × 5초 × 100 상호 작용 = 0.140 vCPU-시간/일 × 30 = 0.38 USD * 메모리: 0.5GB × 5초 × 100 상호 작용 = 0.069GB-시간/일 × 30 USD = 0.02 USD	0.40 USD

AWS 서비스	측정 기준	비용[USD]
Amazon Bedrock AgentCore 런타임(전문 에이전트 3개)	* 상호 작용당 평균 2회 위임 = 200개 에이전트 실행/일 * CPU: 1 vCPU × 5초 × 200 = 0.278 vCPU-시간/일 × 30 = 0.75 USD * 메모리: 0.5GB × 5초 × 200 = 0.139GB-시간/일 × 30 = 0.04 USD	0.79 USD
Anthropic Claude 3.5 Sonnet(수퍼바이저 에이전트)	* 입력: 190K 토큰/일 × 0.003/1K USD = 0.57/일 × 30 = 17.10 USD * 출력: 16K 토큰/일 × 0.015/1K USD = 0.24/일 × 30 = 7.20 USD	24.30 USD
Anthropic Claude 3.5 Sonnet(특수화된 에이전트)	* 상호 작용당 평균 2회의 위임 * 입력: 380K 토큰/일 × $0.003/1K = $1.14/일 × 30 = $34.20 * 출력: 32K 토큰/일 × $0.015/1K = $0.48/일 × 30 = $14.40	48.60 USD
Amazon Bedrock AgentCore 메모리(관리자 에이전트)	* 단기: 100개 이벤트/일 × 0.25 USD/1K × 30 = 0.75 USD * 장기 스토리지: 100개 레코드 × 0.75 USD/1K = 0.08 USD * 장기 검색: 100개 검색/일 × 0.50 USD/1K × 30 = 1.50 USD	2.33 USD
Amazon Bedrock AgentCore 메모리(특수화된 에이전트)	* 단기: 200개 이벤트/일 × $0.25/1K × 30 = $1.50 * 장기 스토리지: 200개 레코드 × $0.75/1K = $0.15 * 장기 검색: 200개 검색/일 × $0.50/1K × 30 = $3.00	4.65 USD

AWS 서비스	측정 기준	비용[USD]
에이전트가 3개인 Workflow Builder의 총 애플리케이션 비용	$8.09(인프라) + $1.19(AgentCore 런타임) + $72.90(모델) + $6.98(메모리)	89.16 USD

Note

- 위임률이 높을수록 토큰 소비가 비례하여 증가합니다.

자세한 AgentCore 요금은 Amazon Bedrock 요금을 참조하세요.

보안

AWS 인프라에 시스템을 빌드하면 보안 책임은 사용자와 AWS가 분담합니다. AWS는 호스트 운영 체제, 가상화 계층 및 서비스가 운영되는 시설의 물리적 보안을 포함한 구성 요소를 운영, 관리 및 제어하기 때문에이 공동 책임 모델은 운영 부담을 줄입니다. AWS 보안에 대한 자세한 내용은 AWS 클라우드 보안을 참조하세요.

Amazon Bedrock에서 파운데이션 모델 사용

Amazon Bedrock은 Amazon Nova 모델의 모델 모음을 다른 주요 파운데이션 모델(FMs. Amazon Bedrock을 사용하는 경우 모든 모델은 AWS 인프라 내에서 호스팅됩니다. 즉, Amazon Bedrock을 LLM 공급자로 사용하면 모든 추론 요청이 AWS 네트워크 내에 유지되고 네트워크 트래픽이 리전을 벗어나지 않습니다.

Note

Amazon Bedrock을 통해 사용할 수 있는 모든 파운데이션 모델(FMs)은 AWS에서 관리하고 소유한 AWS 인프라에서 직접 호스팅됩니다. 모델 공급자는 프롬프트 및 연속 또는 Amazon Bedrock 서비스 로그와 같은 고객 데이터에 액세스할 수 없습니다. Amazon Bedrock의 보안 태세에 대한 자세한 내용은 Amazon Bedrock 사용 설명서의 Amazon Bedrock의 데이터 보호를 참조하세요.

IAM 역할

IAM 역할을 통해 고객은 AWS 클라우드의 서비스 및 사용자에게 세분화된 액세스 정책 및 권한을 할당할 수 있습니다. 이 솔루션은 솔루션의 Lambda 함수에 리전 리소스를 생성할 수 있는 액세스 권한을 부여하는 IAM 역할을 생성합니다.

CloudWatch Logs

배포 대시보드 모델 선택 페이지의 추가 설정에서 사용 사례를 배포하는 동안 상세 정보 모드를 활성화할 수 있습니다. 상세 정보 모드를 사용하면 디버깅 및 프롬프트 실험에 도움이 될 수 있는 자세한 CloudWatch 로그를 사용할 수 있습니다.

> **Note**
>
> 상세 정보 모드가 활성화되면 지식 기반(RAG가 활성화된 경우) 및 프롬프트에서 검색된 문서도 로깅되며, 여기에는 민감한 정보가 포함될 수 있습니다.

VPC

솔루션은 Amazon VPC 구성을 위한 두 가지 옵션을 제공합니다.

1. 솔루션이 Amazon VPC를 빌드하도록 합니다.
2. 솔루션 내에서 사용할 수 있도록 자체 Amazon VPC를 관리하고 가져옵니다.

솔루션이 Amazon VPC를 빌드하도록 허용

솔루션이 Amazon VPC를 빌드하도록 허용하는 옵션을 선택하면 기본적으로 CIDR 범위 10.10.0.0/20으로 2-AZ 아키텍처로 배포됩니다. 각 AZ에 퍼블릭 서브넷 1개와 프라이빗 서브넷 1개와 함께 Amazon VPC IP 주소 관리자(IPAM)를 사용할 수 있습니다. 솔루션은 각 퍼블릭 서브넷에 NAT 게이트웨이를 생성하고 프라이빗 서브넷에 ENIs를 생성하도록 Lambda 함수를 구성합니다. 또한이 구성은 라우팅 테이블과 해당 항목, 보안 그룹 및 해당 규칙, 네트워크 ACLs, VPC 엔드포인트(게이트웨이 및 인터페이스 엔드포인트)를 생성합니다.

자체 Amazon VPC 관리

Amazon VPC로 솔루션을 배포할 때 AWS 계정 및 리전의 기존 Amazon VPC를 사용할 수 있습니다. 고가용성을 보장하려면 두 개 이상의 가용 영역에서 VPC를 사용할 수 있도록 하는 것이 좋습니다.

VPC에는 다음과 같은 VPC 엔드포인트와 VPC 및 라우팅 테이블 구성에 대한 관련 IAM 정책도 있어야 합니다.

배포 대시보드 Amazon VPC의 경우

1. DynamoDB의 게이트웨이 엔드포인트입니다.
2. S3의 게이트웨이 엔드포인트입니다.
3. CloudWatch의 인터페이스 엔드포인트입니다.
4. AWS CloudFormation의 인터페이스 엔드포인트입니다.

사용 사례 Amazon VPC의 경우

1. DynamoDB용 게이트웨이 엔드포인트입니다.
2. S3의 게이트웨이 엔드포인트입니다.
3. CloudWatch의 인터페이스 엔드포인트입니다.
4. Systems Manager Parameter Store의 인터페이스 엔드포인트입니다.

> **Note**
>
> 솔루션에는 만 필요합니다`com.amazonaws.region.ssm`.

5. Amazon Bedrock용 인터페이스 엔드포인트(bedrock-runtime, agent-runtime, bedrock-agent-runtime)
6. 선택 사항: 배포에서 Amazon Kendra를 지식 기반으로 사용하는 경우 Amazon Kendra에 대한 인터페이스 엔드포인트가 필요합니다.
7. 선택 사항: 배포에서 Amazon Bedrock 아래의 LLM을 사용하는 경우 Amazon Bedrock에 대한 인터페이스 엔드포인트가 필요합니다.

> **Note**
>
> 솔루션에는 만 필요합니다`com.amazonaws.region.bedrock-runtime`.

8. 선택 사항: 배포에서 LLM에 Amazon SageMaker AI를 사용하는 경우 Amazon SageMaker AI에 대한 인터페이스 엔드포인트가 필요합니다.

> **Note**
>
> 솔루션은 자체 VPC 배포 가져오기 옵션을 사용할 때 VPC 구성을 삭제하거나 수정하지 않습니다. 그러나 VPCs 생성 옵션에서 솔루션에 의해 생성된 모든 VPC는 삭제됩니다. 따라서 스택/배포 간에 솔루션 관리형 VPC를 공유할 때는 주의해야 합니다.
> 예를 들어 배포 A는 VPC 생성 옵션을 사용합니다. 배포 B는 배포 A에서 생성한 VPC를 사용하여 자체 VPC 가져오기를 사용합니다. 배포 B 전에 배포 A를 삭제하면 VPC가 삭제되었으므로 배포 B가 더 이상 작동하지 않습니다. 또한 배포 B는 Lambda 함수에서 생성한 ENIs를 사용하므로 배포 A를 삭제하면 오류가 발생하고 잔여 리소스가 보존될 수 있습니다.

Amazon CloudFront

이 솔루션은 Amazon S3 버킷에 호스팅된 웹 콘솔을 배포합니다. 지연 시간을 줄이고 보안을 개선하기 위해이 솔루션에는 솔루션의 웹 사이트 버킷 콘텐츠에 대한 퍼블릭 액세스를 제공하는 CloudFront 사용자인 오리진 액세스 ID가 있는 CloudFront 배포가 포함됩니다. 자세한 내용을 알아보려면 Amazon CloudFront 개발자 안내서의 오리진 액세스 ID(OAI)를 사용하여 Amazon S3 콘텐츠에 대한 액세스 제한을 참조하세요.

> **Note**
>
> CloudFront의 계정 수준 소프트 할당량 제한은 응답 헤더 정책 20개입니다. 이 솔루션은 보안을 위해 사용자 지정 응답 헤더 정책을 생성합니다. AWS의 생성형 AI Application Builder 또는 사용 사례를 20개 이상 배포한 경우 할당량 한도에 도달하여 새 배포가 실패할 수 있습니다.

이 문제를 해결하려면 다음 단계에 따라 AWS Service Quotas 콘솔에서 응답 헤더 정책 할당량에 대한 할당량 증가를 요청할 수 있습니다.

1. AWS Service Quotas 콘솔을 엽니다.
2. 탐색 창에서 AWS 서비스를 선택합니다.
3. Amazon CloudFront를 검색하고 선택합니다.
4. 응답 헤더 정책 할당량으로 스크롤하고 할당량 증가 요청을 선택합니다.
5. 프롬프트에 따라 AWS 계정에 대한 할당량 한도 증가를 요청합니다.

응답 헤더 정책 할당량을 늘리면 할당량 제한으로 인해 AWS 또는 해당 사용 사례에서 생성형 AI Application Builder의 새 배포가 실패하지 않도록 할 수 있습니다.

할당량

서비스 할당량(제한이라고도 함)은 AWS 계정의 최대 서비스 리소스 또는 작업 수입니다.

이 솔루션의 AWS 서비스에 대한 할당량

이 솔루션에 구현된 각 서비스의 할당량이 충분한지 확인하세요. 자세한 내용은 AWS 서비스 할당량을 참조하세요.

다음 링크를 선택하여 해당 서비스에 대한 페이지로 이동합니다. 페이지를 전환하지 않고 설명서의 모든 AWS 서비스에 대한 서비스 할당량을 보려면 PDF 대신 서비스 엔드포인트 및 할당량 페이지에서 정보를 확인하세요.

Amazon Bedrock AgentCore 할당량

Agent Builder 배포의 경우 다음 Amazon Bedrock AgentCore 서비스 할당량에 유의하세요.

할당량	미국 동부(버지니아 북부)	기타 리전
계정당 활성 세션 워크로드	1000	500
계정당 총 에이전트 수	1,000	1,000
계정당 버전	1,000	1,000

솔루션 배포

이 솔루션은 AWS CloudFormation 템플릿 및 스택을 사용하여 솔루션의 배포를 자동화합니다. CloudFormation 템플릿은이 솔루션에 포함된 AWS 리소스와 해당 속성을 지정합니다. CloudFormation 스택은 템플릿에 설명된 리소스를 프로비저닝합니다.

배포 프로세스 개요

솔루션을 시작하기 전에이 가이드에서 설명하는 비용, 아키텍처, 보안 및 기타 고려 사항을 검토하세요.

> **Important**
>
> Amazon Bedrock을 사용하려는 경우 모델에 대한 액세스를 요청해야 모델을 사용할 수 있습니다. 자세한 내용은 Amazon Bedrock 사용 설명서의 모델 액세스를 참조하세요.

배포 시간: 약 10분

1단계: 배포 대시보드 스택 시작

2단계: 사용 사례 배포

3단계: 배포 대시보드 마법사를 사용하여 사용 사례 배포

4단계: 배포 후 구성

선택적으로 배포 대시보드 UI 또는 APIs.

- 독립 실행형 텍스트 사용 사례 배포
- 독립 실행형 Bedrock Agent 사용 사례 배포

DynamoDB 채팅 구성을 제공할 수도 있습니다.

> **Important**
>
> 이 솔루션은이 솔루션 사용에 대한 운영 지표(“데이터”)를 AWS로 전송합니다. 당사는 고객이이 솔루션과 관련 서비스 및 제품을 사용하는 방법을 더 잘 이해하기 위해이 데이터를 사용합니다. AWS의이 데이터 수집에는 AWS 개인 정보 보호 정책이 적용됩니다.

AWS CloudFormation 템플릿

배포하기 전에이 솔루션에 대한 CloudFormation 템플릿을 다운로드할 수 있습니다.

View template

generative-ai-application-builder-on-aws.template -이 템플릿을 사용하여 솔루션 및 모든 관련 구성 요소를 시작합니다. 기본 구성은 이 솔루션 섹션의 AWS 서비스에 있는 코어 및 지원 솔루션을 배포하지만 특정 요구 사항에 맞게 템플릿을 사용자 지정할 수 있습니다.

> **Note**
>
> AWS CloudFormation 리소스는 AWS Cloud Development Kit(AWS CDK) 구문에서 생성됩니다.

이 AWS CloudFormation 템플릿은 AWS 클라우드의 AWS에 생성형 AI 애플리케이션 빌더를 배포합니다.

1단계: 배포 대시보드 스택 시작

이 섹션의 단계별 지침에 따라 솔루션을 구성하고 계정에 배포합니다.

배포 시간: 약 10분

1. AWS Management Console에 로그인하고 버튼을 선택하여 `generative-ai-application-builder-on-aws.template` CloudFormation 템플릿을 시작합니다.

   Launch solution

2. 이 템플릿은 기본적으로 미국 동부(버지니아 북부) 리전에서 시작됩니다. 다른 AWS 리전에서 솔루션을 실행하려면 콘솔 탐색 표시줄의 리전 선택기를 사용합니다.

   > **Note**
   >
   > 이 솔루션은 현재 일부 AWS 리전에서 사용할 수 없는 Amazon Kendra 및 Amazon Bedrock을 사용합니다. 이러한 기능을 사용하는 경우 이러한 서비스를 사용할 수 있는 AWS 리전에

서이 솔루션을 시작해야 합니다. 리전별 최신 가용성은 AWS 리전 서비스 목록을 참조하세요.

3. 스택 생성 페이지에서 Amazon S3 URL 텍스트 상자에 올바른 템플릿 URL이 있는지 확인하고 다음을 선택합니다.
4. 스택 세부 정보 지정 페이지에서 솔루션 스택 이름을 할당합니다. 문자 제한 이름 지정에 대한 자세한 내용은 AWS Identity and Access Management 사용 설명서의 IAM 및 STS 제한을 참조하세요.
5. 파라미터에서 이 솔루션 템플릿의 파라미터를 검토하고 필요에 따라 수정합니다. 이 솔루션은 다음과 같은 기본값을 사용합니다.

파라미터	기본값	설명
관리자 사용자 이메일	No	배포 대시보드에 액세스할 수 있는 관리자 사용자의 이메일 주소입니다. 제공된 경우 사용 사례를 배포하고 관리할 수 있는 권한이 있는 Amazon Cognito 그룹과 사용자가 생성됩니다. `placeholder@example.com` 를 사용하여 그룹을 생성할 수도 있지만 사용자는 생성할 수 없습니다. 사용자 풀 설정에 대한 자세한 내용은 수동 사용자 풀 구성을 참조하세요.
VpcEnabled	No	배포 대시보드를 VPC 내에 배포해야 합니까?
CreateNewVpc	No	VpcEnabled가 인 경우에만 사용할 수 있습니다`Yes`. 값이 인 경우 `Yes`스택은 VPC를 생성하고 생성된 VPC 내에 솔루션을 배포합니다. VpcEnabled가 `Yes` 이고 CreateNewVpc가 `No`인 경우

파라미터	기본값	설명
		기존 VPC 구성(ExistingVpcId, ExistingPrivateSubnetIds, ExistingSecurityGroupIds, VpcAzs)을 제공해야 합니다.
IPAMPoolId	(선택 사항 입력)	IPAM을 구성하고 생성된 ID를 입력으로 제공하여이 스택의 배포에서 사용해야 하는 IP 주소 범위를 할당할 수 있습니다. IPAM에 대한 자세한 내용은 IPAM 작동 방식을 참조하세요.
DeployUI	Yes	웹 사용자 인터페이스(및 웹 배포에 필요한 AWS 리소스) 없이 배포 대시보드를 배포할 수 있습니다. 이 경우 솔루션은 REST API 엔드포인트를 포함한 모든 인프라를 배포합니다. 이 옵션은 자체 웹 인터페이스를 배포 대시보드 APIs와 통합하는 데 유용합니다.
ExistingVpcId	(선택 사항 입력)	생성한 기존 VPC에 솔루션을 배포하려는 경우에만 필요합니다.
ExistingPrivateSubnetIds	(선택 사항 입력)	생성한 기존 VPC에 솔루션을 배포하려는 경우에만 필요합니다. Lambda 함수는이 서브넷에 배포됩니다.

파라미터	기본값	설명
ExistingSecurityGroupIds	(선택 사항 입력)	생성한 기존 VPC에 솔루션을 배포하려는 경우에만 필요합니다. 보안 그룹에 아웃바운드 TCP 연결에 대한 권한이 있는지 확인합니다.
VpcAzs	(선택 사항 입력)	생성한 기존 VPC에 솔루션을 배포하려는 경우에만 필요합니다.
CognitoDomainPrefix	(선택 사항 입력)	생성한 기존 Amazon Cognito 사용자 풀에 솔루션을 배포하려는 경우에만 필요합니다. 값을 제공하지 않으면 솔루션이 값을 생성합니다.
ExistingCognitoUserPoolId	(선택 사항 입력)	생성한 기존 Amazon Cognito 사용자 풀에 솔루션을 배포하려는 경우에만 필요합니다.
ExistingCognitoUserPoolClient	(선택 사항 입력)	생성한 기존 Amazon Cognito 사용자 풀에 솔루션을 배포하려는 경우에만 필요합니다. 값을 제공하지 않으면 솔루션이 사용자 풀 클라이언트를 생성합니다. 이 파라미터는 ExistingCognitoUserPoolId 값을 제공하는 경우에만 제공할 수 있습니다.

6. 다음을 선택합니다.
7. 스택 옵션 구성 페이지에서 다음을 선택합니다.
8. 검토 및 생성 페이지에서 설정을 검토하고 확인합니다. 템플릿이 AWS Identity and Access Management(IAM) 리소스를 생성할 것임을 확인하는 상자를 선택합니다.
9. 제출을 선택하여 스택을 배포합니다.

AWS CloudFormation 콘솔의 상태 열에서 스택의 상태를 볼 수 있습니다. 약 10분 후에 CREATE_COMPLETE 상태를 받게 됩니다.

2단계: 사용 사례 배포

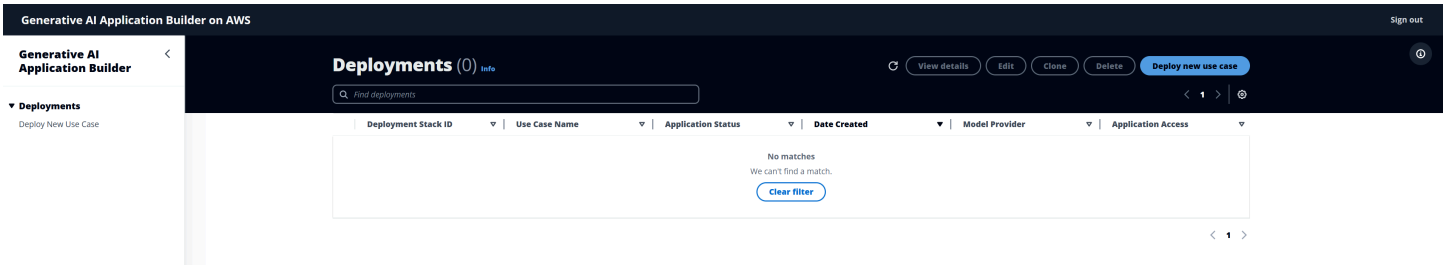
> **Important**
>
> 스택이 성공적으로 배포되면 구성된 관리자 이메일로 가입 이메일이 전송됩니다. 이러한 자격 증명을 사용하여 관리자는 배포 대시보드에 로그인하여 웹 애플리케이션을 사용할 수 있습니다.

> **Note**
>
> AWS 관리 콘솔에 액세스할 수 있는 DevOps 사용자는 스택이 완료될 때 배포 대시보드 UI의 CloudFront URL을 관리자 사용자에게 제공해야 합니다. URL은 CloudFormation 스택의 출력 탭에서 찾을 수 있습니다.

1. 배포 대시보드에 관리자 사용자로 로그인합니다.
2. 애플리케이션 랜딩 페이지에서 새 사용 사례 배포를 선택합니다.

   그러면 사용 사례 구축을 안내하는 배포 마법사가 시작됩니다.

배포 대시보드 랜딩 페이지 설명 - 새 배포



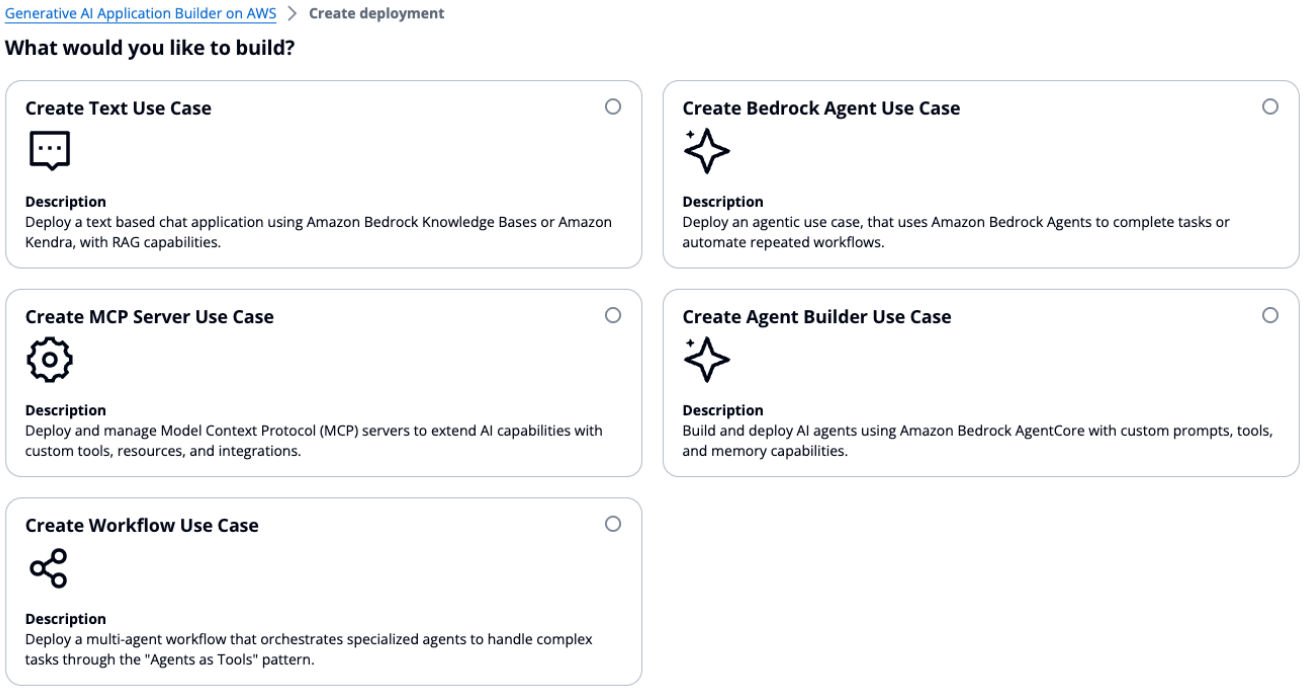
> **Note**
>
> 배포에 사용자를 추가해야 하는 경우 자세한 내용은 Cognito 사용자 풀 관리를 참조하세요.

3단계: 배포 대시보드 마법사를 사용하여 사용 사례 배포

배포 대시보드 마법사에서 다음 중 하나를 선택해야 합니다.

- 텍스트 사용 사례 - 선택적 RAG 기능을 사용하여 채팅 애플리케이션을 배포합니다.
- Bedrock Agent 사용 사례 - Amazon Bedrock Agents를 사용하여 작업을 완료하거나 반복 워크플로를 자동화합니다.
- MCP 서버 - 게이트웨이 또는 런타임 메서드를 사용하여 MCP 서버 배포 및 관리
- 에이전트 빌더 - MCP 통합 및 메모리 관리를 사용하여 AgentCore에서 사용자 지정 에이전트를 빌드하고 배포합니다.
- Workflow Builder - 계층적 위임을 사용하여 여러 Agent Builder 에이전트 오케스트레이션

텍스트 사용 사례 생성, Bedrock 에이전트 사용 사례 생성, MCP 서버 사용 사례 생성, 에이전트 빌더 사용 사례 생성 또는 워크플로 사용 사례 생성의 5가지 옵션을 표시합니다.



3a단계: 텍스트 사용 사례 배포

이 섹션에서는 텍스트 사용 사례를 배포하기 위한 지침을 제공합니다.

사용 사례 선택

텍스트 사용 사례 생성을 선택하면 UI에서 사용 사례 선택 화면이 열립니다. 다음 정보를 제공합니다.

- 사용 사례 이름입니다.
- 사용 사례를 위해 Amazon Cognito 사용자 풀에 추가할 사용 사례의 기본 사용자의 선택적 이메일 주소와 상호 작용할 수 있는 권한이 부여됩니다.
- 이 사용 사례로 UI를 배포할지 여부입니다. 사용 사례와 함께 UI를 배포하지 않으려면 배포된 API 엔드포인트를 애플리케이션에 사용할 수 있습니다.

사용 사례 세부 정보

사용 사례 세부 정보 단계를 사용하면 배포에 대한 추가 설정을 구성할 수 있습니다.

기본적으로 텍스트 사용 사례는 솔루션이 배포 대시보드를 배포할 때 Amazon Cognito 사용자 풀을 생성하고 구성합니다. 이 솔루션은 동일한 사용자 풀에서 새로 생성된 클라이언트를 사용하여 새 사용 사례를 인증합니다. 하지만 사용 사례와 함께 자체 Amazon Cognito 사용자 풀과 클라이언트를 사용하려면 이 단계에서 기존 사용자 풀 ID와 클라이언트 ID를 제공할 수 있습니다.

Important

배포 마법사를 통해 Amazon Cognito 사용자 풀이 생성되면 관리자 사용자는 배포된 모든 사용 사례에 액세스할 수 있습니다. 배포 중에 자체 사용자 풀을 제공하는 경우 관리자에게 배포된 사용 사례에 액세스할 수 있는 권한이 있는지 확인해야 합니다.
또한 Cognito의 앱 클라이언트에서 허용된 콜백 URLs 및 허용된 로그아웃 URLs을 업데이트해야 합니다. 방법:

1. Cognito 콘솔로 이동합니다.
2. 사용자 풀(User Pools)을 선택합니다.
3. 사용자 풀을 선택합니다.
4. 왼쪽 메뉴에서 앱 클라이언트를 선택합니다.
5. 수정할 앱 클라이언트를 선택합니다.
6. 로그인 페이지 탭을 선택합니다.
7. 편집을 선택하고 URLs 추가합니다.
8. 변경 사항 저장을 선택합니다.

또한 사용 사례에 사용자를 더 추가해야 하는 경우 Cognito 사용자 풀 관리 섹션을 참조하세요.

네트워크 구성 선택

이 마법사 단계를 사용하면 기존 또는 새로운 Amazon Virtual Private Cloud(Amazon VPC)를 사용하여 사용 사례를 배포할 수 있습니다. 기존 VPC를 선택하는 경우이 VPC에 사용할 VPC ID, 최대 16개의 서브넷 ID 및 최대 5개의 보안 그룹 IDs를 제공해야 합니다. 기존 VPC를 사용하지 않는 경우 이러한 설정이 자동으로 구성됩니다.

모델 선택

모델 선택 단계의 드롭다운 메뉴에서 모델 공급자를 선택할 수 있습니다. Bedrock과 SageMaker의 두 가지 옵션이 있습니다.

SageMaker를 선택하면 SageMaker AI 콘솔에서 SageMaker AI 모델 엔드포인트를 생성하고 모델이 예상하는 입력 스키마를 제공하고 LLM 응답에 JSONPath를 출력할 수 있습니다. Amazon SageMaker AI를 LLM 공급자로 사용 섹션과 솔루션의 GitHub 리포지토리에 제공된 SageMaker AI 페이로드 예제를 참조할 수 있습니다.

Amazon Bedrock을 선택하면 네 가지 옵션이 표시됩니다.

- 퀵 스타트 모델 - 가격/성능 특성이 서로 다른 모델 모음을 빠르게 시작합니다. 첫 번째 앱을 빌드하는 데 권장됩니다. 이 옵션을 사용하면 제공된 목록에서 모델 이름을 선택할 수 있습니다.
- 기타 파운데이션 모델 - 다양한 기능과 전문화를 갖춘 전체 파운데이션 모델에 액세스할 수 있습니다. 이 옵션을 사용하면 원하는 Bedrock 온디맨드 파운데이션 모델의 모델 ID를 입력할 수 있습니다.
- 추론 프로필 - 추론 프로필은 Bedrock의 교차 리전 추론을 활용하여 최대 사용률 버스트 중에 여러 AWS 리전에 요청을 라우팅하여 처리량을 늘리고 복원력을 개선합니다. 이 옵션을 사용하면 사용하려는 추론 프로파일의 ID를 입력할 수 있습니다.
- 프로비저닝된 모델 - 일관된 성능이 필요한 프로덕션 워크로드를 위한 전용 처리량 용량입니다. 이 옵션을 사용하면 Amazon Bedrock에서 사용할 프로비저닝된/사용자 지정 모델의 ARN을 입력할 수 있습니다.

모델 선택 단계에서는 고급 모델 설정을 선택할 수도 있습니다. Amazon Bedrock 가드레일 구성, Amazon Bedrock의 프로비저닝된 처리량 및 추가 모델 파라미터에 대한 자세한 내용은 고급 LLM 설정을 참조하세요.

교차 리전 추론

리전 간 추론을 통해 Amazon Bedrock 사용자는 다양한 AWS 리전에서 컴퓨팅을 사용하여 계획되지 않은 트래픽 버스트를 원활하게 관리할 수 있습니다. 교차 리전 추론을 사용하려면 추론 프로파일이 필

요합니다. 추론 프로파일은 구성된 AWS 리전 세트의 온디맨드 리소스 풀을 추상화한 것입니다. 소스 리전에서 시작된 추론 요청을 해당 풀에 구성된 다른 리전으로 라우팅할 수 있습니다. 이를 통해 여러 AWS 리전에 트래픽을 분산할 수 있습니다. 이렇게 하면 수요가 가장 많은 기간 동안 처리량을 높이고 복원력을 높일 수 있습니다.

추론 프로파일은 지원하는 모델 및 리전의 이름을 따서 명명됩니다. 포함된 리전 중 하나에서 추론 프로파일을 호출해야 합니다. 예를 들어 다음 표와 같이 추론 프로파일 ID를 `us.anthropic.claude-3-haiku-20240307-v1:0` 사용하면 선택한 모델의 `us-east-1` 및 `us-west-2` 리전을 통해 트래픽을 배포할 수 있습니다. 특정 모델은 특정 리전의 추론 프로파일에서만 사용할 수 있습니다.

추론 프로파일	추론 프로파일 ID	포함된 리전
US Anthropic Claude 3 Haiku	`us.anthropic.claude-3-haiku-20240307-v1:0`	미국 동부 (버지니아 북부) (`us-east-1`) 미국 서부 (오리건) (`us-west-2`)

모델 ID 대신 추론 프로필 ID를 사용하려면 적절한 추론 프로필 ID를 식별해야 합니다. 자세한 내용은 Amazon Bedrock 사용 설명서의 추론 프로파일에 대해 지원되는 리전 및 모델을 참조하세요. Amazon Bedrock 콘솔의 왼쪽 탐색 메뉴에 있는 교차 리전 추론 옵션은 이러한 추론 프로필 IDs 제공합니다.

사용할 추론 프로필 ID를 식별한 후 다음 단계를 수행하여 모델 선택 단계에서 사용할 수 있습니다.

1. 모델 공급자로 Amazon Bedrock을 선택합니다.
2. 추론 프로필 라디오 버튼 옵션을 선택합니다.
3. 나타나는 텍스트 상자에 추론 프로필 ID를 입력합니다.

추론 프로파일에 대한 자세한 내용은 Amazon Bedrock 사용 설명서의 리전 간 추론을 통한 복원력 개선을 참조하세요.

지식 기반 선택

비 검색 증강 생성(RAG) 사용 사례를 배포하려는 경우이 단계를 건너뛸 수 있습니다.

그러나 배포의 일부로 RAG를 활성화하려는 경우 이제 사전 구성된 Amazon Kendra 인덱스 ID 또는 Amazon Bedrock 지식 기반 ID를 제공할 수 있습니다. 솔루션에 사용할 새 Amazon Kendra 인덱스를

생성할 수도 있습니다. 이 솔루션은 현재 Amazon Kendra 및 Amazon Bedrock 지식 기반을 RAG 기반 사용 사례 배포의 지식 기반으로 지원합니다.

RAG 기반 배포에 사용할 수 있도록 지식 기반에 데이터를 수집하는 방법에 대한 지침은 지식 기반 구성 섹션을 참조하세요.

고급 RAG 구성

마법사를 사용하면 쿼리가 지식 기반에 전송될 때마다 검색할 문서 수, 지식 기반에서 문서를 찾을 수 없을 때 LLM의 정적 텍스트 응답, 안전 확인을 위해 LLM 응답과 함께 문서 소스를 표시할지 여부 등 RAG 배포에 사용할 고급 옵션을 선택할 수 있습니다. Amazon Bedrock 지식 기반과 함께 Amazon OpenSearch Serverless를 사용할 때 역할 기반 액세스 제어(RBAC) 또는 검색 유형 재정의와 같은 Amazon Kendra에 대한 지식 기반별 구성을 추가로 구성할 수도 있습니다. 이러한 고급 설정에 대한 자세한 내용은 고급 지식 기반 설정 섹션을 참조하세요.

> **Note**
>
> 지식 기반은 배포 대시보드 및 사용 사례 스택과 동일한 계정 및 리전에 있어야 합니다.

프롬프트 및 토큰 제한 선택

이 단계에서는 LLM과 함께 사용할 프롬프트를 구성할 수 있습니다. 프롬프트에는 `{input}`, `{history}` 및와 같은 자리 표시자가 필요할 수 있습니다`{context}`. 이러한 자리 표시자는 지식 기반에서 검색된 사용자 입력, 대화 기록 및 정보를 그릴 위치에 대해 LLM에 지시합니다.

- Bedrock 모델 공급자의 경우 비 RAG 사용 사례에 대한 제한이 없는 시스템 프롬프트를 제공해야 합니다. 그러나 Bedrock 모델 공급자에 대한 모호하지 않은 프롬프트에는 및 라는 최소 2개의 자리 표시자가 필요합니다. `{input}` `{history}`
- SageMaker 모델 공급자, 시스템 및 모호하지 않은 프롬프트의 경우 둘 다 및 라는 최소 2개의 자리 표시자가 필요합니다`{input}``{history}`.
- RAG 사용 사례의 경우 각 모델 공급자에 대해 `{context}` 자리 표시자가 추가로 필요합니다.

자세한 내용은 프롬프트 구성을 참조하세요. 프롬프트에 대한 토큰 제한 크기를 선택하는 동안 모델 토큰 제한 관리를 위한 팁 섹션을 참조할 수도 있습니다.

멀티모달 입력 활성화

이 단계를 통해 사용 사례에 맞게 멀티모달 입력 기능을 활성화할 수 있습니다. 활성화하면 사용자는 텍스트 쿼리와 함께 이미지와 문서를 업로드하고 전송할 수 있습니다.

지원되는 파일 유형 및 제약 조건:

- 이미지: 메시지당 최대 20개의 이미지. 각 이미지의 크기는 3.75MB, 높이 및 너비는 8,000픽셀 이하여야 합니다. 지원되는 형식: png, jpeg, gif, webp
- 문서: 메시지당 최대 5개의 문서. 각 문서의 크기는 4.5MB 이하여야 합니다. 지원되는 형식: pdf, csv, doc, docx, xls, xlsx, html, txt, md

멀티모달 입력을 사용하는 방법:

1. 사용 사례 배포 중에 MultimodalEnabled 파라미터 활성화
2. 채팅 인터페이스에서 사용자는 두 가지 방법으로 파일을 업로드할 수 있습니다.
   - 채팅 입력 상자에서 업로드 버튼 클릭 또는
   - 파일을 채팅 인터페이스로 직접 끌어서 놓기
3. 파일은 Amazon S3에 업로드되고 선택한 모델에서 처리됩니다.
4. 업로드된 파일은 48시간 후에 자동으로 삭제됩니다.

파일 상태 추적:

DevOps 사용자는 업로드 시간 및 처리 상태를 포함하는 DynamoDB의 파일 메타데이터를 모니터링할 수 있습니다. 파일은 다음과 같은 상태를 가질 수 있습니다.

- 보류 중 - 파일 업로드가 시작되었지만 아직 완료되지 않았습니다. 미리 서명된 URL이 생성될 때의 초기 상태입니다.
- upload - 파일이 S3에 성공적으로 업로드되었으며 모델에서 처리할 준비가 되었습니다.
- 삭제됨 - 사용자가 파일을 삭제했으므로 더 이상 처리할 수 없습니다.
- 유효하지 않음 - 파일 유효성 검사 실패(예: 파일 유형 불일치 또는 보안 유효성 검사 실패).

업로드되지 않은 보류 상태의 파일은 TTL이 만료되면 자동으로 정리됩니다. 업로드된 상태의 파일만 모델에서 처리할 수 있습니다.

S3 멀티모달 버킷 및 DynamoDB 메타데이터 테이블은 `MultimodalDataMetadataTable`각각 `MultimodalDataBucketName` 및 키를 사용하여 배포 대시보드 출력에서 사용할 수 있습니다.

> **Note**
>
> 일부 모델은 멀티모달 입력을 지원하지 않습니다. 이 기능을 활성화하기 전에 선택한 모델이 이미지 및 문서 처리를 지원하는지 확인합니다. 어떤 모델이 이미지를 입력 양식으로 지원하는지 확인하려면 Amazon Bedrock에서 지원되는 파운데이션 모델을 참조하세요.

> **Important**
>
> 사용자가 업로드한 파일은 48시간 수명 주기 정책을 사용하여 Amazon S3에 저장됩니다. 업로드된 파일에 대한 메타데이터는 대화 기록을 위해 24시간 TTL과 함께 Amazon DynamoDB에 저장됩니다.

검토 및 배포.

이 단계 후에는 선택한 설정을 검토하고 사용 사례 배포를 선택합니다. 그러면 새 사용 사례가 배포되고 배포 대시보드 보기에 표시되어 추가로 관리할 수 있습니다.

3b단계: Bedrock Agent 사용 사례 배포

Bedrock 에이전트 사용 사례는 사용 사례 내에서 Amazon Bedrock 에이전트를 호출하기 위한 강력하고 안전한 메커니즘을 제공합니다. 이 기능을 통해 개발자는 강력한 보안 조치를 유지하면서 다양한 파운데이션 모델, 데이터 소스, 소프트웨어 애플리케이션 및 사용자 대화에서 다단계 작업을 오케스트레이션하고 실행할 수 있는 AI 기반 자율 에이전트의 기능을 원활하게 통합할 수 있습니다.

사전 조건

Amazon Bedrock 에이전트를 생성하기 전에 다음이 있는지 확인합니다.

1. Amazon Bedrock 콘솔에 액세스할 수 있는 AWS의 생성형 AI Application Builder가 배포되는 AWS 계정입니다.
2. Amazon Bedrock Agents를 생성하고 관리하기 위한 적절한 IAM 권한.

Amazon Bedrock 에이전트 생성

에이전트 생성에 대한 자세한 지침은 Amazon Bedrock 사용 설명서의 에이전트 수동 생성 및 구성을 참조하세요. 다음과 같은 옵션을 구성할 수 있습니다.

- 에이전트에 대한 지침(프롬프트)
- 지식 기반 - 사용자의 입력을 기반으로 추가 정보를 조회하는 데 사용됩니다.
- 에이전트가 여러 세션에서 정보를 기억할 수 있도록 허용하는 에이전트의 메모리(최대 30일)

Amazon Bedrock 에이전트를 성공적으로 생성한 후 AWS Bedrock 에이전트의 생성형 AI 애플리케이션 빌더 사용 사례 마법사 흐름으로 진행할 수 있습니다. 이렇게 하려면 배포 대시보드에서 새 사용 사례 배포를 선택하고 Bedrock 에이전트 사용 사례 생성을 선택합니다. 마법사에 따라 다음 단계에 따라 사용 사례를 구성합니다.

사용 사례 선택

이 단계는 앞서 설명한 텍스트 사용 사례와 동일합니다.

네트워크 구성 선택

이 단계는 앞서 설명한 텍스트 사용 사례와 동일합니다.

에이전트 선택

이 단계에서는 생성한 Amazon Bedrock 에이전트의 에이전트 ID와 별칭 ID를 제공해야 합니다.

3c단계: MCP 서버 사용 사례 배포

MCP(모델 컨텍스트 프로토콜) 서버 사용 사례를 사용하면 AI 모델 및 에이전트와 통합할 수 있는 MCP 서버를 배포하고 관리할 수 있습니다. MCP 서버는 AI 애플리케이션에 도구, 리소스 및 기능을 노출하는 표준화된 방법을 제공합니다. 기존 Lambda 함수 및 APIs에서 MCP 서버를 생성하거나 컨테이너 이미지를 사용하여 사용자 지정 MCP 서버를 호스팅할 수 있습니다.

사전 조건

MCP Server 사용 사례를 배포하기 전에 다음이 있는지 확인합니다.

1. AWS의 생성형 AI Application Builder가 배포되는 AWS 계정입니다.
2. Amazon Bedrock AgentCore 리소스를 생성하고 관리하기 위한 적절한 IAM 권한.
3. 선택한 생성 방법에 따라:
   - 게이트웨이 메서드(Lambda/API/MCP 서버)의 경우: Lambda 함수, 해당 스키마 파일이 있는 API 엔드포인트(Lambda의 경우 JSON 형식, APIs 경우 OpenAPI/Smithy) 또는 MCP 서버 URL 엔드포인트

- 런타임 메서드(ECR)의 경우: MCP 서버 구현이 포함된 Amazon ECR로 푸시된 도커 컨테이너 이미지

MCP 서버 생성 방법

이 솔루션은 MCP 서버를 생성하는 두 가지 방법을 지원합니다.

Lambda, API 또는 MCP 서버에서 생성(게이트웨이 메서드)

이 메서드는 기존 Lambda 함수, REST APIs 또는 외부 MCP 서버를 래핑하는 MCP 게이트웨이를 생성하여 MCP 도구로 액세스할 수 있도록 합니다. 게이트웨이는 MCP와 기존 서비스 간의 프로토콜 변환을 처리합니다.

- Lambda 대상: 함수의 입력/출력 형식을 설명하는 함수 ARN과 JSON 스키마 파일을 제공하여 기존 Lambda 함수 통합
- OpenAPI 대상: OpenAPI 사양(JSON 또는 YAML 형식)APIs 사용하여 REST API를 OAuth 2.0 또는 API 키 인증 지원과 통합
- Smithy 대상: Smithy 모델 파일(.smithy 또는 .json 형식)을 사용하여 정의된 APIs 통합
- MCP 서버 대상: URL 엔드포인트를 통해 외부 MCP 서버에 직접 연결하여 새 인프라를 배포하지 않고도 기존 MCP 서버를 통합할 수 있습니다.

단일 MCP 게이트웨이 내에서 여러 대상(최대 10개)을 구성할 수 있으며, 각 대상은 서로 다른 도구 또는 기능을 나타냅니다.

ECR 이미지에서 호스팅(런타임 메서드)

이 방법은 Amazon ECR 이미지에서 컨테이너화된 MCP 서버를 배포합니다. 독립 실행형 서비스로 실행해야 하는 사용자 지정 MCP 서버 구현이 있는 경우이 접근 방식을 사용합니다.

- ECR 이미지 URI 제공(태그를 포함해야 함, 예: `:latest` 또는 `:v1.0.0`)
- 필요에 따라 컨테이너에 구성을 전달하도록 환경 변수 구성
- 컨테이너는 MCP 프로토콜을 구현하고 필요한 엔드포인트를 노출해야 합니다.

MCP 서버 배포

MCP 서버 사용 사례를 배포하려면 배포 대시보드에서 새 사용 사례 배포를 선택하고 MCP 서버 사용 사례 생성을 선택합니다. 마법사에 따라 다음 단계에 따라 사용 사례를 구성합니다.

사용 사례 선택

이 단계는 앞서 설명한 텍스트 사용 사례와 동일합니다.

네트워크 구성 선택

현재 퍼블릭 액세스만 활성화되고 VPC는 네트워크 구성에 지원되지 않습니다.

MCP 서버 생성

이 단계에서는 MCP 서버 배포를 구성합니다.

MCP 서버 생성 방법

다음 두 가지 생성 방법 중에서 선택합니다.

- Lambda, API 또는 MCP 서버에서 생성: 기존 Lambda 함수, API 사양 또는 외부 MCP 서버 엔드포인트에서 MCP 게이트웨이 생성
- ECR 이미지에서 호스팅: 컨테이너 이미지에서 사용자 지정 MCP 서버 배포

> **Note**
>
> 배포 후에는 생성 방법을 변경할 수 없습니다. 메서드를 전환해야 하는 경우 새 MCP Server 사용 사례를 배포해야 합니다.

게이트웨이 구성(Lambda/API/MCP 서버 메서드용)

게이트웨이 메서드를 선택한 경우 하나 이상의 대상을 구성합니다.

1. 대상 이름(필수):이 대상 구성을 식별하기 위한 표시 이름입니다.
2. 대상 설명(선택 사항):이 대상이 수행하는 작업에 대한 간략한 설명
3. 대상 유형: 구성할 대상 유형을 선택합니다.
   - Lambda: AWS Lambda 함수의 경우
   - OpenAPI: OpenAPI 사양이 있는 REST APIs 경우
   - Smithy: Smithy 모델 정의가 있는 APIs 경우
   - MCP 서버: URL 엔드포인트를 통해 외부 MCP 서버에 직접 연결
4. 스키마 파일(필수): 대상을 설명하는 스키마 파일을 업로드합니다.

- Lambda의 경우: 입력/출력 형식을 설명하는 JSON 스키마 파일입니다. Lambda 도구 스키마 생성에 대한 자세한 내용은 Amazon Bedrock AgentCore 개발자 안내서의 Lambda 도구 스키마를 참조하세요.
- OpenAPI의 경우: OpenAPI 사양 파일(JSON 또는 YAML). OpenAPI 스키마 요구 사항에 대한 자세한 내용은 Amazon Bedrock AgentCore 개발자 안내서의 OpenAPI 스키마를 참조하세요.
- Smithy의 경우: Smithy 모델 파일(.smithy 또는 .json). Smithy 대상 빌드에 대한 자세한 내용은 Amazon Bedrock AgentCore 개발자 안내서의 Smithy 대상 빌드를 참조하세요.

5. Lambda 함수 ARN(Lambda 대상에 필요): 통합할 Lambda 함수의 ARN입니다.
6. MCP 서버 URL(MCP 서버 대상에 필요): 연결할 외부 MCP 서버의 URL 엔드포인트입니다. URL은 올바르게 인코딩되어야 하며 MCP 서버는 MCP 프로토콜 버전 2025-06-18에서 도구 기능을 지원해야 합니다. 자세한 내용은 Amazon Bedrock AgentCore 개발자 안내서의 MCP 서버 대상을 참조하세요.
7. 아웃바운드 인증(OpenAPI 대상에 필요): REST API 호출에 대한 인증을 구성합니다.
   - 인증 유형: OAuth 2.0 또는 API 키 선택
   - 아웃바운드 인증 공급자 ARN: Amazon Bedrock AgentCore 토큰 볼트에 있는 자격 증명 공급자의 ARN
   - 추가 구성: 인증 유형에 따라:
     - OAuth 2.0의 경우: 범위 및 사용자 지정 파라미터 구성
     - API 키의 경우: 위치(헤더 또는 쿼리 파라미터), 파라미터 이름 및 선택적 접두사 지정

다른 대상 추가를 선택하여 여러 대상(최대 10개)을 추가할 수 있습니다. 각 대상은 MCP 서버에서 노출되는 별도의 도구 또는 기능을 나타냅니다.

ECR 구성(ECR 이미지 메서드용)

런타임 메서드를 선택한 경우 다음을 제공합니다.

1. ECR 이미지 URI(필수): Amazon ECR에서 도커 이미지의 전체 URI
   - 형식: `account-id.dkr.ecr.region.amazonaws.com/repository-name:tag`
   - 이미지는 배포와 동일한 AWS 리전에 있어야 합니다.
   - 태그가 필요합니다(예: , `:latest:v1.0.0`).
2. 환경 변수(선택 사항): 런타임 시 컨테이너에 전달하도록 키-값 페어 구성
   - 이를 사용하여 구성, 자격 증명 또는 사용자 지정 플래그 제공
   - 최대 10개의 환경 변수를 추가할 수 있습니다.

검토 및 배포.

MCP 서버를 구성한 후 선택한 설정을 검토하고 사용 사례 배포를 선택합니다. 그러면 새 MCP 서버 사용 사례가 배포되고 추가 관리를 위해 배포 대시보드 보기에 표시됩니다.

> **Note**
>
> MCP Server 배포는 게이트웨이, 런타임 및 워크로드 ID를 포함하여 Amazon Bedrock AgentCore에 리소스를 생성합니다. 이러한 리소스는 솔루션에서 자동으로 관리되며 사용 사례를 삭제하면 정리됩니다.

3d단계: Agent Builder 사용 사례 배포

Agent Builder를 사용하면 Amazon Bedrock AgentCore에서 프로덕션 지원 AI 에이전트를 생성, 구성 및 배포할 수 있습니다. 이 기능은 시스템 프롬프트, 모델 선택, MCP 서버 통합 및 메모리 관리를 통해 에이전트 동작을 완벽하게 제어합니다.

배포 프로세스는 주로 텍스트 사용 사례와 동일하며 몇 가지 주목할 만한 차이점이 있습니다.

사용 사례 선택

이 단계는 앞서 설명한 텍스트 사용 사례와 동일합니다.

사용 사례 세부 정보

이 단계는 앞서 설명한 텍스트 사용 사례와 동일합니다.

에이전트 구성

이 단계에서는 시스템 프롬프트, 사용 가능한 MCP 서버/스트랜드 도구 및 메모리를 포함한 코어 에이전트 설정을 구성합니다.

시스템 프롬프트

시스템 프롬프트는 에이전트의 동작, 성격 및 기능을 정의합니다. 다음을 수행할 수 있습니다.

- 기본 시스템 프롬프트 템플릿 편집
- 기본값으로 재설정 버튼을 사용하여 원래 템플릿을 복원합니다.
- 도구 사용 및 응답 형식 지정에 대한 지침 포함

MCP 서버 통합(선택 사항)

에이전트에게 엔터프라이즈 도구 및 데이터에 대한 액세스 권한을 제공하도록 모델 컨텍스트 프로토콜 서버를 구성합니다.

1. 드롭다운에서 사용 가능한 MCP 서버 중에서 선택
2. 에이전트가 액세스할 수 있는 즉시 사용 가능한 도구 검토

> **Note**
>
> 배포 전에 MCP 서버를 구성하고 액세스할 수 있어야 합니다. 서버 설정 지침은 MCP 설명서를 참조하세요.

메모리 구성

에이전트가 컨텍스트와 지식을 유지하는 방법을 구성합니다.

- 단기 메모리: 모든 에이전트에 대해 기본적으로 활성화됩니다. 세션 내에서 대화 컨텍스트를 유지합니다.
- 장기 메모리: 세션 간에 인사이트를 추출하고 저장할 수 있도록 전환합니다. 의미 체계 메모리 전략과 함께 AgentCore 메모리를 사용합니다.

검토 및 배포.

이 단계 후에는 선택한 설정을 검토하고 사용 사례 배포를 선택합니다. Agent Builder 배포는 일반적으로 10~15분 후에 완료됩니다. 그러면 배포 대시보드 보기에 새 사용 사례가 표시되어 추가로 관리할 수 있습니다.

3e단계: 워크플로 사용 사례 배포

Workflow Builder를 사용하면 에이전트를 도구로 위임 패턴을 사용하여 여러 Agent Builder 에이전트를 오케스트레이션하는 감독자 에이전트를 생성할 수 있습니다. 이 기능을 사용하면 기존 Agent Builder 배포를 재사용하여 복잡한 다중 에이전트 워크플로를 구축할 수 있습니다.

배포 프로세스는 에이전트 검색 및 선택을 위한 추가 단계와 함께 Agent Builder와 유사한 패턴을 따릅니다.

사용 사례 선택

이 단계는 앞서 설명한 텍스트 사용 사례와 동일합니다.

사용 사례 세부 정보

이 단계는 앞서 설명한 텍스트 사용 사례와 동일합니다.

감독자 에이전트 구성

이 단계에서는 특수 에이전트 빌더 에이전트를 조정할 감독자 에이전트를 구성합니다.

시스템 프롬프트

시스템 프롬프트는 감독자 에이전트가 특수 에이전트에게 작업을 위임하는 방법을 정의합니다. 다음을 수행할 수 있습니다.

- 기본 시스템 프롬프트 템플릿 편집
- 에이전트 선택 및 위임 지침 포함
- 여러 에이전트의 결과를 집계하는 방법 정의
- 기본값으로 재설정 버튼을 사용하여 원래 템플릿을 복원합니다.

> **Note**
>
> 시스템 프롬프트는 각 특수 에이전트를 사용하는 시기와 방법을 명확하게 설명해야 합니다. 에이전트 설명은 적절한 위임에 매우 중요합니다.

모델 선택

감독자 에이전트의 파운데이션 모델을 선택합니다. 감독자 에이전트는이 모델을 사용하여 다음을 수행합니다.

- 사용자 요청 이해
- 적절한 전문 에이전트 선택
- 에이전트 실행 조정
- 응답 집계 및 형식 지정

특수 에이전트 선택

이 단계에서는 감독자가 작업을 위임할 수 있는 Agent Builder 에이전트를 선택합니다.

에이전트 추가

1. 에이전트 추가를 클릭하여 에이전트 선택 대화 상자를 엽니다.
2. 목록에서 하나 이상의 Agent Builder 에이전트를 선택합니다.
3. 감독자에게 제공될 에이전트 설명 검토
4. 선택 확인

> **Note**
>
> - 워크플로에는 특수 에이전트로서 최소 1개의 Agent Builder 사용 사례가 필요합니다.
> - 워크플로를 생성하기 전에 모든 특수 에이전트를 성공적으로 배포해야 합니다.

검토 및 배포.

다음을 포함한 워크플로 구성을 검토합니다.

- 감독자 에이전트 시스템 프롬프트 및 모델
- 특수 에이전트 목록
- 메모리 설정

사용 사례 배포를 선택합니다. 워크플로 배포는 일반적으로 15~20분 후에 완료됩니다. 추가 관리를 위해 배포 대시보드 보기에 새 워크플로가 표시됩니다.

4단계: 배포 후 구성

이 섹션에서는 배포 후 솔루션을 구성하기 위한 권장 사항을 제공합니다.

Amazon S3 버킷 버전 관리, 수명 주기 정책 및 리전 간 복제

이 솔루션은 생성하는 버킷에 수명 주기 구성을 적용하지 않습니다. 다음과 같이 하는 것이 좋습니다:

- 프로덕션 배포를 위한 수명 주기 구성 설정. 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 버킷에 대한 수명 주기 구성 설정을 참조하세요.
- 솔루션이 배포된 사용 사례에 따라 Amazon S3 버킷에 대한 버전 관리 및 리전 간 복제를 활성화합니다.

Amazon DynamoDB 백업

이 솔루션은 DynamoDB를 여러 용도로 사용합니다(이 솔루션의 AWS 서비스 참조). 솔루션은 생성하는 테이블에 대한 백업을 활성화하지 않습니다. 프로덕션 배포를 위해이 기능의 백업을 생성하는 것이 좋습니다. 자세한 내용은 DynamoDB 테이블 백업 및 DynamoDB용 AWS Backup 사용을 참조하세요.

Amazon CloudWatch 대시보드 및 경보

이 솔루션은 CloudWatch에 사용자 지정 대시보드를 배포하여 사용자 지정 게시 지표 및 AWS 서비스 지표에서 차트를 렌더링합니다. CloudWatch 경보를 생성하고 솔루션이 배포된 사용 사례에 따라 알림을 추가하는 것이 좋습니다.

Amazon CloudWatch Logs

Lambda 로그는 만료되지 않도록 구성되며 API Gateway 로그는 10년 만료로 구성됩니다. 엔터프라이즈의 레코드 보존 정책에 맞게 각 로그 그룹의 만료를 업데이트할 수 있습니다.

TLS v1.2 이상의 인증서가 있는 사용자 지정 웹 도메인

솔루션은 CloudFront를 사용하여 웹 UI 및 Edge Optimized API Gateway를 배포합니다. CloudFront의 도메인은 TLS v1.2 이상의 인증서를 적용하지 않습니다. Amazon Route 53을 사용하여 사용자 지정 도메인을 생성하거나, AWS Certificate Manager를 사용하여 인증서를 생성하거나, 조직에 기존 인증서가 있는 경우 기존 인증서를 사용하는 것이 좋습니다.

자세한 내용은 Amazon Route 53 개발자 안내서 및 API Gateway의 사용자 지정 도메인에 대한 최소 TLS 버전 선택을 참조하세요.

Amazon Kendra를 사용한 확장

이 솔루션은 Amazon Kendra를 사용하여 수집된 문서에서 NLP 기반 지능형 검색을 수행하는 기능을 제공합니다. 대규모 워크로드에 대해 다음 CloudFormation 파라미터를 사용하여 Amazon Kendra의 용량을 늘릴 수 있습니다.

파라미터	기본값	설명
Amazon Kendra 추가 쿼리 용량	`0`	인덱스 및 GetQuerySuggestions 용량에 대한 추가 쿼리 용량입니다. 인덱스에 대

파라미터	기본값	설명
		한 추가 용량 단위는 하루에 약 8,000개의 쿼리를 제공합니다.
Amazon Kendra 추가 스토리지 용량	0	인덱스에 대한 추가 스토리지 용량입니다. 단일 용량 단위는 30GB의 스토리지 공간 또는 100,000개의 문서 중 먼저 도달하는 쪽을 제공합니다.
Amazon Kendra 에디션	Developer	Amazon Kendra는 인덱스를 생성할 수 있는 개발자 및 엔터프라이즈 에디션을 제공합니다. Amazon Kendra Edition의 차이점에 대한 자세한 내용은 Amazon Kendra 요금을 참조하세요.

이러한 CloudFormation 파라미터의 값을 수정하려면 스택 배포 시 적절한 값을 선택합니다. 쿼리 및 스토리지 용량 단위에 대한 자세한 내용은 용량 조정을 참조하세요.

> **Note**
>
> 텍스트 사용 사례가 RAG가 활성화된 상태로 배포되지 않은 경우 Amazon Kendra 인덱스가 사용되거나 생성되지 않습니다.

Idp 페더레이션을 사용하여 SSO 설정

이 솔루션을 사용하면 SAML 또는 OIDC 기반 자격 증명 페더레이션을 지원하는 외부 자격 증명 공급자와 통합할 수 있습니다. 솔루션이 배포되면 배포 대시보드 및 개별 사용 사례에 대한 Amazon Cognito 사용자 풀과 개별 앱 클라이언트 통합이 생성됩니다. 외부 Idp를 기반으로 Amazon Cognito 개발자 안내서의 사용자 풀에 대한 자격 증명 공급자 구성 섹션에 제공된 단계에 따라 SSO를 설정하려는 배포 대시보드 또는 사용 사례에 대한 앱 클라이언트 통합을 선택합니다.

사용자 그룹 정보를 RAG 기반 아키텍처의 지식 기반 또는 벡터 스토어에 전달하려면 외부 Idp의 사용자 그룹을 Amazon Cognito 사용자 그룹으로 매핑해야 합니다. 솔루션은 사전 토큰 생성 단계와 매핑할

초기 스캐폴딩 Lambda 함수 트리거를 제공합니다. Lambda 함수에는 그룹 매핑을 제공하도록 업데이트해야 하는 group_mapping.json 파일이 있습니다. Amazon Cognito에서 지원하는 Lambda 트리거에 대한 Lambda 트리거를 사용하여 사용자 풀 워크플로 사용자 지정을 참조하세요.

수동 사용자 풀 구성

배포 중에 관리자 또는 기본 사용자 이메일을 전달하지 않도록 선택한 경우 올바른 권한을 보장하려면 Amazon Cognito에서 적절한 사용자 그룹을 수동으로 생성해야 합니다.

1. 배포 대시보드의 경우 Cognito 사용자 풀Admin에 라는 그룹을 생성합니다.
2. 각 사용 사례에 대해 Cognito 사용자 풀${UseCaseName}-Users에 라는 그룹을 생성합니다. 여기서 ${UseCaseName}는 배포된 사용 사례의 이름입니다.

이러한 그룹은 권한 부여 메커니즘이 올바르게 작동하는 데 필요합니다. 액세스 권한을 부여하려는 모든 사용자를 적절한 그룹에 추가해야 합니다.

placeholder@example.com이 전달되면 Cognito 그룹이 생성되지만 연결된 사용자를 생성하고 그룹에 할당해야 합니다.

로그인 화면 사용자 지정

이 솔루션은 Amazon Cognito 호스팅 UI를 사용하여 로그인 페이지를 렌더링합니다. 기본 제공 로그인 페이지를 사용자 지정하려면 Amazon Cognito 개발자 안내서의 기본 제공 로그인 및 가입 웹 페이지 사용자 지정을 참조하세요.

추가 보안 고려 사항

솔루션을 배포하는 사용 사례에 따라 다음 보안 권장 사항을 검토합니다.

- 고객 관리형 AWS KMS 암호화 키 - 솔루션은 AWS 관리형 AWS KMS 키를 추가 비용 없이 사용할 수 있으므로 기본적으로 이를 사용합니다. 사용 사례를 검토하여 고객 관리형 AWS KMS 키를 사용하도록 솔루션을 업데이트해야 하는지 확인합니다.
- API Gateway 제한 규칙 - 솔루션은 API Gateway에 기본 제한 규칙을 사용하여 배포됩니다. 사용 사례와 예상 트랜잭션 볼륨에 따라 APIs에 대한 제한을 구성하는 것이 좋습니다. 자세한 내용은 Amazon API Gateway API Gateway 개발자 안내서의 처리량 향상을 위한 API 요청 제한을 참조하세요.

- AWS CloudTrail 활성화 - AWS 계정에서 API 호출을 로깅하기 위해 솔루션이 배포된 AWS 계정에서 AWS CloudTrail을 활성화하는 것이 좋습니다. 자세한 내용은 AWS CloudTrail 사용 설명서를 참조하세요.
- 드리프트 감지 - 배포된 솔루션 스택에 대한 의도하지 않거나 악의적인 변경 사항을 식별하고 알림을 받도록 CloudFormation 스택에서 드리프트 감지를 구성하는 것이 좋습니다. 자세한 내용은 AWS CloudFormation 스택에서 드리프트를 자동으로 감지하는 경보 구현을 참조하세요.
- Cognito JSON 웹 토큰(JWTs) -이 솔루션은 Amazon Cognito에서 발급한 JWTs 사용하여 REST API 엔드포인트로 인증합니다. ID 토큰 및 액세스 토큰의 만료 시간이 5분으로 솔루션을 구성했습니다. 사용자가 로그아웃하면 새 토큰을 생성하는 기능이 취소됩니다(새 토큰이 취소됨). 그러나 현재 토큰이 만료될 때까지 API 엔드포인트에 대한 모든 요청은 유효한 토큰이 있으므로 성공적으로 인증됩니다. 사용 사례에 대한 보안 고려 사항을 검토하고 토큰 유효 기간을 조정합니다.

수명 주기 정책 사용자 지정:

프로덕션 배포의 경우 보존 요구 사항에 따라 수명 주기 정책을 검토하고 조정합니다. Amazon Simple Storage Service 사용 설명서의 버킷에 대한 수명 주기 구성 설정을 참조하세요.

멀티모달 파일 스토리지 및 수명 주기

사용 사례에 대해 멀티모달 입력 기능(MultimodalEnabled를 로 설정Yes)을 활성화한 경우 솔루션은 업로드된 파일을 저장할 Amazon S3 버킷과 파일 메타데이터를 추적할 DynamoDB 테이블을 생성합니다.

기본 수명 주기 정책:

- S3 파일: 48시간 후 자동으로 삭제됨
- DynamoDB 메타데이터: 레코드는 24시간 후에 만료됩니다(대화 기록 TTL).

보안 고려 사항:

- 파일은 사용 사례 ID, 사용자 ID, 대화 ID 및 메시지 ID로 분할되며 대신 UUID 이름으로 저장됩니다. UUID와 파일 이름의 매핑은 DynamoDB 메타데이터 테이블에서 사용할 수 있습니다.
- 사용자는 자신의 대화 내에서 업로드한 파일에만 액세스할 수 있습니다.
- 매직 번호 감지를 사용하여 파일 유형 검증 수행
- 업로드된 파일에 악성 콘텐츠가 있는지 스캔하려면 Amazon GuardDuty Malware Protection for S3를 활성화하는 것이 좋습니다.

독립 실행형 텍스트 사용 사례 배포

이 섹션의 단계별 지침에 따라 솔루션을 구성하고 계정에 배포합니다.

배포 시간: 약 10~30분

1. AWS Management Console에 로그인하고 버튼을 선택하여 배포하려는 CloudFront 템플릿을 시작합니다.

BedrockChat.template	
SageMakerChat.template	

2. 이 템플릿은 기본적으로 미국 동부(버지니아 북부) 리전에서 시작됩니다. 다른 AWS 리전에서 솔루션을 실행하려면 콘솔 탐색 표시줄의 리전 선택기를 사용합니다.

   참고:이 솔루션은 현재 일부 AWS 리전에서 사용할 수 없는 Amazon Kendra 및 Amazon Bedrock을 사용합니다. 이러한 기능을 사용하는 경우 이러한 서비스를 사용할 수 있는 AWS 리전에서이 솔루션을 시작해야 합니다. 리전별 최신 가용성은 AWS 리전 서비스 목록을 참조하세요.
3. 스택 생성 * 페이지에서 *Amazon S3 URL *텍스트 상자에 올바른 템플릿 URL이 있는지 확인하고 * 다음을 선택합니다.
4. *스택 세부 정보 지정 * 페이지에서 솔루션 스택에 이름을 할당합니다. 문자 제한 이름 지정에 대한 자세한 내용은 AWS Identity and Access Management 사용 설명서의 IAM 및 STS 제한을 참조하세요.
5. 파라미터에서 이 솔루션 템플릿의 파라미터를 검토하고 필요에 따라 수정합니다. 이 솔루션은 다음과 같은 기본값을 사용합니다.

UseCaseUUID	*<_input_>*	애플리케이션 내에 배포된이 사용 사례를 식별하기 위한 36자 길이의 UUIDv4입니다.
UseCaseConfigRecordKey	*<_input_>*	런타임 시 채팅 공급자 Lambda에 필요한 구성을 포함하는 레코드에 해당하는 키입니다. 테이블의 레코드에는

		이 값과 일치하는 키 속성과 원하는 구성을 포함하는 구성 속성이 있어야 합니다. 이 레코드는 사용 중인 경우 배포 플랫폼에 의해 채워집니다. 이 사용 사례의 독립 실행형 배포의 경우 UseCaseConfigTableName에 정의된 테이블에 수동으로 생성된 항목이 필요합니다.
UseCaseConfigTableName	*<_Requires input_>*	스택은 UseCaseConfigRecordKey 키에서이 이름을 사용하여 테이블에서 구성을 읽습니다.

ExistingRestApiId	(선택 사항 입력)	사용할 기존 API Gateway REST API ID입니다. 제공되지 않으면 새 API Gateway REST API가 생성됩니다. 일반적으로 배포 대시보드에서 배포할 때 제공됩니다. 참고: 기존 APIs 사용하면 여러 독립 실행형 사용 사례를 배포해야 할 때 리소스 중복을 줄이고 APIs 관리를 간소화할 수 있습니다. 독립 실행형 사용 사례에 기존 APIs를 제공할 때는 API가 예상 모델과 함께 필요한 경로(들)로 구성되어 있는지 확인해야 합니다. 사전 구성된 필수 /details 경로(채팅 중에 사용 사례 세부 정보 가져오기)와 선택적으로 /feedback 경로(LLM 채팅 응답에 대한 피드백 수집을 활성화Yes하기 위해 FeedbackEnabled가 로 설정된 경우)를 구성해야 합니다. 또한 ExistingApiRootResourceId, ExistingCognitoUserPoolId 및 ExistingCognitoGroupPolicyTableName도 제공해야 합니다.

ExistingApiRootResourceId	(선택 사항 입력)	사용할 기존 API Gateway REST API 루트 리소스 ID입니다. REST API 루트 리소스 ID는 API의 "리소스" 섹션에서 루트 리소스(/)를 선택하여 AWS 콘솔에서 가져올 수 있습니다. 그러면 리소스 ID가 리소스 세부 정보 패널에 표시됩니다. 또는 REST API에서 설명 API 호출을 실행하여 루트 리소스 ID를 찾을 수 있습니다.
FeedbackEnabled	No	아니요로 설정하면 배포된 사용 사례 스택은 피드백 기능에 액세스할 수 없습니다.
ExistingModelInfoTableName	(선택 사항 입력)	모델 정보 및 기본값을 포함하는 테이블의 DynamoDB 테이블 이름입니다. 배포 플랫폼에서 사용됩니다. 생략하면 모델 기본값을 저장할 새 테이블이 생성됩니다.
DefaultUserEmail	`placeholder@example.com`	이 사용 사례에 대한 기본 사용자의 이메일입니다. 이 이메일의 Amazon Cognito 사용자가 생성되어 사용 사례에 액세스합니다. 제공되지 않으면 Cognito 그룹 및 사용자가 생성되지 않습니다. `placeholder@example.com` 를 사용하여 그룹을 생성할 수도 있지만 사용자는 생성할 수 없습니다. 사용자 풀 설정에 대한 자세한 내용은 수동 사용자 풀 구성을 참조하세요.

ExistingCognitoUserPoolId	(선택 사항 입력)	이 사용 사례가 인증될 기존 Amazon Cognito 사용자 풀의 UserPoolId입니다. 일반적으로 배포 대시보드에서 배포할 때 제공되지만이 사용 사례 스택을 독립적으로 배포할 때는 생략할 수 있습니다.
CognitoDomainPrefix	(선택 사항 입력)	Cognito 사용자 풀 클라이언트에 도메인을 제공하려면 값을 입력합니다. 값을 제공하지 않으면 배포가 값을 생성합니다.
ExistingCognitoUserPoolClient	(선택 사항 입력)	기존 사용자 풀 클라이언트(앱 클라이언트)를 사용할 사용자 풀 클라이언트를 제공합니다. 사용자 풀 클라이언트를 제공하지 않으면 새 클라이언트가 생성됩니다. 이 파라미터는 기존 사용자 풀 ID가 제공된 경우에만 제공할 수 있습니다.
ExistingCognitoGroupPolicyTableName	(선택 사항 입력)	사용자 그룹 정책이 포함된 DynamoDB 테이블의 이름입니다. 이는 사용 사례의 API에서 사용자 지정 권한 부여자가 사용합니다. 일반적으로 배포 플랫폼에서 배포할 때 입력을 제공할 수 있지만이 사용 사례 스택을 독립적으로 배포할 때는 생략할 수 있습니다.

RAGEnabled	`true`	true로 설정하면 배포된 사용 사례 스택은 RAG 기능을 제공하기 위해 생성된 제공된 Amazon Kendra 인덱스를 사용합니다. 로 설정하면 `false`사용자가 LLM과 직접 상호 작용합니다.
KnowledgeBaseType	Bedrock	RAG에 사용할 지식 기반 유형입니다. RAGEnabled가 인 경우에만 설정합니다`true`. Bedrock 또는 Kendra일 수 있습니다. 참고: RAGEnabled가 true인 경우에만 관련이 있습니다.
ExistingKendraIndexId	(선택 사항 입력)	사용 사례에 사용할 기존 Kendra 인덱스의 인덱스 ID입니다. 제공되지 않고 KnowledgeBaseType이 Kendra인 경우 새 인덱스가 생성됩니다. 참고: RAGEnabled가 `true` 이고 KnowledgeBaseType이 인 경우에만 관련이 있습니다`Kendra`.

NewKendraIndexName	(선택 사항 입력)	이 사용 사례에 대해 생성할 새 Kendra 인덱스의 이름입니다. ExistingKendraIndexId가 제공되지 않은 경우에만 적용됩니다. 참고: RAGEnabled가 true이고 KnowledgeBaseType이 Kendra인 경우에만 관련이 있습니다.
NewKendraQueryCapacityUnits	0	이 사용 사례에 대해 생성할 새 Amazon Kendra 인덱스의 추가 쿼리 용량 단위입니다. ExistingKendraIndexId가 제공되지 않은 경우에만 적용됩니다. CapacityUnitsConfiguration을 참조하세요. 참고: RAGEnabled가 `true`이고 KnowledgeBaseType이인 경우에만 관련이 있습니다`Kendra`.
NewKendraStorageCapacityUnits	`0`	이 사용 사례에 대해 생성할 새 Amazon Kendra 인덱스의 추가 스토리지 용량 단위입니다. ExistingKendraIndexId가 제공되지 않은 경우에만 적용됩니다. CapacityUnitsConfiguration을 참조하세요. 참고: RAGEnabled가 `true`이고 KnowledgeBaseType이인 경우에만 관련이 있습니다`Kendra`.

NewKendraIndexEdition	(선택 사항 입력)	이 사용 사례에 대해 생성할 새 Amazon Kendra 인덱스에 사용할 Amazon Kendra 에디션입니다. ExistingKendraIndexId가 제공되지 않은 경우에만 적용됩니다. Amazon Kendra Editions를 참조하세요. 참고: RAGEnabled가 `true`이고 KnowledgeBaseType이인 경우에만 관련이 있습니다`Kendra`.
BedrockKnowledgeBaseId	(선택 사항 입력)	RAG 사용 사례에 사용할 Bedrock 지식 기반의 ID입니다. ExistingKendraIndexId 또는 NewKendraIndexName이 제공된 경우 제공할 수 없습니다. 참고: RAGEnabled가 `true`이고 KnowledgeBaseType이인 경우에만 관련이 있습니다`Bedrock`.
VpcEnabled	No	스택 리소스를 VPC 내에 배포해야 하는지 여부.
CreateNewVpc	No	솔루션이 새 VPC를 생성하고이 사용 사례에 사용되도록 `Yes`하려면를 선택합니다. 참고: VpcEnabled가 인 경우에만 관련이 있습니다`Yes`.

IPAMPoolId	(선택 사항 입력)	Amazon VPC IP 주소 관리자를 사용하여 CIDR 범위를 할당하려면 사용할 IPAM 풀 ID를 제공합니다. 참고: VpcEnabled가 Yes 이고 CreateNewVpc가 인 경우에만 관련이 있습니다No.
ExistingVpcId	(선택 사항 입력)	사용 사례에 사용할 기존 VPC의 VPC ID입니다. 참고: VpcEnabled가 Yes 이고 CreateNewVpc가 인 경우에만 관련이 있습니다No.
ExistingPrivateSubnetIds	(선택 사항 입력)	Lambda 함수를 배포하는 데 사용할 기존 프라이빗 서브넷 IDs를 쉼표로 구분한 목록입니다. 참고: VpcEnabled가 Yes 이고 CreateNewVpc가 인 경우에만 관련이 있습니다No.
ExistingSecurityGroupIds	(선택 사항 입력)	Lambda 함수를 구성하는 데 사용할 기존 VPC의 쉼표로 구분된 보안 그룹 목록입니다. 참고: VpcEnabled가 Yes 이고 CreateNewVpc가 인 경우에만 관련이 있습니다No.
VpcAzs	(선택 사항 입력)	VPCs의 서브넷이 생성되는 AZs의 쉼표로 구분된 목록 참고: VpcEnabled가 Yes 이고 CreateNewVpc가 인 경우에만 관련이 있습니다No.

UseInferenceProfile	No	구성된 모델이 Bedrock인 경우 Bedrock 추론 프로파일을 사용하고 있는지 여부를 표시할 수 있습니다. 이렇게 하면 스택 배포 중에 필요한 IAM 정책을 구성할 수 있습니다. 자세한 내용은 다음 https://docs.aws.amazon.com/bedrock/latest/userguide/cross-region-inference.html 참조하십시오.
DeployUI	예	이 배포에 대한 프런트엔드 UI를 배포하는 옵션을 선택합니다. 아니요를 선택하면는 APIs를 호스팅하기 위한 인프라, API에 대한 인증 APIs 및 백엔드 처리만 생성합니다.

6. 다음을 선택합니다.
7. 스택 옵션 구성 페이지에서 다음을 선택합니다.
8. 검토 페이지에서 설정을 검토하고 확인합니다. 템플릿이 AWS Identity and Access Management(IAM) 리소스를 생성할 것임을 확인하는 상자를 선택합니다.
9. 스택 생성을 선택하여 스택을 배포합니다.

AWS CloudFormation 콘솔의 상태 열에서 스택의 상태를 볼 수 있습니다. 약 10~30분 후에 CREATE_COMPLETE 상태를 받게 됩니다.

독립 실행형 Bedrock Agent 사용 사례 배포

이 섹션의 단계별 지침에 따라 솔루션을 구성하고 계정에 배포합니다.

배포 시간: 약 10~30분

1. AWS Management Console에 로그인하고 버튼을 선택하여 CloudFront 템플릿을 시작합니다.

BedrockAgent.template	Launch solution

2. 이 템플릿은 기본적으로 미국 동부(버지니아 북부) 리전에서 시작됩니다. 다른 AWS 리전에서 솔루션을 실행하려면 콘솔 탐색 표시줄의 리전 선택기를 사용합니다.

> **Note**
>
> 이 솔루션은 현재 일부 AWS 리전에서 사용할 수 없는 Amazon Bedrock을 사용합니다. 이러한 기능을 사용하는 경우 이러한 서비스를 사용할 수 있는 AWS 리전에서이 솔루션을 시작해야 합니다. 리전별 최신 가용성은 AWS 리전 서비스 목록을 참조하세요.

3. 스택 생성 페이지에서 Amazon S3 URL 텍스트 상자에 올바른 템플릿 URL이 있는지 확인하고 다음을 선택합니다.
4. 스택 세부 정보 지정 페이지에서 솔루션 스택 이름을 할당합니다. 문자 제한 이름 지정에 대한 자세한 내용은 AWS Identity and Access Management 사용 설명서의 {https---docs-aws-amazon-com-https---docs-aws-amazon-com-IAM-latest-UserGuide-reference-iam-limits-html}[IAM 및 AWS STS 할당량]을 참조하세요.
5. 파라미터에서 이 솔루션 템플릿의 파라미터를 검토하고 필요에 따라 수정합니다. 이 솔루션은 다음과 같은 기본값을 사용합니다.

파라미터	기본 항목	설명
UseCaseUUID	*<_Requires input_>*	애플리케이션 내에 배포된이 사용 사례를 식별하기 위한 36자 길이의 UUIDv4입니다.
UseCaseConfigRecordKey	*<## ##>*	런타임 시 채팅 공급자 Lambda 함수에 필요한 구성을 포함하는 레코드에 해당하는 키입니다. 테이블의 레코드에는이 값과 일치하는 키 속성과 원하는 구성을 포함하는 구성 속성이 있어야 합니다.

파라미터	기본 항목	설명
		이 레코드는 사용 중인 경우 배포 플랫폼에 의해 채워집니다. 이 사용 사례의 독립 실행형 배포의 경우 UseCaseConfigTableName에 정의된 테이블에 수동으로 생성된 항목이 필요합니다.
UseCaseConfigTableName	`<## ##>`	스택은 여기에 제공된 테이블에서 UseCaseConfigRecordKey에 정의된 레코드 키를 사용하여 사용 사례 구성을 읽습니다.
DefaultUserEmail	`placeholder@example.com`	이 사용 사례에 대한 기본 사용자의 이메일입니다. 이 솔루션은이 이메일에 대한 Amazon Cognito 사용자를 생성하여 사용 사례에 액세스합니다.

파라미터	기본 항목	설명
ExistingRestApiId	(선택 사항 입력)	사용할 기존 API Gateway REST API ID입니다. 제공되지 않으면 새 API Gateway REST API가 생성됩니다. 일반적으로 배포 대시보드에서 배포할 때 제공됩니다. 참고: 기존 APIs 사용하면 여러 독립 실행형 사용 사례를 배포해야 할 때 리소스 중복을 줄이고 APIs 관리를 간소화할 수 있습니다. 독립 실행형 사용 사례에 기존 APIs를 제공할 때는 API가 예상 모델과 함께 필요한 경로(들)로 구성되어 있는지 확인해야 합니다. 필요한 사전 구성된 /details 경로(채팅 중에 사용 사례 세부 정보 가져오기)와 선택적으로 /feedback 경로(LLM 채팅 응답에 대한 피드백 수집을 활성화Yes하기 위해 FeedbackEnabled가 로 설정된 경우)를 구성해야 합니다. 또한 ExistingApiRootResourceId, ExistingCognitoUserPoolId 및 ExistingCognitoGroupPolicyTableName도 제공해야 합니다.

파라미터	기본 항목	설명
ExistingApiRootResourceId	(선택 사항 입력)	사용할 기존 API Gateway REST API 루트 리소스 ID입니다. REST API 루트 리소스 ID는 API의 "리소스" 섹션에서 루트 리소스(/)를 선택하여 AWS 콘솔에서 가져올 수 있습니다. 그러면 리소스 ID가 리소스 세부 정보 패널에 표시됩니다. 또는 REST API에서 설명 API 호출을 실행하여 루트 리소스 ID를 찾을 수 있습니다.
FeedbackEnabled	No	아니요로 설정하면 배포된 사용 사례 스택은 피드백 기능에 액세스할 수 없습니다.
CognitoDomainPrefix	(선택 사항 입력)	Amazon Cognito 사용자 풀 클라이언트에 도메인을 제공하려면 값을 입력합니다. 값을 제공하지 않으면 솔루션이 값을 생성합니다.
ExistingCognitoUserPoolId	(선택 사항 입력)	이 사용 사례를 인증하려는 기존 Amazon Cognito 사용자 풀의 UserPoolId입니다. 참고: 일반적으로 배포 대시보드에서 배포할 때이 ID를 제공하지만이 사용 사례 스택을 독립적으로 배포할 때는 생략할 수 있습니다.

파라미터	기본 항목	설명
ExistingCognitoUserPoolClient	(선택 사항 입력)	기존 클라이언트를 사용할 사용자 풀 클라이언트(앱 클라이언트)를 제공합니다. 사용자 풀 클라이언트를 제공하지 않으면 솔루션이 하나를 생성합니다. 이 파라미터는 ExistingCognitoUserPoolId를 제공한 경우에만 제공할 수 있습니다.
ExistingCognitoGroupPolicyTableName	(선택 사항 입력)	사용자 그룹 정책이 포함된 DynamoDB 테이블의 이름입니다. 이는 사용 사례의 API에서 사용자 지정 권한 부여자가 사용합니다. 참고: 일반적으로 배포 대시보드에서 배포할 때 이 이름을 제공하지만이 사용 사례 스택을 독립적으로 배포할 때는 생략할 수 있습니다.
VpcEnabled	No	스택 리소스를 VPC 내에 배포할지 여부입니다.
CreateNewVpc	No	솔루션이 새 VPC를 생성하고 이 사용 사례에 사용할Yes지 여부를 선택합니다. 참고:이 파라미터는 VpcEnabled가 인 경우에만 관련이 있습니다Yes.

파라미터	기본 항목	설명
IPAMPoolId	(선택 사항 입력)	IPAM을 사용하여 CIDR 범위를 할당하려면 사용할 IPAM 풀 ID를 제공합니다. 참고:이 파라미터는 VpcEnabled가 Yes 이고 CreateNewVpc가 인 경우에만 관련이 있습니다No.
ExistingVpcId	(선택 사항 입력)	사용 사례에 사용할 기존 VPC의 VPC ID입니다. 참고:이 파라미터는 VpcEnabled가 Yes 이고 CreateNewVpc가 인 경우에만 관련이 있습니다No.
ExistingPrivateSubnetIds	(선택 사항 입력)	Lambda 함수를 배포하는 데 사용할 기존 프라이빗 서브넷IDs를 쉼표로 구분한 목록입니다. 참고:이 파라미터는 VpcEnabled가 Yes 이고 CreateNewVpc가 인 경우에만 관련이 있습니다No.
ExistingSecurityGroupIds	(선택 사항 입력)	Lambda 함수를 구성하는 데 사용할 기존 VPC의 보안 그룹을 쉼표로 구분한 목록입니다. 참고:이 파라미터는 VpcEnabled가 Yes 이고 CreateNewVpc가 인 경우에만 관련이 있습니다No.
VpcAzs	(선택 사항 입력)	VPCs의 서브넷이 생성되는 AZs의 쉼표로 구분된 목록 참고: VpcEnabled가 Yes 이고 CreateNewVpc가 인 경우에만 관련이 있습니다No.

파라미터	기본 항목	설명
BedrockAgentId	*<## ##>*	사용할 Amazon Bedrock 에이전트의 ID입니다.
BedrockAgentAliasId	*<## ##>*	사용할 Amazon Bedrock 에이전트의 별칭 ID입니다.
DeployUI	Yes	이 배포에 대한 프런트엔드 채팅 UI를 배포하는 옵션을 선택합니다. No 선택하면 APIs를 호스팅하기 위한 인프라, API에 대한 인증 APIs, 채팅 UI 없이 백엔드 처리가 생성됩니다.

6. 다음을 선택합니다.
7. 스택 옵션 구성 페이지에서 다음을 선택합니다.
8. 검토 페이지에서 설정을 검토하고 확인합니다. 템플릿이 IAM 리소스를 생성함을 확인하는 확인란을 선택하세요.
9. 스택 생성을 선택하여 스택을 배포합니다.

AWS CloudFormation 콘솔의 상태 열에서 스택의 상태를 볼 수 있습니다. 약 10~30분 후에 CREATE_COMPLETE 상태를 받게 됩니다.

DynamoDB 채팅 구성 제공

사용 사례를 배포할 때 UseCaseConfigRecordKey 및 UseCaseConfigTableName은 일반적으로 배포 대시보드에 의해 채워지는 필수 CloudFormation 파라미터입니다. 배포 대시보드 스택은이 테이블의 생성 및 구성을 처리하는 반면,는 파라미터의 배포 API 트리거 모집단을 호출합니다.

독립 실행형 배포를 수행할 때는 다음을 수행해야 합니다.

1. 키의 해시 키를 사용하여 DynamoDB 테이블을 생성합니다.
2. 테이블에 사용 사례에 대한 구성을 형식의 레코드로 포함하는 레코드를 생성합니다. `{key: some_use_case_key, config: {your_configuration}.`

3. 배포 시 선택한 UseCaseConfigTableName 및 UseCaseConfigRecordKey(`some_use_case_key`이 예제에서는) 파라미터를 사용 사례 스택에 전달합니다.

독립 실행형 배포에 적합한 구성을 생성하려면 배포 대시보드에서 필요한 사용 사례를 생성하고 구성 테이블에서 레코드를 복사할 수 있습니다. 그렇지 않으면 Bedrock 배포에 대한 다음 예제를 기반으로 자체 구성을 만들 수 있습니다.

```
{
 "UseCaseName": "SampleUseCase",
 "ConversationMemoryParams": {
 "ConversationMemoryType": "DynamoDB",
 "HumanPrefix": "H",
 "AiPrefix": "A",
 "ChatHistoryLength": 20
 },
 "KnowledgeBaseParams": {
 "KnowledgeBaseType": "Bedrock",
 "NumberOfDocs": 2,
 "ScoreThreshold": 0,
 "ReturnSourceDocs": false,
 "BedrockKnowledgeBaseParams": {
 "BedrockKnowledgeBaseId": "SOME_ID",
 "OverrideSearchType": null
 }
 },
 "LlmParams": {
 "ModelProvider": "Bedrock",
 "BedrockLlmParams": { "ModelId": "anthropic.claude-v2" },
 "PromptParams": {
 "PromptTemplate": "some prompt",
 "MaxPromptTemplateLength": 187500,
 "MaxInputTextLength": 187500,
 "UserPromptEditingEnabled": true,
 "DisambiguationEnabled": true,
 "DisambiguationPromptTemplate": "some prompt"
 },
 "ModelParams": {},
 "Temperature": 1,
 "RAGEnabled": true,
 "Streaming": true,
 "Verbose": false
```

```
  }
}
```

Service Catalog AppRegistry를 사용하여 솔루션 모니터링

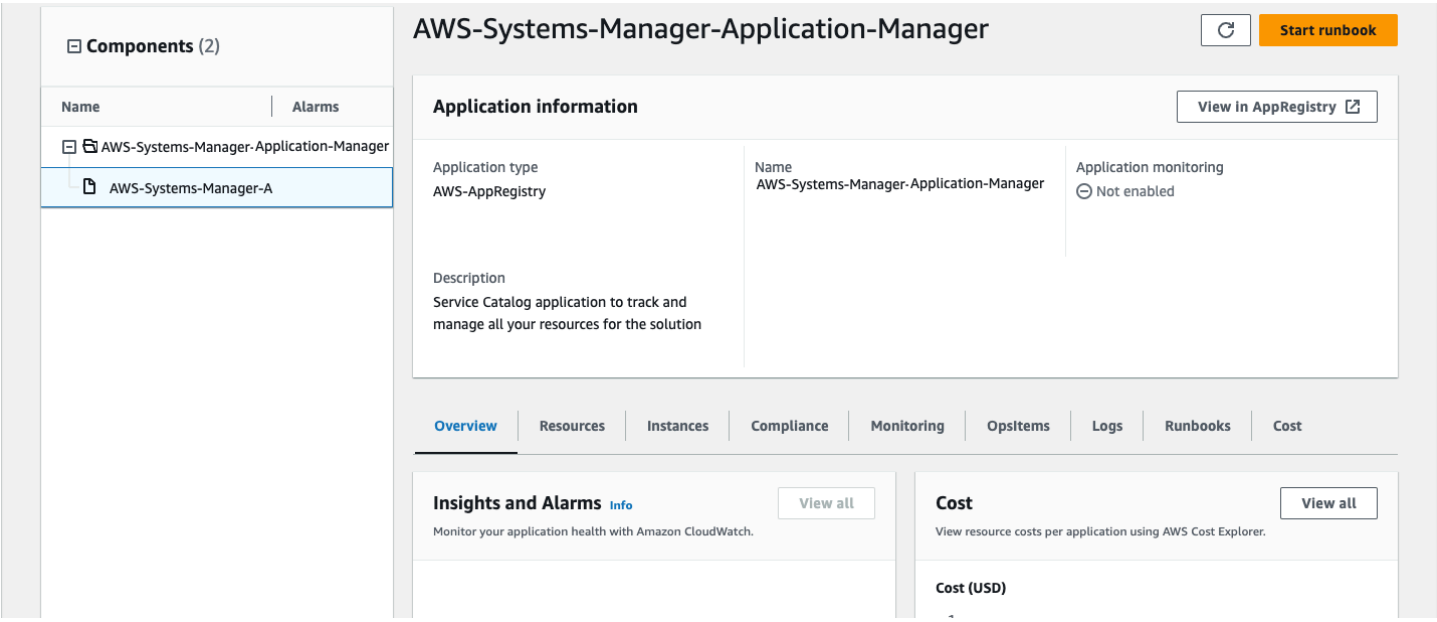
솔루션에는 CloudFormation 템플릿과 기본 리소스를 Service Catalog AppRegistry 및 Systems Manager Application Manager 모두에 애플리케이션으로 등록하는 Service Catalog AppRegistry 리소스가 포함되어 있습니다.

Systems Manager Application Manager는이 솔루션과 해당 리소스에 대한 애플리케이션 수준 보기를 제공하므로 다음을 수행할 수 있습니다.

- 중앙 위치에서 리소스, 스택 및 AWS 계정에서 배포된 리소스 비용,이 솔루션과 관련된 로그를 모니터링합니다.
- 애플리케이션의 컨텍스트에서이 솔루션의 리소스에 대한 작업 데이터를 봅니다. 배포 상태, CloudWatch 경보, 리소스 구성 및 운영 문제를 예로 들 수 있습니다.

다음 그림은 Application Manager의 솔루션 스택에 대한 애플리케이션 보기의 예를 보여줍니다.

Application Manager의 솔루션 스택을 보여줍니다.



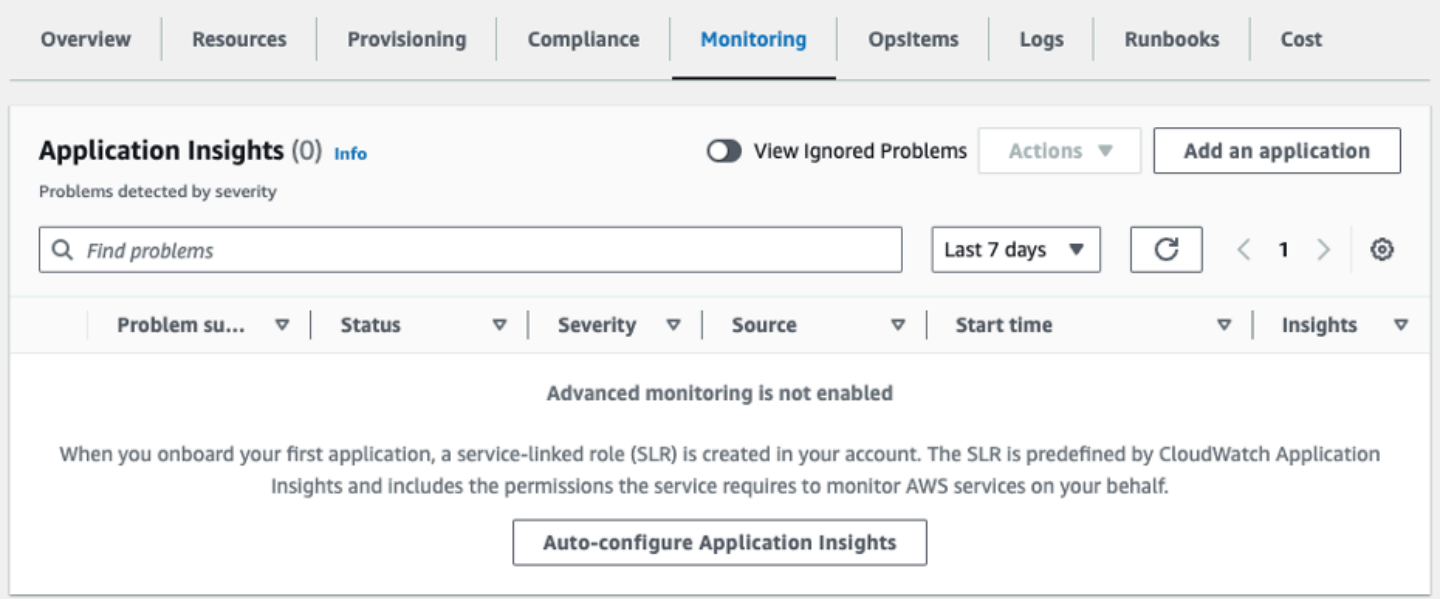
CloudWatch Application Insights 활성화

1. Systems Manager 콘솔에 로그인합니다.
2. 탐색 창에서 Application Manager를 선택합니다.
3. 애플리케이션에서이 솔루션의 애플리케이션 이름을 검색하고 선택합니다.

애플리케이션 이름은 애플리케이션 소스 열에 앱 레지스트리가 있고 솔루션 이름, 리전, 계정 ID 또는 스택 이름의 조합이 있습니다.

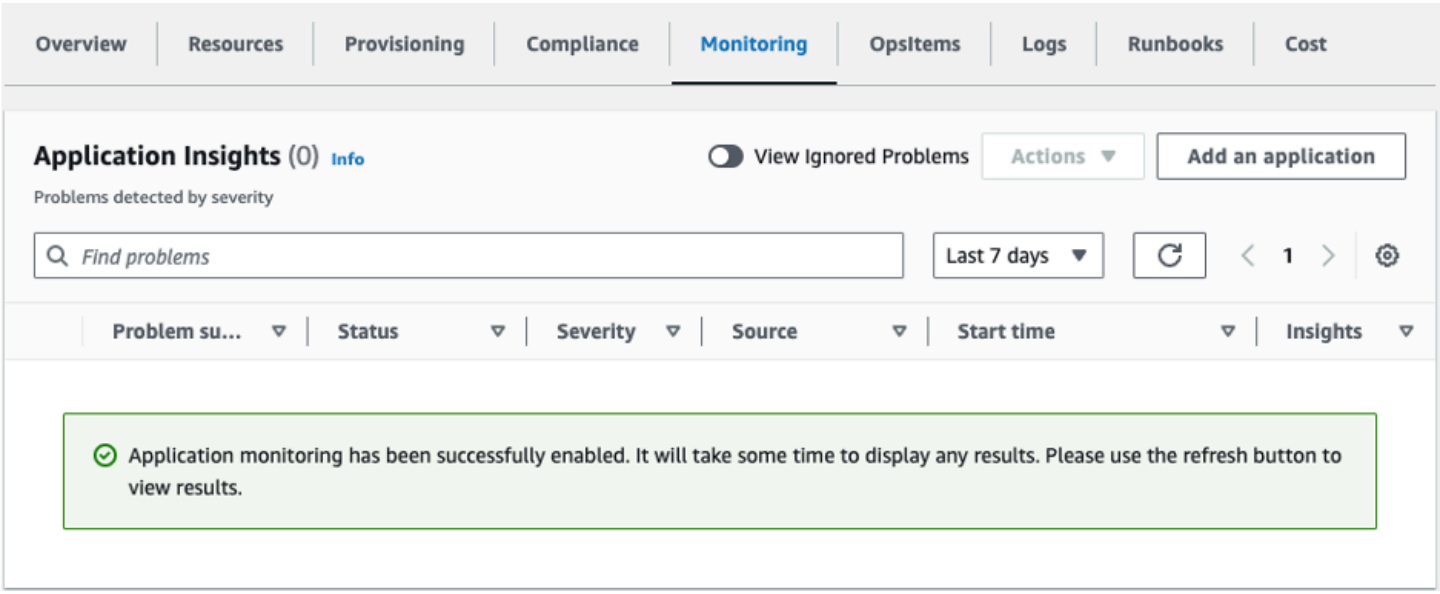
4. 구성 요소 트리에서 활성화하려는 애플리케이션 스택을 선택합니다.
5. 모니터링 탭의 Application Insights에서 Application Insights 자동 구성을 선택합니다.

감지된 문제가 없고 자동 구성 옵션이 표시된 Application Insights 대시보드입니다.



이제 애플리케이션 모니터링이 활성화되고 다음 상태 상자가 나타납니다.

성공적인 모니터링 활성화 메시지를 보여주는 Application Insights 대시보드입니다.



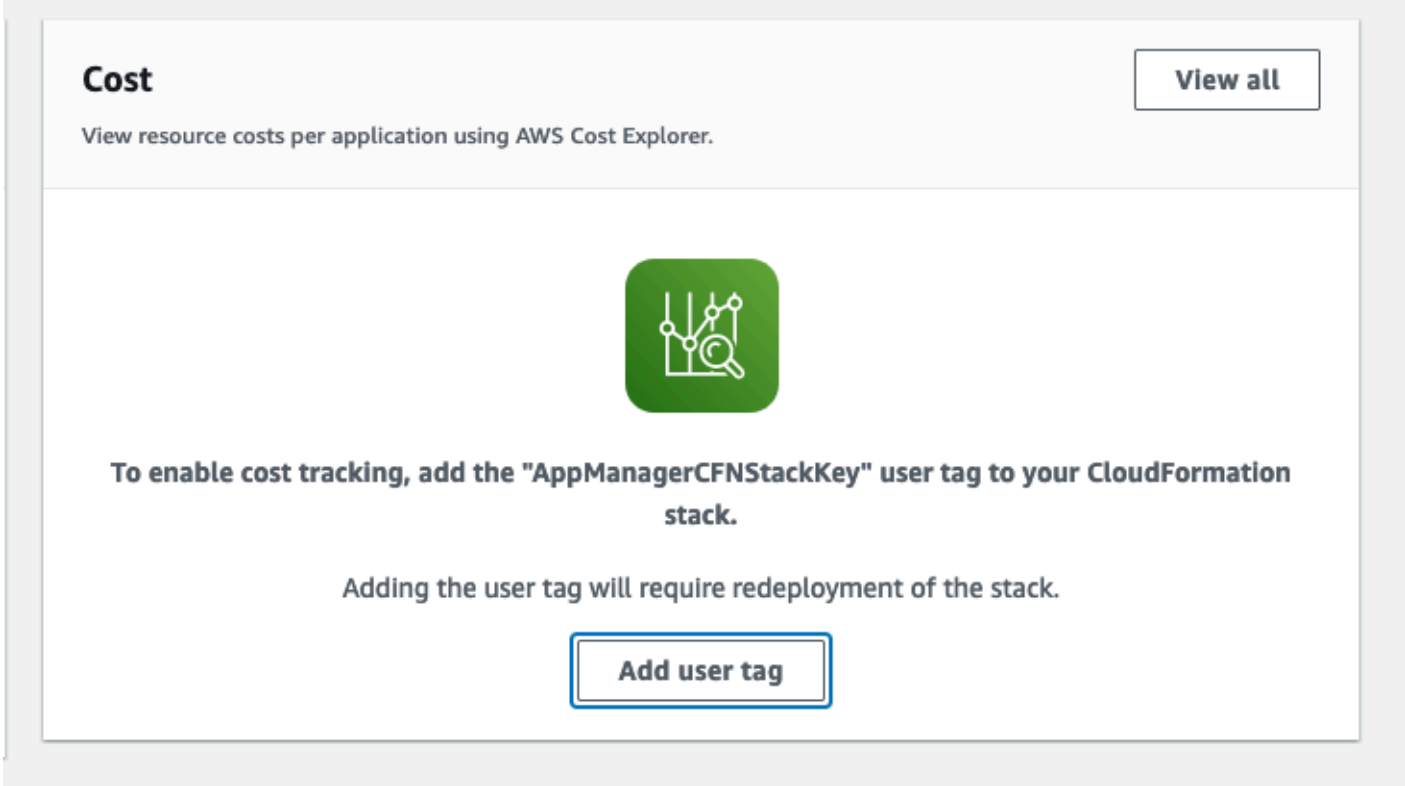
솔루션과 연결된 비용 태그 확인

솔루션과 관련된 비용 할당 태그를 활성화한 후 이 솔루션의 비용을 보려면 비용 할당 태그를 확인해야 합니다. 비용 할당 태그를 확인하려면 다음을 수행합니다.

1. Systems Manager 콘솔에 로그인합니다.
2. 탐색 창에서 Application Manager를 선택합니다.
3. 애플리케이션에서 이 솔루션의 애플리케이션 이름을 선택합니다.

   애플리케이션 이름은 애플리케이션 소스 열에 앱 레지스트리가 있고 솔루션 이름, 리전, 계정 ID 또는 스택 이름의 조합이 있습니다.
4. 개요 탭의 비용에서 사용자 태그 추가를 선택합니다.

   Application Cost 사용자 태그 추가 화면을 보여주는 스크린샷



5. 사용자 태그 추가 페이지에서 `confirm`를 입력한 다음 사용자 태그 추가를 선택합니다.

활성화 프로세스가 완료되고 태그 데이터가 표시되는 데 최대 24시간 정도 걸릴 수 있습니다.

솔루션과 관련된 비용 할당 태그 활성화

Cost Explorer를 활성화한 후 이 솔루션의 비용을 보려면 이 솔루션과 관련된 비용 할당 태그를 활성화해야 합니다. 비용 할당 태그는 조직의 관리 계정에서만 활성화할 수 있습니다. 비용 할당 태그를 활성화하려면 다음을 수행합니다.

1. AWS Billing and Cost Management and Cost Management 콘솔에 로그인합니다.
2. 탐색 창에서 비용 할당 태그를 선택합니다.
3. 비용 할당 태그 페이지에서 AppManagerCFNStackKey 태그를 필터링한 다음 표시된 결과에서 태그를 선택합니다.
4. 활성화를 선택합니다.

AWS Cost Explorer

먼저 활성화해야 하는 AWS Cost Explorer와의 통합을 통해 Application Manager 콘솔 내에서 애플리케이션 및 애플리케이션 구성 요소와 관련된 비용의 개요를 볼 수 있습니다. Cost Explorer는 시간 경과에 따른 AWS 리소스 비용 및 사용량을 파악하여 비용을 관리하는 데 도움이 됩니다. 솔루션에 대해 Cost Explorer를 활성화하려면 다음을 수행합니다.

1. AWS Cost Management 콘솔에 로그인합니다.
2. 탐색 창에서 Cost Explorer를 선택하여 시간 경과에 따른 솔루션의 비용 및 사용량을 확인합니다.

솔루션 업데이트

이전에 솔루션을 배포한 경우이 절차에 따라 솔루션의 CloudFormation 스택을 업데이트하여 최신 기능과 개선 사항을 확인합니다. 업그레이드 프로세스에는 세 부분이 있습니다.

- 1단계: 배포 대시보드 업데이트
- 2단계: 사용 사례 구성 마이그레이션
- 3단계: 사용 사례 업데이트

> **Note**
>
> 1. v2.0.0에서는 Anthropic 및 Hugging Face와의 통합이 Amazon Bedrock 및 Amazon SageMaker AI를 위해 더 이상 사용되지 않았습니다. SageMaker JumpStart를 통해 Hugging Face를 통해 사용 가능한 모델을 배포할 수 있습니다. 자세한 내용은 Amazon SageMaker AI에서 Hugging Face 사용을 참조하세요.
> 2. 이 단계를 실행하기 전에 비프로덕션 환경에서 업데이트 프로세스를 테스트해야 합니다.

1단계: 배포 대시보드 업데이트

1. CloudFormation 콘솔에 로그인하고 기존 CloudFormation 스택을 선택한 다음 업데이트를 선택합니다.
2. 현재 템플릿 교체를 선택합니다.
3. 템플릿 지정에서 다음을 수행합니다.
   a. Amazon S3 URL을 선택합니다.
   b. 최신 CloudFormation 템플릿 링크를 복사합니다.
   c. Amazon S3 URL 상자에 링크를 붙여넣습니다.
   d. Amazon S3 URL 텍스트 상자에 올바른 템플릿 URL이 표시되는지 확인하고 다음을 선택합니다. 다음을 다시 선택합니다.
4. 파라미터에서 템플릿의 파라미터를 검토하고 필요에 따라 수정합니다. 파라미터에 대한 자세한 내용은 1단계: 배포 대시보드 스택 시작을 참조하세요.
5. 다음을 선택합니다.
6. 스택 옵션 구성 페이지에서 다음을 선택합니다.

7. 검토 페이지에서 설정을 검토하고 확인합니다. 템플릿이 IAM 리소스를 생성할 것임을 확인하는 확인란을 선택합니다.
8. 변경 세트 보기를 선택하고 변경 사항을 확인합니다.
9. 스택 생성을 선택하여 스택을 배포합니다.

AWS CloudFormation 콘솔의 상태 열에서 스택의 상태를 볼 수 있습니다. 약 10분 후에 UPDATE_COMPLETE 상태가 표시됩니다.

기존 솔루션 버전이 v2.0.0 이전인 경우 업데이트하면 웹 UI 스택(로그인 화면 `amplify-ui` 구현을 Cognito 호스팅 UI로 대체)과 새 CloudFront URL이 생성되며, 스택 상태가 UPDATE_COMPLETE이면 CloudFormation 콘솔의 출력 섹션에서 가져올 수 있습니다.

> **Note**
>
> v2.0.0 이전 버전을 사용하여 생성된 기존 사용 사례는 아래 설명된 단계를 완료할 때까지 표시되지 않습니다.

2단계: 사용 사례 구성 마이그레이션(2.0.0 미만의 버전에서만 업데이트)

버전 2.0.0에서는 저장을 위한 스키마와 사용 사례 구성을 저장할 AWS 서비스가 변경되었습니다. gaab_v2_migration.py 스크립트를 사용하여 GAAB v2 마이그레이션 사용 설명서에 설명된 단계를 따릅니다. 스크립트를 실행한 후 배포 대시보드에 액세스하여 배포된 사용 사례를 볼 수 있습니다.

> **Note**
>
> 아래 단계에 따라 사용 사례 마이그레이션을 완료해야 합니다.

3단계: 사용 사례 업데이트

최신 버전의 GAAB에서 사용할 수 있는 새로운 기능을 사용하여 배포된 사용 사례를 편집할 수 있습니다. 이 솔루션의 기능을 사용하는 방법에 대한 자세한 내용은 솔루션 사용을 참조하세요.

사용 사례를 최신 버전으로 업데이트하려면 배포 대시보드에서 '편집'사용 사례 단계를 완료해야 합니다(변경할 수는 없음). 이 작업은 최신 템플릿 버전으로 CloudFormation 스택 업데이트를 트리거합니다.

> **Note**
>
> 솔루션의 1.x 또는 2.x 버전으로 생성된 사용 사례는 이후 버전에서는 작동하지 않을 수 있습니다. 따라서 배포 대시보드를 통해 v3.0.0 이전 버전으로 생성된 기존 사용 사례를 복제하는 것이 좋습니다. 그런 다음 v3.0.0 이상을 사용하여 생성된 새 사용 사례로 점진적으로 마이그레이션하고를 바꿉니다.

문제 해결

이 섹션에서는 솔루션 배포 및 사용에 대한 문제 해결 지침을 제공합니다.

이 지침으로 문제가 해결되지 않을 경우 Support에 문의하세요에서 이 솔루션에 대한 지원 사례를 여는 방법을 안내해 드립니다.

문제: VPC를 자동으로 생성하여 VPC 지원 구성 배포 실패

CloudFormation에서 VPC 네트워킹 리소스를 프로비저닝할 수 없어 배포 대시보드 스택 또는 사용 사례 스택이 배포에 실패합니다.

해결 방법

계정의 VPCs 및 탄력적 IPs에 대한 할당량 제한을 확인합니다. AWS 계정당, AWS 리전당 탄력적 IPs 및 VPCs에 대한 기본 한도는 각각 5개입니다.

> **Note**
>
> 솔루션이 VPC를 생성할 때 단일 VPC 지원 배포(배포 대시보드 또는 사용 사례)는 각 AZ에 퍼블릭 서브넷 1개와 프라이빗 서브넷 1개가 있는 2-AZ 배포이며 각 퍼블릭 서브넷은 NAT 게이트웨이 1개를 배포합니다. NAT 게이트웨이가 2개인 경우 배포는 할당량 한도에서 퍼블릭 IP 주소 2개를 사용합니다.

알아야 할 몇 가지 제한(계정별, 리전별):

- VPCs 수 - 5
- 퍼블릭 IP 주소 수 - 5
- 게이트웨이 VPC 엔드포인트 수 - 20
- 인터페이스 VPC 엔드포인트 수 - 20

문제: 배포 대시보드 스택이 삭제된 후에는 CloudFormation에서 사용 사례 스택을 삭제할 수 없습니다.

모든 사용 사례 스택이 삭제되기 전에 CloudFormation에서 배포 대시보드 스택이 삭제되면 사용 사례가 잠긴(사용할 수 없음) 상태가 될 수 있습니다. 이는 배포 대시보드 스택에 의해 생성된 IAM 역할이 더 이상 존재하지 않아 사용 사례 스택을 수정할 수 없기 때문입니다.

해결 방법

> **Warning**
>
> 사용 후 즉시 수동으로 생성된 역할을 정리해야 합니다. 이는 사용자가 역할 상승에 악용할 수 있는 승격된 권한입니다.

삭제된 IAM 역할을 다시 생성하여 CloudFormation 스택 삭제를 활성화합니다.

1. CloudFormation 콘솔을 열고 잠긴 스택과 연결된 역할을 확인합니다.
   a. 역할 ARN은 IAM 역할이라는 스택 정보 섹션에서 찾을 수 있습니다.
   b. 역할 이름은 IAM 역할 ARN에서 :role/ 뒤에 오는 이름입니다(예: arn:aws:iam::<account-id>:role/<role-name>).
2. 삭제된 역할과 동일한 이름으로 IAM에 새 역할을 생성합니다.
   a. AWS 서비스를 신뢰할 수 있는 엔터티로 선택하고 드롭다운에서 CloudFormation을 선택합니다.
   b. 필요한 권한을 추가합니다. 필요한 권한이 확실하지 않은 경우 AWS 관리형 AdministratorAccess 정책을 사용할 수 있습니다.
   c. 1단계에서 얻은 것과 정확히 동일한 역할 이름을 입력합니다.
3. CloudFormation 콘솔로 돌아가 잠긴 스택을 삭제합니다.
4. 잠긴 스택이 모두 성공적으로 삭제되면 IAM으로 돌아가 2단계에서 생성된 모든 역할을 삭제합니다.

문제: 사용 사례 UI에 설정 변경 사항이 반영되지 않음

사용 사례가 업데이트되면 UI가 CloudFront에 배포됩니다. 그러나 CloudFront는 배포와 일부 설정이 사용자에게 표시되는 방식을 지시하는 구성 파일을 캐시하므로 이러한 변경 사항은 즉시 반영되지 않을 수 있습니다.

해결 방법

CloudFront 배포를 무효화하여 새 구성을 프런트엔드 사용자에게 강제로 전파할 수 있습니다.

1. CloudFormation 콘솔을 열고 사용 사례 스택과 연결된 CloudFront 배포를 확인합니다.
   a. 사용 사례 스택은 사용 사례를 배포할 때 사용한 것과 동일한 이름으로 시작해야 합니다.
   b. UI에 해당하는 중첩 스택을 찾습니다. 중첩 스택 이름은 WebAppS3UINestedStackS3UINestedStackResource로 시작해야 합니다.
   c. 리소스 탭에서 AWS::CloudFront::Distribution 유형의 리소스를 찾은 다음 물리적 ID를 선택합니다. 그러면 CloudFront 콘솔에서 배포가 열립니다.
2. 무효화 탭으로 이동한 다음 무효화 생성을 선택하고 /*의 경로를 입력합니다. 이렇게 하면 모든 경로가 무효화됩니다.
3. 자체 브라우저에서 사용 사례와 관련된 쿠키 및 캐시된 파일을 삭제합니다.

AWS Support에 문의

AWS Business Support+, AWS Enterprise Support 또는 통합 운영이 있는 경우 AWS 지원 센터를 사용하여이 솔루션에 대한 전문가 지원을 받을 수 있습니다. 이후 단원에서는 그 방법에 대해서 설명합니다.

사례 생성

1. 지원 센터에 로그인합니다.
2. 사례 생성을 선택합니다.

지원 방법

1. 기술을 선택합니다.
2. 서비스에서 솔루션을 선택합니다.
3. 범주에서 기타 솔루션을 선택합니다.
4. 심각도에서 사용 사례에 가장 적합한 옵션을 선택합니다.
5. 서비스, 카테고리 및 심각도를 입력하면 인터페이스가 일반적인 문제 해결 질문에 대한 링크를 제공합니다. 이러한 링크로 질문을 해결할 수 없는 경우 다음 단계: 추가 정보를 선택합니다.

추가 정보

1. 제목에 질문 또는 문제를 요약하는 텍스트를 입력합니다.
2. 설명에서 AWS의 생성형 AI Application Builder 솔루션 이름을 포함하여 문제를 자세히 설명합니다.
3. 파일 연결을 선택합니다.
4. AWS Support에서 요청을 처리하는 데 필요한 정보를 첨부합니다.

사례를 더 빠르게 해결할 수 있도록 지원

1. 필요한 정보를 입력합니다.
2. 다음 단계: 지금 해결하거나 AWS에 문의하기를 선택합니다.

지금 해결 또는 문의

1. 지금 해결 솔루션을 검토합니다.
2. 이러한 솔루션의 문제를 해결할 수 없는 경우 문의를 선택하고 요청된 정보를 입력한 다음 제출을 선택합니다.

솔루션 제거

> **Note**
>
> 배포 대시보드를 통해 생성된 배포는 솔루션 외부에서 관리되지 않습니다. CloudFormation에서 스택을 삭제하기 전에 배포 대시보드 내에서 모든 배포를 삭제하고 정리해야 합니다.

AWS 관리 콘솔에서 또는 AWS 명령줄 인터페이스를 사용하여 생성형 AI Application Builder on AWS 솔루션을 제거할 수 있습니다. 이 솔루션에서 생성한 Amazon S3 버킷, Amazon Kendra 인덱스 또는 CloudWatch Logs를 수동으로 삭제해야 합니다. AWS Solutions는 보존할 데이터를 저장한 경우 Amazon S3 버킷, Amazon Kendra 인덱스 또는 CloudWatch Logs를 자동으로 삭제하지 않습니다.

AWS 관리 콘솔 사용

1. AWS CloudFormation 콘솔에 로그인합니다.
2. 스택 페이지에서 이 솔루션의 설치 스택을 선택합니다.
3. 삭제를 선택합니다.

AWS Command Line Interface 사용

사용자 환경에서 AWS Command Line Interface(AWS CLI)를 사용할 수 있는지 확인합니다. 설치 지침은 AWS CLI 사용 설명서의 AWS 명령줄 인터페이스란 무엇입니까?를 참조하세요. AWS CLI를 사용할 수 있는지 확인한 후 다음 명령을 실행합니다.

```
$ aws cloudformation delete-stack --stack-name <installation-stack-name>
```

수동 제거 단계

Amazon S3 버킷 삭제

이 솔루션은 실수로 데이터가 손실되지 않도록 AWS CloudFormation 스택을 삭제하기로 결정한 경우 솔루션이 생성한 Amazon S3 버킷을 유지하도록 구성됩니다. 솔루션을 제거한 후 데이터를 보존할 필요가 없는 경우이 Amazon S3 버킷을 수동으로 삭제할 수 있습니다. 다음 단계에 따라 Amazon S3 버킷을 삭제합니다.

1. Amazon S3 콘솔에 로그인합니다.
2. 탐색 창에서 버킷을 선택합니다.
3. <stack-name> S3 버킷을 찾습니다.
4. S3 버킷을 선택하고 삭제를 선택합니다.

AWS CLI를 사용하여 S3 버킷을 삭제하려면 다음 명령을 실행합니다. --force 옵션을 사용할 때 먼저 버킷을 비울 필요가 없습니다.

```
$ aws s3 rb s3://<bucket-name> --force
```

Amazon Kendra 인덱스 삭제

우발적인 데이터 손실을 방지하기 위해이 솔루션은 AWS CloudFormation 스택이 삭제될 때 솔루션 생성 Amazon Kendra 인덱스를 유지하도록 구성됩니다. 솔루션을 제거한 후 더 이상 데이터를 보존할 필요가 없는 Amazon Kendra 인덱스를 수동으로 삭제할 수 있습니다. 다음 단계에 따라 Amazon Kendra 인덱스를 삭제합니다.

1. Amazon Kendra 콘솔에 로그인합니다.
2. 탐색 창에서 인덱스를 선택합니다.
3. 삭제할 인덱스를 찾아 선택합니다.
4. 삭제를 선택하여 선택한 인덱스를 삭제합니다.

AWS CLI를 사용하여 Amazon Kendra 인덱스를 삭제하려면 다음 명령을 실행합니다.

```
$ aws kendra delete-index --id<index-id>
```

CloudWatch Logs 삭제

우발적인 데이터 손실을 방지하기 위해 CloudFormation 스택을 삭제하기로 결정한 경우 CloudWatch Logs를 유지하도록이 솔루션을 구성했습니다. 솔루션을 제거한 후 데이터를 보존할 필요가 없는 경우 로그를 수동으로 삭제할 수 있습니다. 다음 단계에 따라 CloudWatch Logs를 삭제합니다.

1. Amazon CloudWatch 콘솔에 로그인합니다.
2. 탐색 창에서 로그 그룹을 선택합니다.
3. 솔루션에서 생성한 로그 그룹을 찾습니다.

4. 로그 그룹 중 하나를 선택합니다.
5. Actions를 선택하고 삭제를 선택합니다.

모든 솔루션 로그 그룹을 삭제할 때까지 단계를 반복합니다.

솔루션 사용

UI 액세스

스택 배포 프로세스(배포 대시보드 및 사용 사례 모두) 중에 구성된 이메일 주소로 이메일이 전송됩니다. 이메일에는 웹 인터페이스에 가입하고 액세스하는 데 사용할 수 있는 사용자의 임시 자격 증명이 포함되어 있습니다.

> **Note**
>
> AWS Management Console에 액세스할 수 있는 DevOps 사용자는 스택이 완료될 때 배포 대시보드 UI의 CloudFront URL을 관리자 사용자에게 제공해야 합니다.

사용 사례의 경우 배포 대시보드 UI에 액세스할 수 있는 관리자 사용자는 배포가 완료될 때 비즈니스 사용자에게 사용 사례 UI의 CloudFront URL을 제공해야 합니다.

로그인한 사용자는 관리자의 경우 배포 대시보드, 비즈니스 사용자의 경우 사용 사례 등 솔루션 UIs와 상호 작용할 수 있습니다.

배포를 업데이트하는 방법

배포 대시보드 홈 페이지(또는 배포의 세부 정보 페이지)에서 배포에 사용되는 구성을 편집할 수 있습니다. CREATE_COMPLETE 또는 UPDATE_COMPLETE 상태인 배포만 편집할 수 있습니다.

사용 사례 이름을 제외하고 배포에 대해 다른 모든 옵션을 편집할 수 있습니다. 편집 및 재배포하려는 값을 변경하기만 하면 됩니다.

편집 범위에 따라 재배포 시간이 달라집니다. 간단한 설정이 변경된 경우(예: 모델 파라미터) 몇 초, 더 큰 인프라 관련 옵션이 변경된 경우(예: 텍스트 사용 사례 RAG에 대한 Amazon Kendra 인덱스 생성 요청) 30분 이상 걸릴 수 있습니다.

편집이 성공적으로 완료되면 애플리케이션 상태는 UPDATE_COMPLETE 상태를 보고합니다. 이때 CloudFront URL을 통해 배포된 UI에 액세스하고 수정된 배포와 상호 작용할 수 있습니다.

> **Note**
>
> 다른 설정 또는 LLM을 비교하려는 경우 여러 배포를 side-by-side 실행하는 것이 더 쉬울 수 있습니다. LLMs 복제 기능을 사용하면 기존 구성을 빠르게 사용하여 새 배포를 시작할 수 있습니다.

배포를 복제하는 방법

배포 대시보드 홈 페이지(또는 배포의 세부 정보 페이지)에서 배포에 사용되는 구성을 복제할 수 있습니다. 배포를 복제하면 새 사용 사례 배포 마법사가 시작되지만 대부분의 필드는 동일한 값으로 미리 채워집니다.

이는 설정이 변경된 배포를 빠르게 복제하거나, 삭제된 배포를 재생성하거나, 동일한 배포에서 여러 LLMs 비교하는 데 도움이 되는 편리한 작업입니다.

배포를 삭제하는 방법

배포 대시보드 홈 페이지(또는 배포의 세부 정보 페이지)에서 배포가 더 이상 필요하지 않으면 삭제할 수 있습니다. 배포를 삭제하면 CloudFormation 스택 삭제 작업이 호출되고 배포를 위한 리소스 프로비저닝이 해제됩니다.

기본적으로 삭제된 배포는 여전히 대시보드에 남아 복제 기능을 활성화합니다. UI에서 추적이 중지되도록 대시보드에서 배포를 완전히 제거하려면 삭제 확인 창에서 영구 삭제를 선택합니다.

> **Important**
>
> 일부 리소스는 스택 삭제 중에 남아 있으므로 수동으로 삭제해야 합니다. 보존되는 리소스와 정리 방법에 대한 자세한 내용은 수동 제거 섹션을 참조하세요.

대규모 언어 모델(LLM) 구성

사용 사례에 적합한 LLM은 요구 사항과 큐레이션하려는 고객 경험 유형에 따라 달라집니다. 이 솔루션은 규범적인 것으로 보이지 않지만 애플리케이션에 가장 적합한 도구를 평가하는 데 필요한 도구를 제공하는 것을 목표로 합니다.

AI 생성 공간은 빠르게 진화하고 있으므로 최신 모델, 최적화 기술 및 모범 사례를 최신 상태로 유지하여 고객에게 적합한 경험을 구축할 수 있습니다.

> **Note**
>
> 비공개 또는 민감한 데이터로 작업하는 경우 AWS 서비스(예: Amazon Bedrock 또는 Amazon SageMaker AI)를 사용하여 LLM 옵션을 선택해야 합니다. 이렇게 하면 타사 공급자가 호스팅하는 LLM을 사용하는 것과 비교할 때 데이터를 리전 및 AWS 네트워크에 보관하여 배포의 전반적인 보안 태세가 향상됩니다.

Amazon SageMaker AI를 LLM 공급자로 사용

v1.3.0부터 Amazon SageMaker AI를 텍스트 사용 사례의 모델 공급자로 사용할 수 있습니다. 이 기능을 사용하면 솔루션의 AWS 계정 내에 이미 있는 SageMaker AI 추론 엔드포인트를 사용할 수 있습니다. 다음은 시작하는 몇 가지 방법입니다.

> **Important**
>
> 솔루션은 SageMaker AI 엔드포인트의 수명 주기를 관리하지 않습니다. 추가 요금 발생을 중단할 필요가 없는 SageMaker AI 엔드포인트를 삭제하는 것은 사용자의 책임입니다.

SageMaker AI 엔드포인트 생성

Amazon SageMaker AI JumpStart를 사용하여 엔드포인트를 빠르게 배포할 수 있습니다.

텍스트 생성 기반 SageMaker AI 엔드포인트를 사용하고 기본 SageMaker AI 서비스를 사용하여 배포할 수도 있습니다. 추론용 모델을 배포하는 방법에 대한 단계별 가이드는 SageMaker AI JumpStart 설명서를 참조하세요.

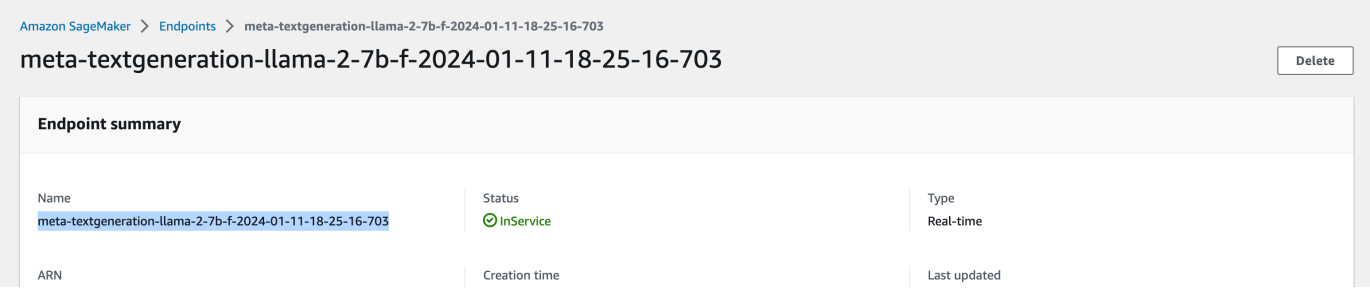
> **Note**
>
> 파운데이션 모델/LLMs 일반적으로 상당히 크고 종종 대규모 가속 컴퓨팅 인스턴스를 사용해야 할 수 있습니다. 이러한 더 큰 인스턴스 중 다수는 기본적으로 AWS 계정에서 사용하지 못할 수 있습니다. 일반적인 배포 실패를 방지하려면 배포하기 전에 기본 SageMaker AI 할당량을 참조하고 할당량 증가를 요청해야 합니다.

SageMaker AI 엔드포인트를 사용하여 텍스트 사용 사례 배포 생성

추론을 위해 SageMaker AI 엔드포인트를 사용하여 새 텍스트 사용 사례를 배포하려면:

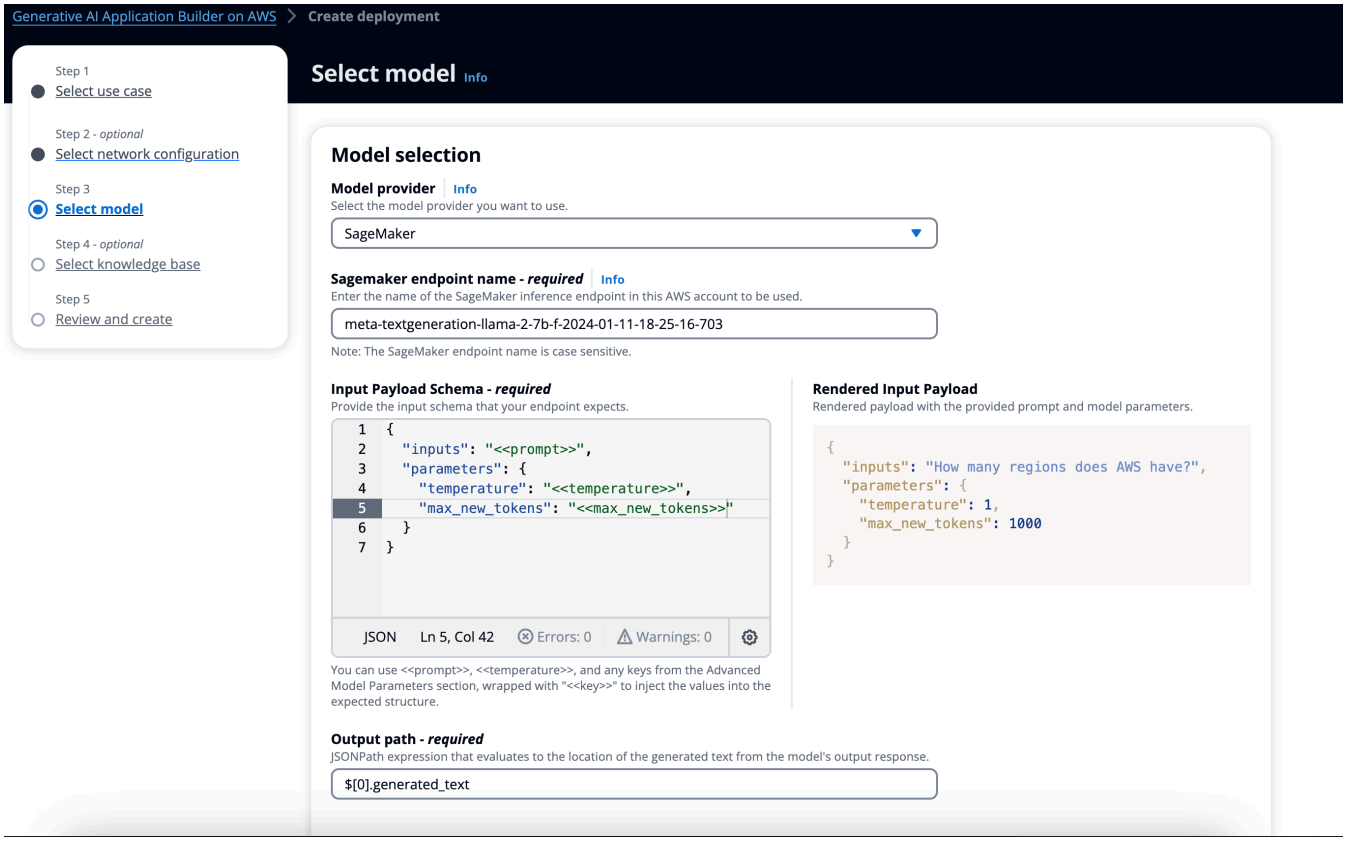
1. 배포 대시보드 마법사를 통해 새 사용 사례를 생성하고 모델 선택 페이지에 도달할 때까지 양식을 작성합니다.
2. 모델 페이지에서 모델 공급자로 SageMaker AI를 선택합니다. 이렇게 하면 세 가지 주요 사용자 입력이 필요한 사용자 지정 양식이 생성됩니다.
   - 사용하려는 SageMaker AI 엔드포인트의 이름입니다. DevOps 사용자는 AWS 콘솔에서 이를 얻을 수 있습니다. 엔드포인트는 솔루션이 배포된 것과 동일한 계정 및 리전에 있어야 합니다.

     AWS 콘솔에서 엔드포인트 이름의 위치



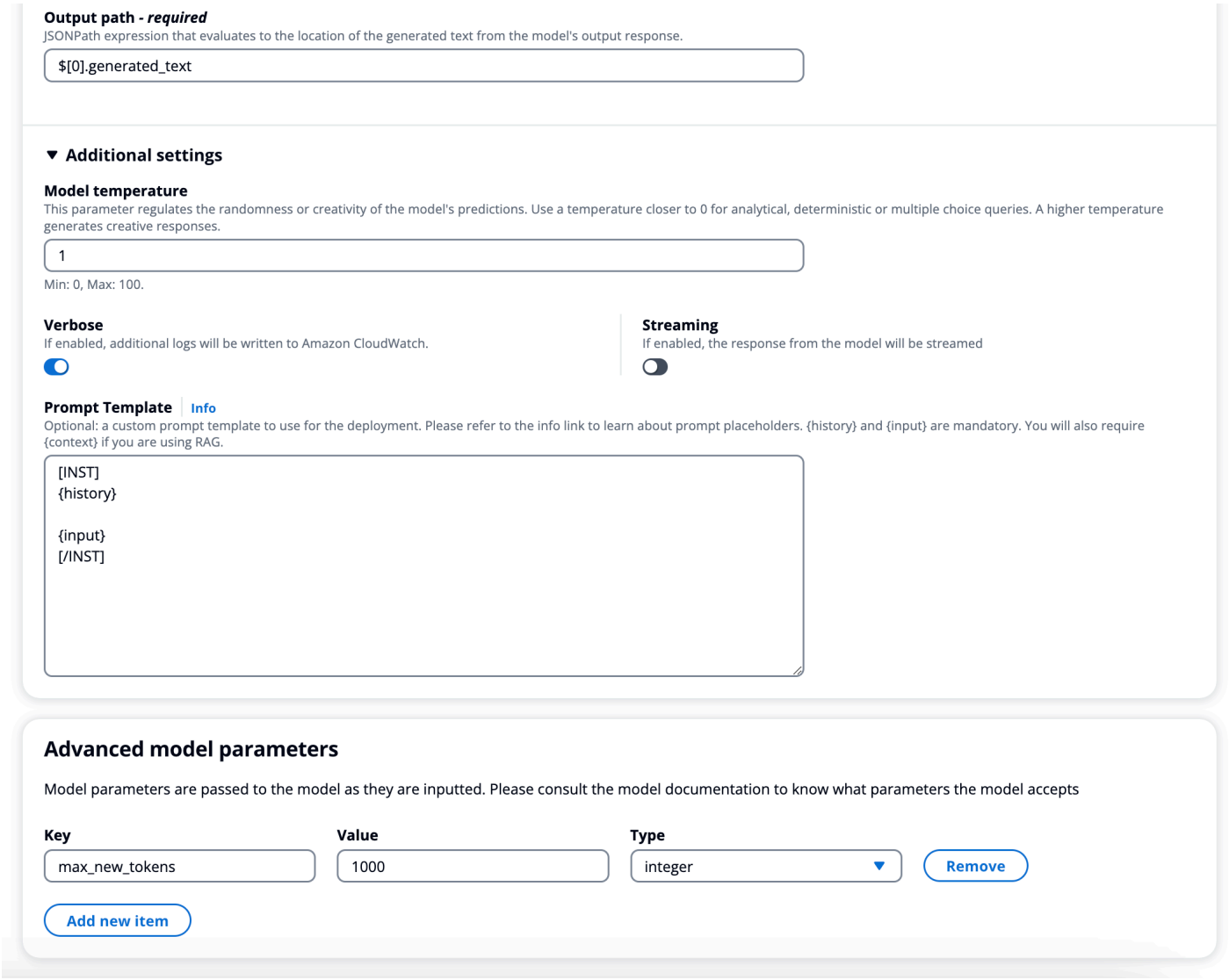
   - 엔드포인트에서 예상되는 입력 페이로드의 스키마입니다. 가장 광범위한 엔드포인트 세트를 지원하려면 관리자 사용자는 엔드포인트가 입력 형식 지정을 어떻게 예상하는지 솔루션에 알려야 합니다. 모델 선택 마법사에서 엔드포인트로 전송할 솔루션의 JSON 스키마를 제공합니다. 자리 표시자를 추가하여 요청 페이로드에 정적 및 동적 값을 주입할 수 있습니다. 사용할 수 있는 옵션:
     - 필수 자리 표시자: \<\<prompt\>\>는 런타임 시 SageMaker AI 엔드포인트로 전송할 전체 입력(예: 프롬프트 템플릿에 따른 기록, 컨텍스트 및 사용자 입력)으로 동적으로 대체됩니다.
     - 선택적 자리 표시자: \<\< temperature\>\> *,* 및 고급 모델 파라미터에 정의된 모든 파라미터를 엔드포인트에 제공할 수 있습니다. \<\< 및 \>\>로 묶인 자리 표시자를 포함하는 모든 문자열(예: \<\<max_new_tokens\>\>)은 동일한 이름의 고급 모델 파라미터 값으로 대체됩니다.

       입력 스키마 예제 - 사용자 지정 고급 파라미터인 max_new_tokens와 함께 필수 필드, 프롬프트 및 온도를 설정합니다. 출력 경로는 유효한 JSONPath 문자열로 제공되어야 합니다.



3. 출력 페이로드 내에서 LLMs 생성한 문자열 응답의 위치입니다. 이는 엔드포인트의 반환 객체 및 응답 내에서 사용자에게 표시되는 최종 텍스트 응답에 액세스할 것으로 예상되는 위치를 나타내는 JSONPath 표현식으로 제공되어야 합니다.

   SageMaker AI 입력 스키마 내에서 사용할 고급 모델 파라미터를 추가하는 예제(이전 옵션/설정은 그림 2 참조)



> **Note**
>
> SageMaker AI는 이제 동일한 엔드포인트 뒤에 여러 모델을 호스팅하도록 지원하며, 이는 현재 버전의 SageMaker AI Studio(Studio Classic 아님)에 엔드포인트를 배포할 때 기본 구성입니다.
>
> 엔드포인트가 이러한 방식으로 구성된 경우 사용하려는 모델의 이름에 해당하는 값과 함께 InferenceComponentName을 고급 모델 파라미터 섹션에 추가해야 합니다.

고급 LLM 설정

Amazon Bedrock을 사용하는 동안 Amazon Bedrock 가드레일, Amazon Bedrock용 프로비저닝된 처리량, 추가 모델 파라미터와 같은 모델에 대한 일부 고급 설정을 구성할 수 있습니다.

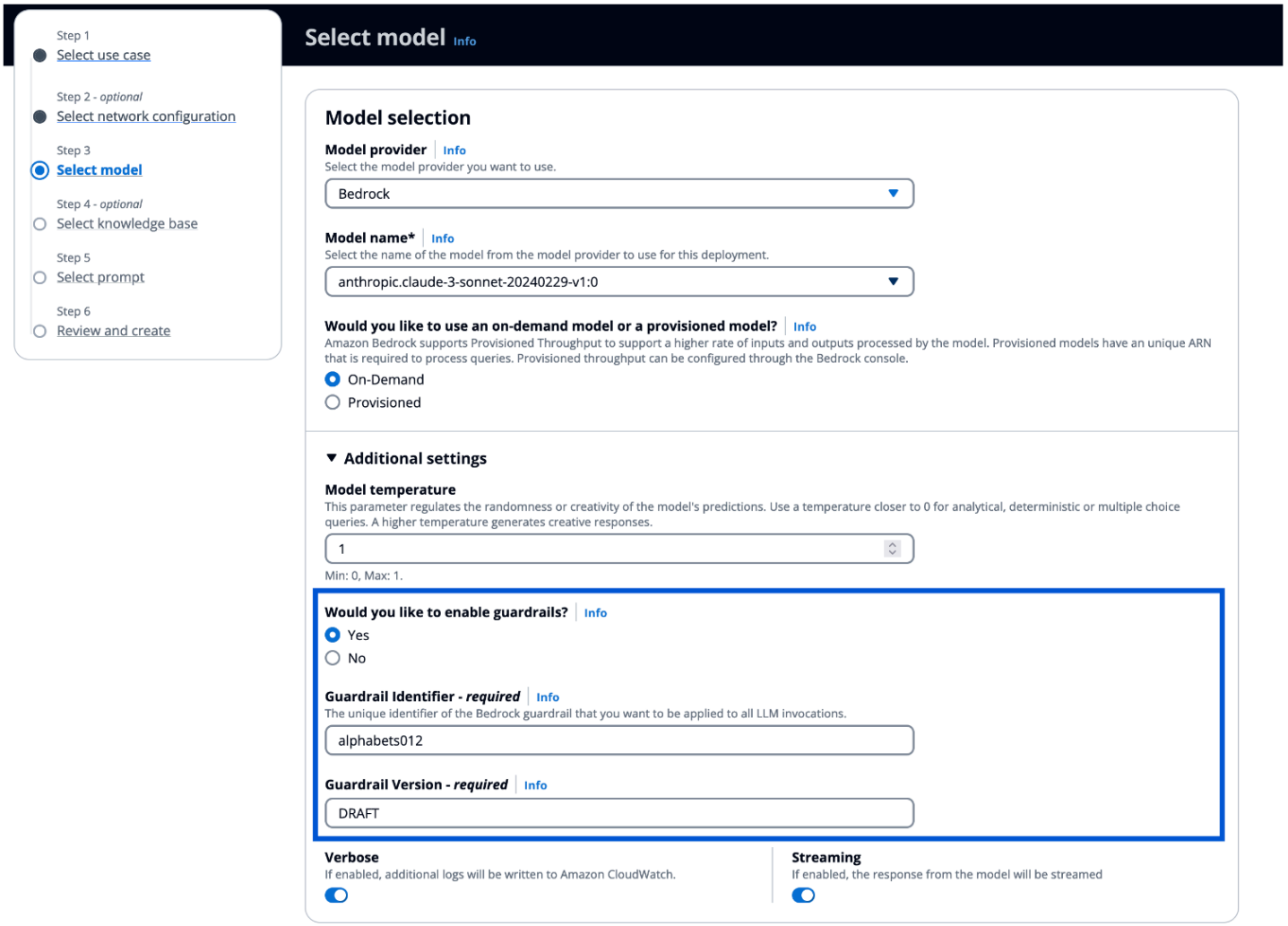
Amazon Bedrock Guardrails

Amazon Bedrock Guardrails는 사용자 구성 정책에 따라 사용자 입력 및 LLM 응답을 평가하고 사용자가 사용 사례에 대해 선택하는 기본 LLM에 관계없이 추가 보호 계층을 제공하는 Amazon Bedrock의 기능입니다. 가드레일은 바람직하지 않거나 유해한 범주에 속하는 콘텐츠를 방지하기 위한 2가지 정책으로 구성됩니다.

1. 금융 애플리케이션에 대한 투자 조언과 같이 사용자의 애플리케이션 컨텍스트에서 바람직하지 않은 주제 세트를 정의하기 위해 주제를 거부했습니다.
2. 콘텐츠 필터****유해한 콘텐츠가 포함된 입력 사용자 프롬프트 또는 모델 응답을 필터링할 수 있습니다.

생성형 AI Application Builder 솔루션에서 사용하려면 가드레일 생성 마법사를 사용하여 Amazon Bedrock 콘솔에서 가드레일을 구성해야 합니다. 생성되면 Guardrail 식별자 및 Guardrail 버전을 제공하여 모델 선택 단계의 추가 설정에서 생성형 AI Application Builder 솔루션 마법사를 통해 생성된 채팅 사용 사례에이 Guardrail을 추가할 수 있습니다.

Depicts Deployment Wizard - Amazon Bedrock Guardrails 활성화

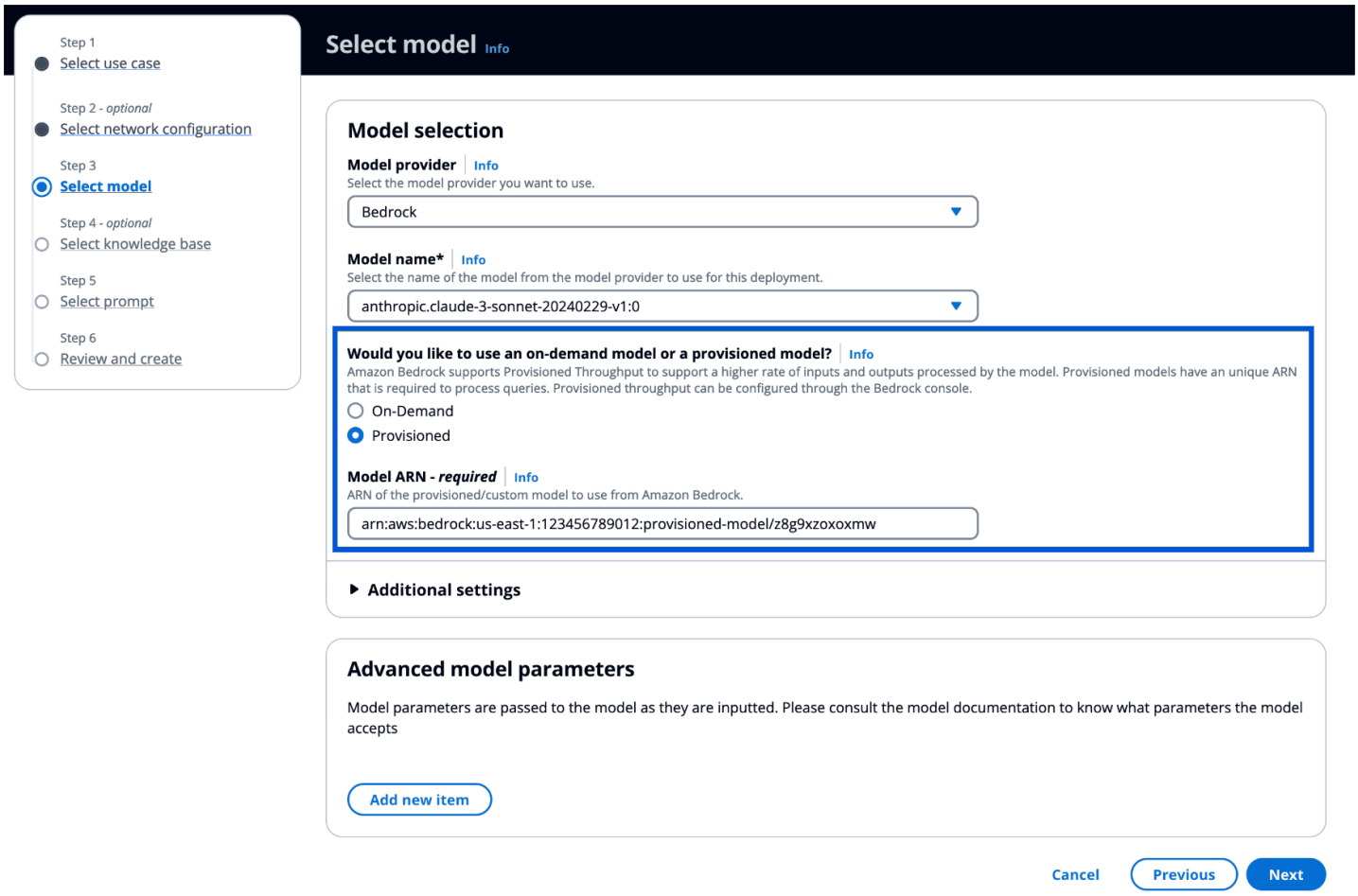


Amazon Bedrock의 프로비저닝된 처리량

각 온디맨드 Amazon Bedrock 모델은 모델 추론에 대한 리전별 계정 할당량 제한을 따릅니다. 예를 들어 Bedrock의 Anthropic Claude 2.x는 현재 us-east-1 및 us-west-2 리전에서 분당 500개의 요청과 500,000개의 토큰을 처리할 수 있습니다. 미세 조정되거나 지속적인 사전 훈련된 모델과 함께 솔루션을 사용할 수도 있습니다. 이러한 인스턴스의 경우 Amazon Bedrock은 프로비저닝된 처리량을 허용하므로 프로덕션 등급 애플리케이션에서 사용할 수 있도록 기본, 미세 조정되거나 지속적인 사전 훈련된 모델에 대해 대규모의 일관된 추론 워크로드를 실행할 수 있습니다.

Amazon Bedrock 콘솔 내에서 프로비저닝된 처리량을 구매하면 사용할 모델 ARN이 생성됩니다. 이제 모델 선택 단계의 생성형 AI Application Builder 마법사에서이 모델 ARN을 제공할 수 있습니다. 이렇게 하려면 Bedrock을 모델 공급자로 선택하고 Amazon Bedrock 콘솔에서이 프로비저닝된 모델 ARN을 생성하는 데 사용된 기본 모델 이름을 선택합니다. 그런 다음 온디맨드 모델과 프로비저닝된 모델 중에서 선택할 때 '프로비저닝된 모델'을 선택하고 모델 ARN을 제공합니다.

Depicts Deployment Wizard - Amazon Bedrock에 프로비저닝된 처리량 활성화



> **Note**
>
> 가드레일과 프로비저닝된 처리량은 배포 대시보드 및 사용 사례 스택과 동일한 리전에 있어야 합니다.

모델 파라미터

LLMs 구현과 관련된 다양한 파라미터를 수락하는 경우가 많습니다. 모델 공급자는 지원되는 파라미터 세트와 그 용도를 설명하는 문서를 제공하는 경우가 많습니다.

솔루션은 모델 파라미터를 기본 모델로 직접 전달하므로 파라미터가 올바르게 설정되었는지 확인하는 것이 중요합니다. 지원되는 파라미터에 대한 최신 정보는 모델 공급자의 설명서를 참조하세요.

에이전트 빌더 구성

Agent Builder는 프로덕션 지원 AI 에이전트를 생성하기 위한 포괄적인 구성 옵션을 제공합니다. 이 섹션에서는 Agent Builder 배포를 구성하고 관리하는 방법을 설명합니다.

시스템 프롬프트 구성

시스템 프롬프트는 에이전트의 동작, 성격 및 기능을 정의합니다. 시스템 프롬프트를 구성하려면:

1. Agent Builder 마법사에서 에이전트 구성 단계로 이동합니다.
2. 텍스트 편집기에서 시스템 프롬프트 템플릿을 편집합니다.
3. 다음에 대한 명확한 지침을 포함합니다.
   - 에이전트의 역할 및 목적
   - 사용 가능한 도구(MCP 서버) 사용 방법
   - 응답 형식 지정 기본 설정
   - 동작 지침
4. 필요한 경우 기본값으로 재설정 버튼을 사용하여 원래 템플릿을 복원합니다.

에이전트 프롬프트 모범 사례:

- 에이전트의 기능 및 제한 사항에 대해 구체적으로 설명합니다.
- 원하는 동작의 명확한 예를 제공합니다.
- 도구 사용 및 호출 시기에 대한 지침 포함
- 응답 형식 기대치 정의
- 에이전트 동작에 대한 경계 설정

MCP 서버 통합

모델 컨텍스트 프로토콜(MCP) 서버는 에이전트에게 엔터프라이즈 도구 및 데이터 소스에 대한 액세스 권한을 제공합니다. MCP 서버를 구성하려면:

1. 에이전트 구성 단계에서 MCP 서버 섹션을 찾습니다.
2. 드롭다운 메뉴에서 사용 가능한 MCP 서버 중에서 선택합니다.

> **Note**
>
> 에이전트 배포 전에 MCP 서버를 구성하고 액세스할 수 있어야 합니다. 에이전트는 구성된 MCP 서버에서 노출되는 도구를 자동으로 검색하고 사용합니다. 서버 설정 및 도구 구성은 MCP 설명서를 참조하세요.

메모리 설정

Agent Builder는 컨텍스트와 지식을 유지하기 위한 두 가지 유형의 메모리를 제공합니다.

단기 메모리

모든 에이전트에 대해 기본적으로 활성화됩니다.

- 세션 내에서 대화 컨텍스트를 유지합니다.
- 사용자 메시지 및 에이전트 응답을 자동으로 캡처합니다.
- 적절한 격리를 위해 actorId 및 sessionId에 의해 구성됨
- 구성 필요 없음

장기 메모리

세션 간에 인사이트를 저장하기 위한 선택적 기능:

1. 에이전트 구성 단계에서 메모리 구성 섹션을 찾습니다.
2. 토글 활성화할 장기 메모리를 활성화합니다.
3. 활성화되면 에이전트는 다음을 수행할 수 있습니다.
   - 대화 전반에 걸쳐 중요한 정보 추출 및 저장
   - 이전 세션에서 관련 컨텍스트 검색
   - 사용자 기본 설정 및 기록에 대한 지식 구축

> **Note**
>
> 장기 메모리는 의미 체계 메모리 전략 및 기본 보존 설정과 함께 AgentCore 메모리를 사용합니다.

Agent Builder 배포 모니터링

Agent Builder는 CloudWatch 대시보드 및 지표를 통해 포괄적인 모니터링을 제공합니다.

CloudWatch 대시보드 액세스

1. AWS 계정의 CloudWatch 콘솔로 이동합니다.
2. 왼쪽 탐색 창에서 대시보드를 선택합니다.
3. 라는 대시보드를 찾습니다`AgentBuilder-<UseCaseId>`.
4. 실시간 지표 및 과거 성능 데이터를 봅니다.

로그 액세스 및 분석

에이전트 로그는 CloudWatch Logs에서 사용할 수 있습니다.

1. AWS 콘솔에서 CloudWatch Logs로 이동합니다.
2. 접두사가 인 로그 그룹을 찾습니다`/aws/bedrock-agentcore/runtimes/`.
3. CloudWatch Insights를 사용하여 로그를 쿼리하고 분석합니다.
4. 특정 요청 IDs 또는 오류 패턴을 검색합니다.

워크플로 빌더 구성

Workflow Builder를 사용하면 특수 에이전트 빌더 에이전트에게 작업을 위임하는 감독자 에이전트를 통해 다중 에이전트 오케스트레이션이 가능합니다.

워크플로 생성

1. 배포 대시보드로 이동합니다.
2. 워크플로 사용 사례 생성을 선택합니다.
3. 감독자 에이전트를 구성합니다.
   - 이름: 워크플로의 설명 이름
   - 설명: 목적 및 기능
   - 시스템 프롬프트: 에이전트 위임 및 조정 지침
   - 모델: 감독자 에이전트를 위한 파운데이션 모델

감독자 프롬프트 모범 사례:

- 각 특수 에이전트를 사용해야 하는 경우를 명확하게 설명
- 여러 에이전트의 결과를 집계하기 위한 지침 포함
- 응답 형식 지정 기대치 정의
- 위임 동작의 경계 설정

에이전트 선택

특수 에이전트로 포함할 Agent Builder 에이전트를 선택합니다.

1. 워크플로 구성에서 에이전트 추가를 클릭합니다.
2. 사용 가능한 Agent Builder 에이전트 찾아보기 또는 검색
3. 에이전트 설명 검토
4. 워크플로에 포함할 에이전트 선택

에이전트 설명

감독자 에이전트는 에이전트 설명을 사용하여 위임할 에이전트를 결정합니다. 설명이 다음을 명확하게 설명하는지 확인합니다.

- 에이전트의 특수 도메인 또는 기능
- 에이전트가 처리하는 작업 유형
- 입력/출력 기대치

워크플로 테스트

배포 후:

1. 배포 대시보드를 통해 워크플로에 액세스
2. 여러 에이전트가 필요한 쿼리로 테스트
3. CloudWatch 로그에서 에이전트 위임 모니터링
4. 응답 품질 및 위임 패턴 검토
5. 위임이 최적이 아닌 경우 감독자 프롬프트 조정

모델 토큰 한도 관리를 위한 팁

참고: 솔루션은 다양한 LLMs에서 부과하는 토큰 제한을 직접 관리하려고 시도하지 않습니다. 프롬프트가 모델 공급자가 적용하는 사용 가능한 한도 내에 있는지 테스트하고 확인합니다.

프롬프트 크기를 제어하는 데 도움이 되도록 다음을 시도하세요.

1. 사용하려는 모델에서 부과하는 제한을 숙지합니다. 이러한 값은 모델마다 크게 다를 수 있으므로 시작하기 전에 사용 가능한 예산이 무엇인지 아는 것이 중요합니다.
2. 해당 예산을 염두에 두고 초기 프롬프트를 생성하고 프롬프트의 동적 요소에 대해 얼마나 절약할지 고려합니다. 예를 들어 사용자 입력, 채팅 기록, 문서 발췌문 등이 있습니다.
3. 프롬프트 구성 페이지에서 후행 기록 크기 제한을 설정하여 프롬프트에 포함된 대화 전환 수를 제한합니다.
4. 지식 기반 구성 마법사에서 문서 반환 한도를 설정합니다. 작업을 수행하기에 충분한 컨텍스트를 LLM에 제공하는 것 사이에 적절한 균형을 맞춰야 하지만, 토큰 제한을 초과하거나 지연 시간에 부정적인 영향을 미치기에는 그렇게 많지 않아야 합니다.
5. 일부 버퍼를 그대로 둡니다. 일반적인 사례에 맞게 예산을 책정하지 말고 긴 입력 쿼리, 큰 문서 발췌문 또는 긴 대화와 같은 엣지 사례에 대해 생각해 보고 실험해 보세요.

MCP 서버 Docker 이미지 빌드 단계

AWS에서 생성형 AI Application Builder와 함께 MCP(모델 컨텍스트 프로토콜) 서버를 사용하려면 첫 번째 단계로 프라이빗 Amazon ECR 리포지토리에 빌드되고 저장된 Docker 이미지가 필요합니다.

> **Note**
>
> 현재 Amazon Bedrock AgentCore 런타임에 배포된 기존 MCP 서버는 GAAB로 내보낼 수 없습니다. MCP 서버를 GAAB를 통해 생성된 에이전트에 연결하려면 GAAB를 통해 생성해야 합니다.

1단계: MCP 서버 생성

먼저 MCP 서버 구현을 준비해야 합니다. MCP 서버 생성에 대한 자세한 지침은 Amazon Bedrock AgentCore 개발자 안내서 - MCP 서버 생성을 참조하세요.

다음과 같은 프로젝트 구조를 권장합니다.

```
.
### __init__.py
### extras/
#   ### extra_dependencies.py
#   ### Dockerfile
### requirements.txt
### server.py <-- Server Entry point
```

Dockerfile 구조의 경우 다음 예와 유사한 형식을 사용하는 것이 좋습니다.

```
FROM ghcr.io/astral-sh/uv:python3.13-bookworm-slim
WORKDIR /app

# All environment variables in one layer
ENV UV_SYSTEM_PYTHON=1 \
    UV_COMPILE_BYTECODE=1 \
    UV_NO_PROGRESS=1 \
    PYTHONUNBUFFERED=1 \
    DOCKER_CONTAINER=1 \
    AWS_REGION=us-east-1 \
    AWS_DEFAULT_REGION=us-east-1

COPY requirements.txt requirements.txt
# Install from requirements file
RUN uv pip install -r requirements.txt

RUN uv pip install aws-opentelemetry-distro>=0.10.1

# Signal that this is running in Docker for host binding logic
ENV DOCKER_CONTAINER=1

# Create non-root user
RUN useradd -m -u 1000 bedrock_agentcore
USER bedrock_agentcore

EXPOSE 9000
EXPOSE 8000
EXPOSE 8080

# Copy entire project (respecting .dockerignore)
COPY . .

# Use the full module path
```

```
CMD ["opentelemetry-instrument", "python", "-m", "server"]
```

2단계: 로컬에서 MCP 서버 테스트

AWS에 배포하기 전에 MCP 서버를 로컬에서 테스트하여 예상대로 작동하는지 확인하는 것이 중요합니다. 로컬 테스트에 대한 자세한 지침은 Amazon Bedrock AgentCore 개발자 안내서 - 로컬에서 MCP 서버 테스트를 참조하세요.

3단계: Amazon ECR에 배포

MCP 서버가 로컬에서 생성 및 테스트되면 다음 단계에 따라 Amazon ECR에 배포합니다.

1. 최신 버전의 AWS CLI 및 Docker가 설치되어 있는지 확인합니다. 자세한 내용은 Amazon ECR 시작하기를 참조하세요.
2. 인증 토큰을 검색하고 Docker 클라이언트를 레지스트리에 인증합니다. AWS CLI를 사용합니다.

```
aws ecr get-login-password --region us-east-1 | docker login --username AWS --
password-stdin <account-id>.dkr.ecr.us-east-1.amazonaws.com
```

3. 다음 명령을 사용하여 Docker 이미지를 빌드합니다. Docker 파일을 처음부터 빌드하는 방법에 대한 자세한 내용은 Docker 설명서를 참조하세요. 이미지가 이미 빌드된 경우이 단계를 건너뛸 수 있습니다.

```
docker build -t <repository-name> .
```

4. 빌드가 완료되면 이미지를이 리포지토리로 푸시할 수 있도록 이미지에 태그를 지정합니다.

```
docker tag <repository-name>:latest <account-id>.dkr.ecr.us-east-1.amazonaws.com/
<repository-name>:latest
```

5. 다음 명령을 실행하여이 이미지를 새로 생성된 AWS 리포지토리로 푸시합니다.

```
docker push <account-id>.dkr.ecr.us-east-1.amazonaws.com/<repository-name>:latest
```

전체 배포 지침은 Amazon Bedrock AgentCore 개발자 안내서 - AWS에 MCP 서버 배포를 참조하세요.

4단계: GAAB에서 ECR URI 사용

Docker 이미지를 Amazon ECR에 성공적으로 푸시한 후 ECR 콘솔에서 이미지 URI를 복사합니다. AWS의 생성형 AI Application Builder 배포 마법사를 통해 MCP 서버를 배포할 때이 URI를 사용합니다.

다양한 MCP 게이트웨이 대상을 생성하는 단계

Amazon Bedrock AgentCore Gateway를 사용하면 기존 AWS 서비스 및 APIs를 에이전트가 사용할 수 있는 MCP 도구로 변환할 수 있습니다. 게이트웨이는 여러 대상 유형을 지원하므로 다양한 백엔드 서비스를 원활하게 통합할 수 있습니다.

지원되는 대상 유형은 다음과 같습니다.

- Lambda 대상: AWS Lambda 함수를 MCP 도구로 변환합니다. 자세한 지침은 Amazon Bedrock AgentCore 개발자 안내서 - Lambda 대상 추가를 참조하세요.
- OpenAPI 대상: OpenAPI 사양을 사용하여 REST APIs. 자세한 지침은 Amazon Bedrock AgentCore 개발자 안내서 - OpenAPI 스키마를 참조하세요.
- Smithy 대상: 유형 안전 API 통합을 위한 Smithy 모델 정의를 사용하여 MCP 도구를 빌드합니다. 자세한 지침은 Amazon Bedrock AgentCore 개발자 안내서 - Smithy 대상 구축을 참조하세요.
- MCP 서버 대상: URL 엔드포인트를 통해 외부 MCP 서버에 직접 연결하여 기존 MCP 서버를 통합할 수 있습니다. 자세한 지침은 Amazon Bedrock AgentCore 개발자 안내서 - MCP 서버 대상을 참조하세요.

MCP Gateway 대상 생성에 대한 추가 예제 및 자습서는 Amazon Bedrock AgentCore 샘플 리포지토리를 참조하세요.

지식 기반 구성

이 섹션에서는 솔루션에 대해 선택한 지식 기반에 데이터를 수집하는 방법을 설명합니다. 이 솔루션은 현재 RAG 기반 사용 사례 배포를 위한 지식 기반으로 Amazon Kendra 및 Amazon Bedrock 지식 기반을 지원합니다.

Amazon Kendra

Amazon Kendra를 지식 기반으로 사용하는 경우 다양한 데이터 소스 커넥터를 사용하여 다양한 소스에서 데이터를 수집하는 방법에 대한 자세한 내용은 Amazon Kendra 개발자 안내서를 참조하세요.

중요: 우발적인 데이터 손실을 방지하기 위해 배포 또는 스택이 삭제될 때 솔루션은 Kendra 인덱스(솔루션에서 생성하든 그렇지 않든)를 자동으로 삭제하지 않습니다. 지식 기반을 삭제하고 비용 발생을 중지하려면 보존되는 리소스와 정리 방법에 대한 자세한 내용은 수동 제거 섹션을 참조하세요.

Amazon Bedrock 지식 기반

Amazon Bedrock 지식 기반은 각각 데이터를 인덱싱할 수 있는 기능을 갖춘 다양한 벡터 스토어에서 지원할 수 있습니다. 지식 기반을 설정하고 채우려면 Amazon Bedrock 사용 설명서를 참조하세요. 특히 다음을 수행하려고 합니다.

- 먼저 데이터 소스 설정
- 그런 다음 지원되는 벡터 스토어에서 지식 기반에 대한 벡터 인덱스를 설정합니다. 지식 기반 생성 중에 Bedrock 콘솔에서 "새 벡터 스토어 빠른 생성" 옵션을 사용하는 경우이 작업을 건너뛸 수 있습니다.
- 마지막으로 지식 기반을 생성하고 구성된 데이터 소스를 동기화할 수 있습니다.

고급 지식 기반 설정

지식 기반 필터링 및 역할 기반 액세스 제어를 사용하는 RAG와 같은 고급 지식 기반 설정을 솔루션과 함께 사용할 수 있습니다. 지식 기반 필터링은 지식 기반 중 하나에 적용할 수 있지만 역할 기반 액세스 제어가 있는 RAG는 Amazon Kendra에서 특별히 사용할 수 있습니다.

지식 기반 필터링

이 솔루션을 사용하면 마법사 지식 기반 단계의 고급 RAG 구성 섹션에서 사용 사례를 배포할 때 Amazon Kendra 속성 필터 또는 Bedrock 지식 기반 검색 필터를 지정할 수 있습니다. https://docs.aws.amazon.com/bedrock/latest/userguide/kb-test-config.html 이러한 필터는 검색 전략, 쿼리인 기본 문서의 언어 등 지식 기반의 데이터 소스를 쿼리하는 방법을 정의합니다.

두 경우 모두 JSON 객체는 각 서비스 설명서에 지정된 형식(위에 연결됨)에 따라 필터 설정을 지정하는 데 사용됩니다.

예제 1: Kendra AttributeFilter

```
{
 "EqualsTo": {
 "Key": "_language_code",
 "Value": {
 "StringValue": "es"
```

```
 }
 }
}
```

예제 2: Bedrock RetrievalFilter

```
{
 "equals": {
 "key": "language",
 "value": "es"
 }
}
```

Amazon Kendra를 사용한 역할 기반 액세스 제어가 포함된 RAG

역할 기반 액세스 제어(RBAC)를 사용하면 Amazon Kendra 인덱스의 특정 문서에 액세스하거나 검색 결과에서 특정 문서를 볼 수 있는 사용자 또는 그룹을 제어할 수 있습니다. AWS의 생성형 AI Application Builder(GAAB) 사용 사례로 Amazon Kendra 인덱스 ID에 대한 RBAC를 구성하려면 다음 단계를 따르세요.

1: Amazon Kendra 인덱스 구성

1. Amazon Kendra 인덱스가 생성되고 여기에 하나 이상의 데이터 소스가 추가되었는지 확인합니다.
2. 사용자 그룹을 기반으로 데이터 소스에 대한 액세스 제어를 구성합니다. S3 데이터 소스의 경우 설명서의 지침에 따라 Amazon Cognito 사용자 풀에 생성된 것과 동일한 그룹 이름을 사용하여 액세스 제어 목록(ACLs)을 설정합니다. 이렇게 하면 사용자가 그룹 멤버십에 따라 볼 수 있는 권한이 부여된 문서 및 검색 결과에만 액세스할 수 있습니다.

> **Note**
>
> 생성한 Kendra 인덱스의 사용자 액세스 제어에서 토큰 기반 사용자 액세스 제어를 아니요로 둡니다. 2단계에서 역할 기반 액세스 제어를 활성화하면 AWS의 생성형 AI Application Builder가 사용자 인증 토큰에서 적절한 클레임을 추출하고 속성 필터를 생성합니다.

2. GAAB 배포 마법사를 사용하여 RAG 사용 사례 배포

1. 마법사의 4단계에 도달할 때까지 GAAB 배포 마법사의 화면 마법사 지침에 따라 RAG를 구성합니다.

2. 배포 마법사의 지식 기반 선택 단계에서 지식 기반 유형으로 Amazon Kendra를 선택합니다.
3. 기존 Amazon Kendra 인덱스가 있는지 또는 새 인덱스를 생성할지 지정합니다. 기존 인덱스가 있는 경우 사용자 그룹을 기반으로 액세스 제어 목록(ACLs)으로 구성된 Amazon Kendra 인덱스의 ID를 제공합니다.
4. 역할 기반 액세스 제어 옵션을 활성화합니다. 이 옵션을 사용하면 Amazon Kendra 인덱스에서 반환된 검색 결과가 사용자의 역할 및 그룹 권한을 기반으로 필터링됩니다.
5. 사용 사례를 검토하고 배포합니다.

3. Amazon Cognito 구성

1. GAAB 배포에 사용되는 Amazon Cognito 사용자 풀을 찾습니다. 이 Amazon Cognito 사용자 풀은 일반적으로 기본 배포 대시보드 CloudFormation 스택에서 생성됩니다.
2. Amazon Cognito 사용자 풀에서 새 사용자를 생성합니다. 사용자를 생성할 때 사용자가 이메일을 통해 임시 로그인 자격 증명을 받을 수 있도록 '이메일 초대 전송' 옵션을 선택합니다. 이렇게 하면 새 사용자가 GAAB 애플리케이션에 가입하고 액세스할 수 있습니다.
3. Amazon Cognito 사용자 풀에서 사용자 그룹을 생성합니다. 그룹 이름이 Amazon Kendra 인덱스 ACLs에 구성된 그룹과 정확히 일치하는지 확인합니다. 이는 RBAC를 활성화하는 데 매우 중요합니다. 사용자의 그룹 멤버십에 따라 액세스할 수 있는 검색 결과가 결정되기 때문입니다.
4. 역할 및 액세스 권한에 따라 사용자를 적절한 그룹에 할당합니다. 사용자는 Amazon Kendra 인덱스 ACL에 필요한 그룹과 GAAB 배포 중에 생성된 사용 사례별 그룹 모두에 추가되어야 합니다. 이를 통해 사용자는 특정 사용 사례 및 관련 검색 결과에 액세스하는 데 필요한 권한을 갖게 됩니다.

이 단계를 따르면 GAAB 배포에 대한 역할 기반 액세스 제어(RBAC)를 구성하여 사용자가 할당된 사용자 그룹 및 권한에 따라 권한이 부여된 정보 및 기능에만 액세스하고 상호 작용할 수 있도록 할 수 있습니다.

> **Note**
>
> 현재 Amazon Kendra만 AWS의 생성형 AI 애플리케이션 빌더의 지식 기반에 대한 RBAC를 지원합니다. Amazon Bedrock 지식 기반에서는 RBAC가 지원되지 않지만 메타데이터 필터를 사용하여 일정 수준의 필터링을 달성할 수 있습니다. 자세한 내용은 Amazon Bedrock 사용 설명서를 참조하세요.

프롬프트 구성

배포 대시보드 마법사에는 사용자와 AI 모델 간의 상호 작용을 안내하는 프롬프트 환경과 템플릿을 사용자 지정할 수 있는 프롬프트 구성 단계가 있습니다. AI 어시스턴트로부터 정확하고 관련 있는 응답을 얻으려면 이러한 설정을 올바르게 구성하는 것이 중요합니다.

이 섹션에서는 AI 프롬프트의 전반적인 경험과 동작을 제어합니다.

- 최대 프롬프트 템플릿 길이:이 설정은 프롬프트 템플릿의 최대 길이(자)를 결정합니다. 값이 높을수록 AI 모델에 더 많은 컨텍스트를 제공할 수 있으므로 응답이 더 정확할 수 있습니다. 그러나 프롬프트가 너무 길면 노이즈가 발생하여 성능에 부정적인 영향을 미칠 수도 있습니다. Amazon Bedrock 모델의 경우 최대 프롬프트 템플릿 길이(문자)의 기본값은 기본 모델 토큰 제한을 사용하여 계산됩니다. Bedrock 내에서 모델 이름을 편집하고 변경하면 '기본값으로 재설정' 버튼이 강조 표시되어 새로 선택한 모델의 기본값을 채택하는 데 사용할 수 있습니다. Amazon SageMaker AI 모델의 경우 적절한 기본값이 제공되지만 기본 모델을 확인하고 이에 따라 이러한 최대 프롬프트 템플릿 길이와 입력 텍스트 길이를 선택하는 것이 좋습니다. 자세한 내용은 모델 토큰 한도 관리에 대한 팁 섹션을 참조하세요.
- 최대 입력 텍스트 길이:이 설정은 사용자 입력 텍스트의 최대 길이(자)를 제한합니다. 입력이 길수록 관련 없는 정보가 포함되어 AI 모델에서 관련 없거나 부정확한 응답을 얻을 위험이 커질 수 있습니다.
- 사용자 프롬프트 편집:이 옵션을 사용하면 사용자가 채팅 UI를 통해 프롬프트 템플릿을 수정할 수 있는 기능을 활성화하거나 비활성화할 수 있습니다. 이 기능을 비활성화하면 일관성을 유지하고 프롬프트에 의도하지 않은 변경 사항을 방지하는 데 도움이 될 수 있습니다.

프롬프트 템플릿

이 섹션에서는 AI 모델에서 사용할 실제 프롬프트 템플릿을 정의할 수 있습니다. 프롬프트 템플릿은 일반적으로 사용자의 입력, 참조 구절 및 채팅 기록과 같은 다양한 구성 요소에 대한 자리 표시자를 포함하는 구조를 따릅니다.

- 프롬프트 템플릿: 원하는 프롬프트 템플릿을 작성하거나 붙여 넣을 수 있는 기본 텍스트 영역입니다. 템플릿은 AI 모델에 필요한 컨텍스트와 지침을 제공하도록 만들어야 합니다. 여기에는 일반적으로 다음과 같은 자리 표시자가 포함됩니다.
  - `{input}`:이 자리 표시자는 Sagemaker AI 배포에 필수이며 사용자의 입력 또는 쿼리로 대체됩니다.
  - `{history}`:이 자리 표시자는 Sagemaker AI 배포에 필수이며 현재 대화의 채팅 기록으로 대체됩니다.

  - `{context}`:이 자리 표시자는 RAG 배포에 필수이며 구성된 지식 기반에서 가져온 문서 발췌문으로 대체됩니다.
- 질문 재구분:이 옵션(RAG 배포에만 사용 가능)은 AI 모델로 전달되기 전에 사용자의 원래 입력 쿼리를 재구분할지 아니면 모호하게 할지를 결정합니다. 쿼리를 다시 작성하면 모델이 사용자의 의도를 더 잘 이해하는 데 도움이 되어 잠재적으로 응답이 더 정확해질 수 있습니다.

프롬프트 템플릿과 환경을 구성할 때는 노이즈 또는 성능 문제를 일으킬 수 있는 너무 길거나 관련이 없는 정보를 피하면서 AI 모델에 충분한 컨텍스트와 지침을 제공하는 것 사이의 균형을 맞추는 것이 중요합니다.

고급 프롬프트 설정

이 섹션에서는 AI 모델에 대화 기록이 표시되는 방식을 제어할 수 있습니다.

- 후행 기록 크기:이 설정은 최종 프롬프트에 포함되어야 하는 이전 메시지 수를 결정합니다. 이 값을 0으로 설정하면 프롬프트 템플릿 또는 모호하지 않은 프롬프트 템플릿에 기록이 삽입되지 않습니다. 참고: 0으로 설정하더라도 프롬프트 템플릿에 {history} 자리 표시자가 있어야 합니다. 런타임 시 빈 문자열로 대체됩니다.
  - 참고:이 값에는 짝수를 제공하는 것이 좋습니다. 홀수를 제공하면 페어링된 상호 작용의 AI 응답만 반환됩니다.
- 인적 접두사: 대화 기록에서 사용자가 보낸 메시지를 식별하는 데 사용되는 접두사입니다.
- AI 접두사: 대화 기록에서 AI 모델이 반환한 메시지를 식별하는 데 사용되는 접두사입니다.

모호하지 않은 프롬프트 구성

이 섹션에서는 구성된 지식 기반에 사용자 입력을 보내기 전에 사용자 입력을 명확하게 구분하기 위한 동작과 템플릿을 구성할 수 있습니다.

- 모호화 해제 활성화:이 옵션은 지식 기반에 보내기 전에 사용자 입력을 모호화해야 하는지 여부를 결정합니다.
- 모호화 해제 프롬프트 템플릿: 지식 기반에 연결할 때 사용자 입력을 모호화하는 데 사용되는 프롬프트 템플릿입니다. 이 프롬프트에서 생성된 출력은 지식 기반에 전송된 쿼리로 사용됩니다. 모호성을 비활성화하면 사용자의 원시 쿼리가 지식 기반에 변경 없이 전송됩니다.

예를 들어 명확성을 활성화한 경우 후속 사용자 쿼리는 "어떤 비용이 드나요?"입니다. 는 "번호판을 갱신하는 데 드는 비용은 얼마입니까?"라는 명확성을 잃어 더 나은 검색 쿼리로 이어질 수 있습니다.

배포된 텍스트 사용 사례 사용

텍스트 사용 사례에 내장된 UI는 비즈니스 사용자가 관리자 사용자가 생성한 배포를 빠르게 탐색하고 실험할 수 있도록 하기 위한 것입니다. 비즈니스 사용자가 변경한 구성은 세션에만 적용됩니다. 비즈니스 사용자는 이러한 변경 사항을 관리자 사용자와 공유해야 합니다. 관리자 사용자는 모든 사용자가 사용할 수 있도록 이러한 변경 사항으로 기본 배포를 업데이트할 수 있습니다.

채팅 UI는 다음 구성 요소로 구성됩니다.

- 채팅 창
- 채팅 입력 상자
- Settings
- 대화 지우기

채팅 창

대화의 다양한 턴을 유지합니다. 오른쪽에서 시작하는 메시지는 비즈니스 사용자의 메시지이고 왼쪽에서 시작하는 메시지는 구성된 LLM의 메시지입니다. 응답을 쉽게 복사할 수 있도록 모든 LLM 응답에 작은 클립보드 아이콘이 있습니다.

채팅 입력 상자

채팅 창 하단에는 채팅 입력 상자가 고정되어 있습니다. 여기서 비즈니스 사용자는 메시지를 입력하여 LLM으로 전송할 수 있습니다. 입력 상자 바로 위에 연결 상태가 있습니다. 연결이 끊어지면(예: 비활성으로 인해) 다음에 채팅 메시지가 전송될 때 새 연결이 자동으로 생성됩니다. 이 요청은 추가 WebSocket 연결 시간으로 인해 약간 더 오래 걸릴 것으로 예상됩니다.

특정 구성에 따라 입력에 적용되는 최대 길이가 있을 수 있습니다. 이 제한을 초과하면 사용자에게 알림이 수신되고 메시지가 전송되지 않습니다.

참고: Amazon Kendra에서 RAG를 사용하는 경우 Retrieve API는 쿼리를 30개의 토큰 단어로 자릅니다. 사용자 입력이 더 길어질 것으로 예상되는 경우 검색 성능에 어떤 영향을 미칠 수 있는지 평가합니다.

Settings

비즈니스 사용자가 다양한 구성을 빠르게 실험할 수 있도록 설정 패널을 사용할 수 있어 특정 배포 구성 옵션을 on-the-fly 편집할 수 있습니다.

(예: 프롬프트 템플릿). 이러한 변경은 새 세션이 시작될 때만 수행할 수 있습니다. 대화가 시작되면 대화를 지우면 구성 설정 편집이 다시 활성화됩니다.

참고: 관리자 사용자는 배포 설정을 잠그도록 선택할 수 있습니다. 프롬프트 단계에서 마법사를 통해 배포 시 라이브 편집을 방지할 수 있습니다.

대화 지우기

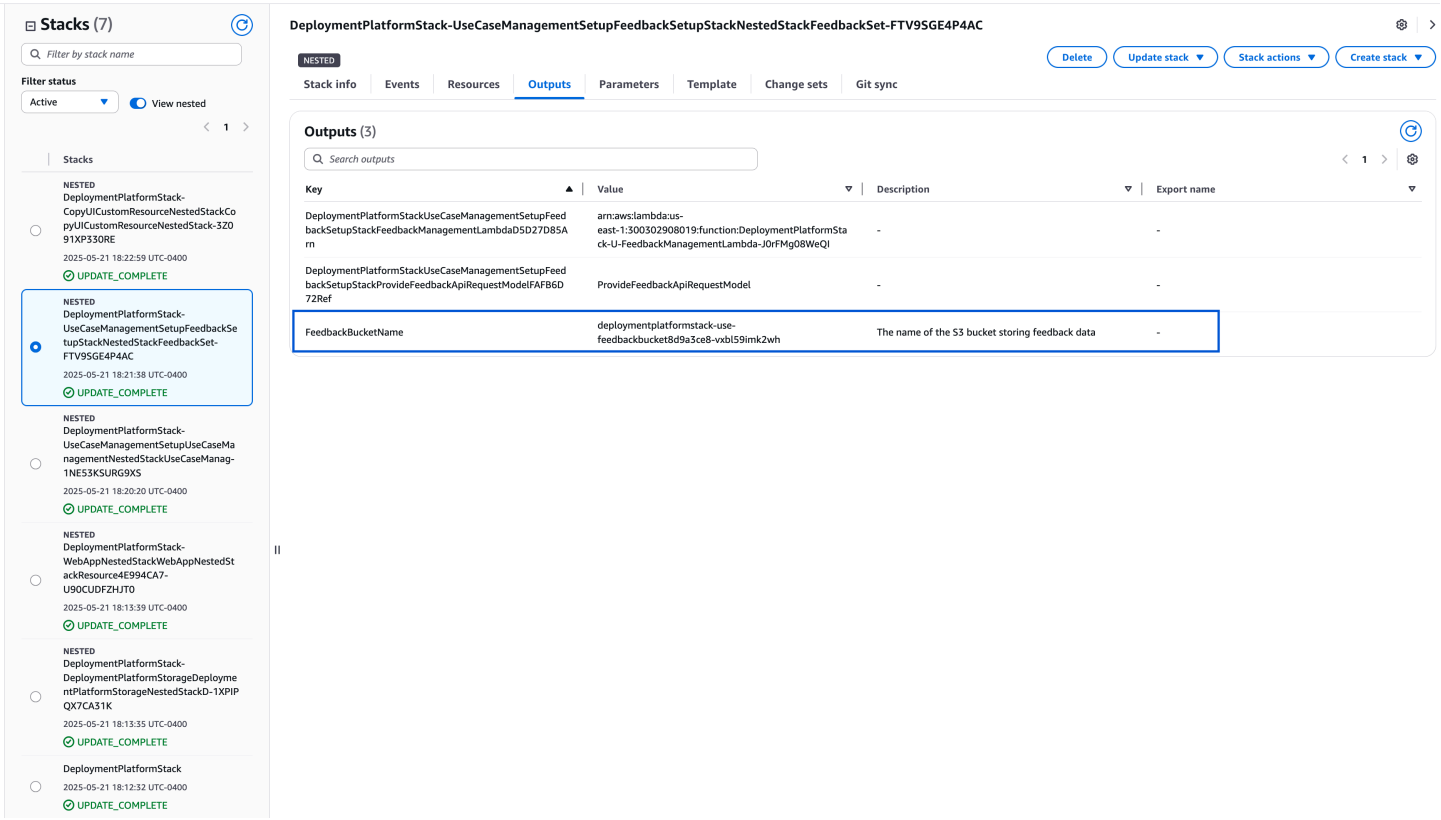
대화 과정에서 솔루션은 채팅 기록을 유지하여 대화 경험을 가능하게 합니다. 이렇게 하면 쿼리 모호화 및 후속 질문이 활성화됩니다. 대화를 재설정하고이 상호작용에 대한 모든 채팅 기록을 삭제하려면 채팅 창 상단에서 *대화 지우기 *를 선택합니다. 대화가 지워지면 설정을 다시 편집할 수 있는 새 세션이 생성됩니다.

사용자 수집 피드백 액세스 및 분석

v3.0.0부터 배포 대시보드는 중첩된 피드백 스택을 배포하여 대시보드와 함께 배포된 텍스트 및 Bedrock 에이전트 사용 사례가 LLM/에이전트가 생성하는 응답에 대한 피드백 수집 기능을 갖도록 합니다. 특히 사용자는 선택적 설명과 함께 긍정적 또는 부정적 피드백을 제공할 수 있습니다. 사용자가 부정적인 피드백을 제공하는 경우 '부정확', '미완료 또는 부족', '유해' 및/또는 '기타'와 같은 부정적인 범주 중 하나를 추가로 선택할 수 있습니다.

사용자가 피드백을 제공하면 피드백은 사용 사례 ID, 연도 및 월로 분할된 S3 버킷에 저장됩니다. 사용 사례 ID는 배포 대시보드에서 찾을 수 있으며 피드백 S3 버킷은 배포 대시보드 스택의 피드백 중첩 스택 출력에서 찾을 수 있습니다.

배포 스택 그림 - 피드백 버킷 이름 찾기



사용자 피드백은 최소한의 정보가 포함된 API 요청으로 전송됩니다.

```
{
  "useCaseRecordKey": "a1b2c3d4-e5f6g7h8",
  "conversationId": "12345678-1234-1234-1234-123456789012",
  "messageId": "87654321-4321-4321-4321-210987654321",
  "rephrasedQuery": "What are the key features of the Generative AI Application Builder
 on AWS?",
  "sourceDocuments": [
    "s3://bucket-name/document1.pdf",
    "s3://bucket-name/document2.pdf"
  ],
  "feedback": "positive",
  "feedbackReason": [
    "Incomplete or insufficient"
  ],
  "comment": "The response was helpful but could include more details about important
 features."
}
```

그런 다음이 페이로드는 배포 시 사용 사례의 올바른 구성을 useCaseRecordKey 식별하는를 사용하여 Lambda에 의해 처리됩니다. 이 구성은 실제 userInput 및를 검색하는 데 추가로 사용되는 ConversationTable 이름(모든 대화 및 인적 및 AI 메시지 시퀀스 포함)과 같은 피드백에 대한 특정 세부 정보를 가져오는 데 사용됩니다llmResponse. Bedrock Agent 사용 사례의 agentAliasId 경우 agentId 및 ,이 구성을 사용하는 텍스트 사용 사례의 경우 modelProviderbedrockModelId, 등과 같은 추가 세부 정보도이 피드백 레코드에 연결됩니다. 이 구성에 액세스하는 방법에 대한 자세한 내용은 아래의 사용자 지정 피드백 매핑 섹션을 참조하세요. 수신되는 각 피드백 요청은 JSON 객체로 저장되며 텍스트 사용 사례의 경우 샘플 피드백 레코드는 다음과 같을 수 있습니다.


```
{
   "useCaseId": "12345678-1234-1234-1234-123456789012",
   "useCaseRecordKey": "c07a2e3b-2f31b1e0",
   "userId": "22345678-1234-1234-1234-123456789012",
   "conversationId": "dd51de5d-5af1-4ec6-91d2-aadf14352109",
   "messageId": "32345678-1234-1234-1234-123456789012",
   "userInput": "What are its key features?",
   "rephrasedQuery": "What are the key features of the Generative AI Application
 Builder on AWS?",
   "llmResponse": "Generative AI Application Builder on AWS can help you build
 production ready enterprise chatbots rapidly.",
   "feedback": "negative",
   "feedbackReason": [
      "Incomplete or insufficient"
   ],
   "comment": "The response was helpful but could include more details about important
 features.",
   "timestamp": "2025-05-22T18:48:08.340Z",
   "feedbackId": "42345678-1234-1234-1234-123456789012",
   "useCaseType": "Text",
   "modelProvider": "Bedrock",
   "bedrockModelId": "amazon.nova-lite-v1:0",
   "ragEnabled": "false"
}
```


또는 Bedrock Agent 사용 사례의 경우 다음과 같습니다.


```
{
   "useCaseId": "12345678-1234-1234-1234-123456789012",
   "useCaseRecordKey": "c07a2e3b-2f31b1e0",
   "userId": "22345678-1234-1234-1234-123456789012",
   "conversationId": "dd51de5d-5af1-4ec6-91d2-aadf14352109",
   "messageId": "32345678-1234-1234-1234-123456789012",
```

```
   "userInput": "What are its key features?",
   "llmResponse": "Generative AI Application Builder on AWS can help you build
 production ready enterprise chatbots rapidly.",
   "feedback": "negative",
   "feedbackReason": [
      "Incomplete or insufficient"
   ],
   "comment": "The response was helpful but could include more details about important
 features.",
   "timestamp": "2025-05-22T18:48:08.340Z",
   "feedbackId": "42345678-1234-1234-1234-123456789012",
   "useCaseType": "Agent",
   "agentId": "AHFXUJCAK1",
   "agentAliasId": "KSEDKOS0BL"
}
```

그런 다음이 피드백을 추가 처리, 분석 및 모델 재훈련/피드백 루프에 사용할 수 있습니다. 사용자 지정 매핑을 추가하여 피드백 lambda에 저장되는 피드백 레코드를 개선할 수도 있습니다.

사용자 지정 피드백 매핑

배포 대시보드에는 키를 사용하여 배포 대시보드 스택의 스택 출력에서 찾을 수 LLMConfigTable 있는가 포함되어 있습니다LLMConfigTableName. LLMConfigTable에는 배포 대시보드 마법사를 통해 사용 사례를 배포하는 동안 관리자가 선택한 설정에 따라 각 사용 사례에 대한 구성이 포함되어 있습니다. 각 사용 사례 구성은 로 식별됩니다useCaseRecordKey. 다음은의 샘플 사용 사례 구성 레코드입니다LLMConfigTable.

```
{
   "key": "2dd76cfa-bc1a14da",
   "config": {
      "ConversationMemoryParams": {
        ...
      },
      "FeedbackParams": {
         "CustomMappings": {
            "NumberOfDocs": "$.KnowledgeBaseParams.NumberOfDocs",
            "ScoreThreshold": "$.KnowledgeBaseParams.ScoreThreshold"
         },
         "FeedbackEnabled": true
      },
      "IsInternalUser": "true",
      "KnowledgeBaseParams": {
```

```
         "KendraKnowledgeBaseParams": {
            "ExistingKendraIndexId": "d2831033-667f-4539-ab28-e6c7c7c5988b",
            "RoleBasedAccessControlEnabled": false
         },
         "KnowledgeBaseType": "Kendra",
         "NumberOfDocs": 5,
         "ReturnSourceDocs": false,
         "ScoreThreshold": 0.3
      },
      "LlmParams": {
         "BedrockLlmParams": {
            "BedrockInferenceType": "QUICK_START",
            "ModelId": "amazon.nova-lite-v1:0"
         },
         "ModelParams": {},
         "ModelProvider": "Bedrock",
         "PromptParams": {
            ...
         },
         "RAGEnabled": true,
         "Streaming": false,
         "Temperature": 0.1,
         "Verbose": false
      },
      "UseCaseName": "test-rag-usecase",
      "UseCaseType": "Text"
   }
}
```

사용 사례에 대해 피드백이 활성화된 경우이 구성에는 FeedbackParams 모든 추가 필드의 JSONPaths를 피드백 S3 버킷에 저장된 피드백 JSON 레코드에 추가할 수 있는 CustomMappings 객체가 포함됩니다. 예를 들어 위의 샘플 사용 사례 구성의 경우 CustomMappings에는 ScoreThreshold JSONPaths의 루트config로 로 시작하는 CustomMappings 객체에 NumberOfDocs 및 JSONPaths가 추가로 포함되어 있습니다. 이 구성을 사용하면 피드백 S3 버킷에 저장된 각 JSON 레코드가 이미 제공된 필드를 제외하고 이러한 2개의 추가 값을 가져오기 시작합니다.

피드백 데이터 분석

피드백 데이터는 S3에 JSON 객체로 저장됩니다. 다음은이 피드백 데이터에 대한 액세스와 실행 가능성을 높이기 위한 몇 가지 접근 방식입니다.

AWS Glue 및 Amazon Athena 사용

AWS Glue와 Amazon Athena는 피드백 데이터를 카탈로그화, 쿼리 및 분석할 수 있는 서버리스 방법을 제공합니다.

AWS Glue를 사용하면 S3 버킷의 데이터를 검사하고, 스키마를 추론하고, 카탈로그에 모든 관련 메타데이터를 기록하는 AWS Glue 크롤러를 생성할 수 있습니다. 그런 다음 Amazon Athena와 같은 서비스를 사용하여 데이터를 쿼리할 수 있습니다.

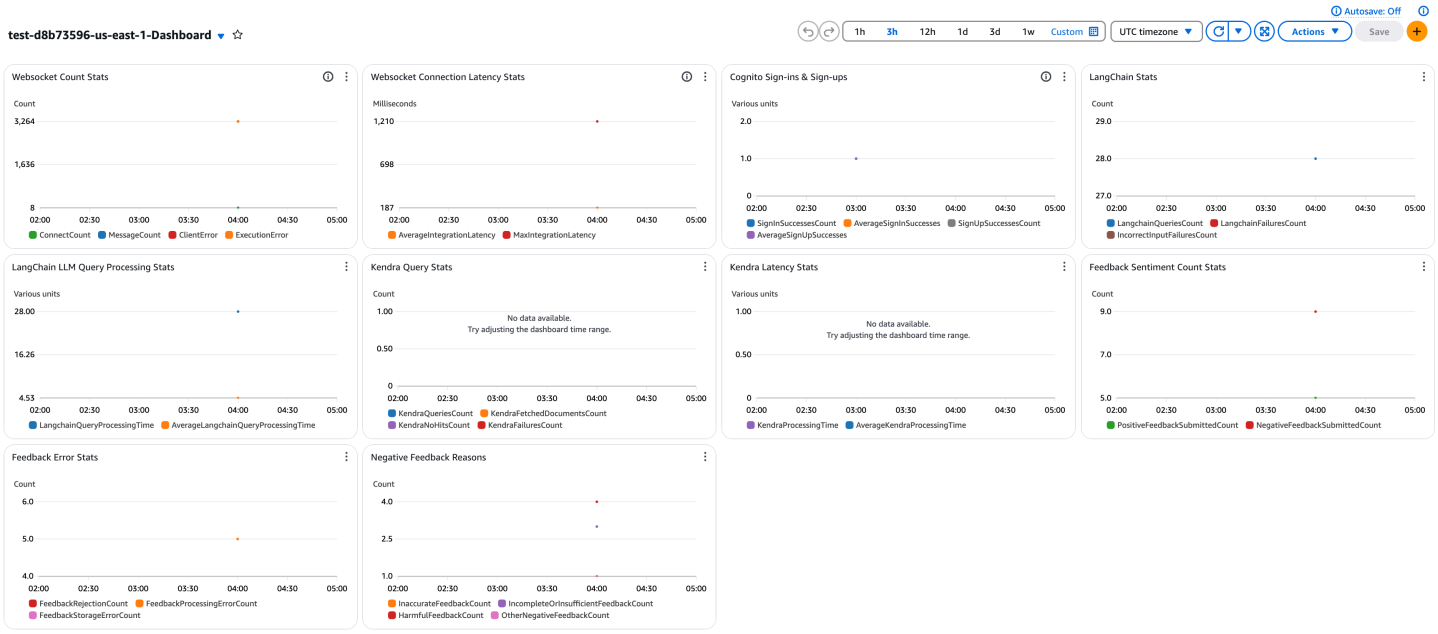
AWS Glue 데이터 카탈로그를 사용하여 피드백 S3 버킷을 Amazon Athena와 연결하는 단계는 AWS Athena 설명서를 참조할 수 있습니다. S3 Amazon Athena AWS Glue 또한 Glue의 보다 강력한 기능 중 일부를 사용하여이 데이터에 대해 ETL(Extract Transform & Load) 작업을 수행하고 이를 분석 또는 모델 재훈련 사용 사례에 적합한 형식으로 변환할 수 있습니다. Glue를 사용하면 특정 피드백 유형으로 레코드를 필터링하고, 누락된 정보를 채우고,이 데이터를 다른 S3 버킷 또는 다른 AWS 데이터 스토어와 같은 다른 스토리지 위치에 로드하는 등의 작업을 수행할 수 있습니다.

> **Note**
>
> 사용 사례에 따라 피드백 데이터가 희소할 수 있으므로 비용을 최적화하기 위해 매일 밤이 아닌 주기적으로(예: 매주) 실행되도록 Glue 크롤러를 예약하는 것이 좋습니다.

솔루션의 CloudWatch 대시보드 사용

또한 사용 사례별로 긍정적 및 부정적 피드백, 부정적 피드백 이유 범주 등에 대한 추세를 제공할 수 있는 솔루션과 함께 패키징된 CloudWatch 대시보드에 액세스할 수 있습니다. AWS CloudWatch 콘솔의 대시보드에서 사용 사례 이름을 사용하여이 대시보드를 찾을 수 있습니다.

사용 사례 CloudWatch 대시보드를 보여줍니다.



이 대시보드에서 추가 위젯을 빌드하거나 Amazon Quick Sight 대시보드를 생성할 수도 있습니다.

피드백 데이터 분석 모범 사례

- S3 버킷에 데이터 수명 주기 정책을 구현하여 이전 피드백 데이터를 저렴한 스토리지 계층에 아카이브
- 각 사용 사례에 대해 별도의 분석을 생성하여 모델별 개선 기회를 식별합니다.
- 부정적인 피드백이 허용 수준을 초과할 때 알림을 트리거하는 피드백 임계값 설정
- 이해관계자 및 모델 개선 팀과 공유하기 위해 중요한 인사이트를 주기적으로 내보내기

배포에 대한 작업 지표 보기

배포 대시보드 및 사용 사례 스택에는 솔루션의 다양한 운영 지표를 추적하는 자체 CloudWatch 대시보드가 함께 제공됩니다. 이러한 CloudWatch 대시보드를 사용하여 다양한 배포를 비교할 수 있습니다. 대시보드에 액세스하려면:

1. CloudWatch 콘솔로 이동합니다.
2. 스택 이름 또는 범용 고유 식별자(UUID)를 조회하여 사전 구축된 대시보드를 검색합니다.

예를 들어 텍스트 사용 사례에는 WebSocket 연결 수, 사용자 로그인 및 가입 수, LLM이 완료를 처리하는 데 걸린 시간 등을 추적하는 그래프가 함께 제공됩니다. 고객은 이러한 그래프를 사용하여 배포의 다양한 _quantitative _metrics를 비교할 수 있습니다.

Example

다양한 사용 사례에 적용된 다양한 모델의 정성적 결과를 비교하기는 어렵습니다. 복제 기능을 사용하면 출력을 나란히 비교할 수 있도록 여러 배포를 빠르게 실행할 수 있습니다.

CloudWatch Logs 인사이트 액세스

이 솔루션은 Lambda 함수에 대한 오류, 경고, 정보 및 디버깅 메시지를 기록합니다. 로깅할 메시지 유형을 선택하려면:

1. AWS Lambda 콘솔에서 해당 함수를 찾습니다.
2. POWERTOOLS_LOG_LEVEL 환경 변수를 추가합니다.
3. 변수를 해당 메시지 유형으로 설정합니다.

자세한 지침은 AWS Lambda Lambda 개발자 안내서의 Lambda 환경 변수 생성을 참조하세요.

다음 표에는 선택할 수 있는 로그 수준 유형이 나열되어 있습니다.

수준	설명
오류	로그에는 작업이 실패하는 모든 항목에 대한 정보가 포함됩니다.
경고	로그에는 함수에서 불일치를 일으킬 수 있지만 반드시 작업이 실패하는 것은 아닌 모든 항목에 대한 정보가 포함됩니다. 로그에는 ERROR 메시지도 포함됩니다.
INFO	로그에는 함수의 작동 방식에 대한 상위 수준 정보가 포함됩니다. 로그에는 ERROR 및 WARNING 메시지도 포함됩니다.

수준	설명
DEBUG	로그에는 함수 문제를 디버깅할 때 유용할 수 있는 정보가 포함됩니다. 로그에는 ERROR, WARNING 및 INFO 메시지도 포함됩니다.

다음 절차에 따라이 솔루션에 CloudWatch Logs 인사이트를 추가합니다.

1. 관련 로그 그룹을 식별합니다.
   a. AWS CloudFormation 콘솔에 로그인합니다.
   b. 대상 스택을 선택합니다.
   c. 리소스 탭을 선택하고 대상 Lambda 함수를 검색합니다.
   d. AWS Lambda 콘솔에 로그인하고 각 대상 Lambda 함수를 선택합니다.
   e. 각 대상 Lambda 함수에 대해 모니터링 탭을 선택하고 CloudWatch Logs 보기를 선택합니다.
   f. 인사이트를 추출하려는 로그 그룹의 이름을 복사합니다.
2. Amazon CloudWatch 콘솔로 이동합니다.
3. 탐색 메뉴의 로그에서 Logs Insights를 선택합니다.
4. Logs Insights 페이지에서 로그 탭을 선택합니다.
5. 1단계에서 로그 그룹 이름을 검색합니다.
6. 다음 예제 쿼리 중 하나를 복사하여 쿼리 필드에 붙여 넣습니다.
   a. 모든 클라이언트 예외를 식별하려면:

```
fields @message
|filter @message like /(?i)Exception/|stats count(*) as exceptionCount by @message
```

   b. 함수 이름별로 호출 수를 검색하려면:

```
stats count(*) by function_name
```

   c. 5분 간격으로 호출 수를 검색하려면:

```
stats count(*) as invocations by bin(5m)
```

   d. 모든 AWS X-Ray 트레이IDs를 검색하려면:

```
filter @message like "XRAY TraceId"
|parse @message "XRAY TraceId: * " as traceId|stats count(*) by traceId
```

e. 특정 X-Ray 트레이스 ID와 관련된 로그를 검색하려면:

```
filter @message like "your-traceid-here"
```

f. 승인되지 않은 WebSocket 오류를 검색하려면:

```
fields
@ingestionTime,
@log,
@logStream,
@message,
@requestId,
@timestamp,
errorMessage,
errorType
|filter @message like /Unauthorized/ and @message like /websocket/|sort @timestamp
 desc
```

g. 게시된 지표 수를 검색하려면:

```
filter @message like "CloudWatchMetrics"
|parse @message /"Metrics":\s*\[(?<metrics>.*?)\]/|stats count(*) as metric_count
 by metrics
```

개발자 안내서

이 섹션에서는 솔루션의 소스 코드, 통합 가이드, 사용자 지정 가이드 및 API 참조를 제공합니다.

소스 코드

이 솔루션의 소스 파일을 다운로드하고 사용자 지정 사항을 다른 사람과 공유하려면 GitHub 리포지토리를 방문하세요.

AWS 템플릿의 생성형 AI Application Builder는 AWS 클라우드 개발 키트(AWS CDK)를 사용하여 생성됩니다. 자세한 내용은 README.md 파일을 참조하세요.

통합 가이드

전체 솔루션은 쉽게 확장할 수 있도록 설계되었습니다. 이 솔루션의 오케스트레이션 계층은 LangChain을 사용하여 빌드됩니다. LangChain(또는 이러한 구성 요소에 LangChain 커넥터를 제공하는 타사)에서 지원하는 모든 모델 공급자, 지식 기반 또는 대화 메모리 유형을이 솔루션에 추가할 수 있습니다.

지원되는 LLMs 확장

사용자 지정 LLM 공급자와 같은 다른 모델 공급자를 추가하려면 솔루션의 다음 세 가지 구성 요소를 업데이트해야 합니다.

1. 사용자 지정 LLM 공급자로 구성된 채팅 애플리케이션을 배포하는 새 `TextUseCase` CDK 스택을 생성합니다.
   a. 이 솔루션의 GitHub 리포지토리를 복제하고 README.md 파일에 제공된 지침에 따라 빌드 환경을 설정합니다.
   b. `source/infrastructure/lib/bedrock-chat-stack.ts` 파일을 복사(또는 새로 생성)하고 동일한 디렉터리에 붙여넣은 다음 이름을 로 바꿉니다`custom-chat-stack.ts`.
   c. 파일의 클래스 이름을와 같은 적절한 클래스로 바꿉니다`CustomLLMChat`.
   d. 사용자 지정 LLM의 자격 증명을 저장하는 Secrets Manager 암호를이 스택에 추가하도록 선택할 수 있습니다. 다음 단락에서 설명하는 채팅 Lambda 계층에서 모델 호출 중에 이러한 자격 증명을 검색할 수 있습니다.
2. 추가할 모델 공급자의 Python 라이브러리가 포함된 Lambda 계층을 빌드하고 연결합니다. Amazon Bedrock 사용 사례 채팅 애플리케이션의 경우 `langchain-aws` Python 라이브러리에는 AWS 모델

공급자(Amazon Bedrock 및 SageMaker AI), 지식 기반(Amazon Kendra 및 Amazon Bedrock 지식 기반) 및 메모리 유형(예: DynamoDB)에 연결하기 위한 LangChain 패키지 위에 사용자 지정 커넥터가 포함되어 있습니다. 마찬가지로 다른 모델 공급자에도 자체 커넥터가 있습니다. 이 계층을 사용하면이 모델 공급자의 Python 라이브러리를 연결하여 LLM을 호출하는 채팅 Lambda 계층에서 이러한 커넥터를 사용할 수 있습니다(3단계). 이 솔루션에서는 사용자 지정 자산 번들러를 사용하여 CDK 측면을 사용하여 연결된 Lambda 계층을 빌드합니다. 사용자 지정 모델 공급자 라이브러리에 대한 새 계층을 생성하려면:

a. `source/infrastructure/lib/utils/lambda-aspects.ts` 파일의 `LambdaAspects` 클래스로 이동합니다.

b. 파일에 제공된 Lambda 측면 클래스의 기능을 확장하는 방법에 대한 지침을 따릅니다(예: `getOrCreateLangchainLayer` 메서드 추가). 이 새 메서드(예: `getOrCreateCustomLLMLayer`)를 사용하려면 `source/infrastructure/lib/utils/constants.ts` 파일의 `LLM_LIBRARY_LAYER_TYPES` 열거형도 업데이트합니다.

3. `chat` Lambda 함수를 확장하여 새 공급자에 대한 빌더, 클라이언트 및 핸들러를 구현합니다.

에는 이러한 LLM을 빌드하기 위한 지원 클래스와 함께 다양한 LLM에 대한 LangChain 연결이 `source/lambda/chat` 포함되어 있습니다. LLMs 이러한 지원 클래스는 Builder 및 객체 지향 설계 패턴을 따라 LLM을 생성합니다.

각 핸들러(예: `bedrock_handler.py`)는 먼저 클라이언트를 생성하고 환경에 필요한 환경 변수가 있는지 확인한 다음 `get_model` 메서드를 호출하여 LangChain LLM 클래스를 가져옵니다. 그런 다음 생성 메서드를 호출하여 LLM을 호출하고 응답을 가져옵니다. LangChain은 현재 Amazon Bedrock용 스트리밍 기능을 지원하지만 SageMaker AI용 스트리밍 기능은 지원하지 않습니다. 스트리밍 또는 비스트리밍 기능에 따라 적절한 WebSocket 핸들러(`WebsocketStreamingCallbackHandler` 또는 `WebsocketHandler`)를 호출하여 `post_to_connection` 메서드를 사용하여 응답을 WebSocket 연결로 다시 보냅니다.

`clients/builder` 폴더에는 Builder 패턴을 사용하여 LLM Builder를 빌드하는 데 도움이 되는 클래스가 포함되어 있습니다. 먼저, 구성할 지식 기반, 대화 메모리 및 모델의 유형에 대한 세부 정보를 저장하는 DynamoDB 구성 스토어에서 `use_case_config`를 검색합니다. 또한 모델 파라미터 및 프롬프트와 같은 관련 모델 세부 정보도 포함되어 있습니다. 그런 다음 Builder는 지식 기반을 생성하고, LLM에 대한 대화 컨텍스트를 유지하기 위한 대화 메모리를 생성하고, 스트리밍 및 비스트리밍 사례에 적합한 LangChain 콜백을 설정하고, 제공된 모델 구성을 기반으로 LLM 모델을 생성하는 단계를 수행하는 데 도움이 됩니다. DynamoDB 구성은 배포 대시보드에서 사용 사례를 배포할 때(또는 배포 대시보드 없이 독립 실행형 사용 사례 스택 배포에서 사용자가 제공하는 경우) 사용 사례 생성 시 저장됩니다.

clients/factories 하위 폴더는 LLM 구성을 기반으로 적절한 대화 메모리 및 지식 기반 클래스를 설정하는 데 도움이 됩니다. 이를 통해 구현에서 지원하려는 다른 지식 기반 또는 메모리 유형을 쉽게 확장할 수 있습니다.

shared 하위 폴더에는 빌더가 공장 내에서 인스턴스화하는 지식 기반 및 대화 메모리의 특정 구현이 포함되어 있습니다. 또한 LangChain LLM 모델에서 사용하는 콜백과 함께 RAG 사용 사례에 대한 문서를 검색하기 위해 LangChain 내에서 호출된 Amazon Kendra 및 Amazon Bedrock 지식 기반 리트리버도 포함되어 있습니다.

LangChain 구현은 LangChain Expression Language(LCEL)를 사용하여 대화 체인을 함께 구성합니다. RunnableWithMessageHistory 클래스는 사용자 지정 LCEL 체인과의 대화 기록을 유지 관리하는 데 사용되며, 소스 문서를 반환하고 지식 기반에 전송된 재구분된(또는 모호하지 않은) 질문을 사용하여 LLM으로도 전송할 수 있습니다.

사용자 지정 공급자의 자체 구현을 생성하려면 다음을 수행할 수 있습니다.

a. bedrock_handler.py 파일을 복사하고 사용자 지정 클라이언트(예: custom_handler.py)를 생성하는 사용자 지정 핸들러(예: CustomProviderClient)를 생성합니다(다음 단계에서 지정).

b. 클라이언트 폴더에 bedrock_client.py 복사합니다. 이름을 custom_provider_client.py (또는와 같은 특정 모델 공급자 이름)로 바꿉니다CustomProvider. 가를 상속CustomProviderClient하는 등 클래스의 이름을 적절하게 지정합니다LLMChatClient.

   에서 제공하는 메서드를 사용하거나 자체 구현을 LLMChatClient 작성하여 이를 재정의할 수 있습니다.

   get_model 메서드는 CustomProviderBuilder (다음 단계 참조)를 빌드하고 빌더 단계를 사용하여 채팅 모델을 구성하는 construct_chat_model 메서드를 호출합니다. 이 메서드는 빌더 패턴에서 디렉터 역할을 합니다.

c. 복사clients/builders/bedrock_builder.py하여 로 이름을 custom_provider_builder.py 바꾸고 그 안의 클래스를 LLMBuilder()를 상속CustomProviderBuilder하는 로 이름을 바꿉니다llm_builder.py. LLMBuilder 에서 제공하는 메서드를 사용하거나 자체 구현을 작성하여 이를 재정의할 수 있습니다. 빌더 단계는 , set_model_defaults set_knowledge_base및와 같은 클라이언트의 construct_chat_model 메서드 내에서 순서대로 호출됩니다set_conversation_memory.

`set_llm_model` 메서드는 앞에 호출된 메서드를 사용하여 설정된 모든 값을 사용하여 실제 LLM 모델을 생성합니다. 특히 DynamoDB의 LLM 구성에서 검색`rag_enabled variable`된를 기반으로 RAG(`CustomProviderRetrievalLLM`) 또는 비 RAG(`CustomProviderLLM`) LLM을 생성할 수 있습니다.

이 구성은 `LLMChatClient` 클래스의 `retrieve_use_case_config` 메서드에서 가져옵니다.

d. RAG 또는 비 RAG 사용 사례가 필요한지 여부에 따라 `llm_models` 하위 폴더에서 `CustomProviderLLM` 또는 `CustomProviderRetrievalLLM` 구현을 구현합니다. 이러한 모델을 구현하는 대부분의 기능은 비 RAG `BaseLangChainModel` 및 RAG 사용 사례에 대해 각각 및 `RetrievalLLM` 클래스에서 제공됩니다.

`llm_models/bedrock.py` 파일을 복사하고 필요한 변경을 수행하여 사용자 지정 공급자를 참조하는 LangChain 모델을 호출할 수 있습니다. 예를 들어 Amazon Bedrock은 `ChatBedrock` 클래스를 사용하여 LangChain을 사용하여 채팅 모델을 생성합니다.

생성 메서드는 LangChain LCEL 체인을 사용하여 LLM 응답을 생성합니다.

또한 `get_clean_model_params` 메서드를 사용하여 LangChain 또는 모델 요구 사항에 따라 모델 파라미터를 삭제합니다.

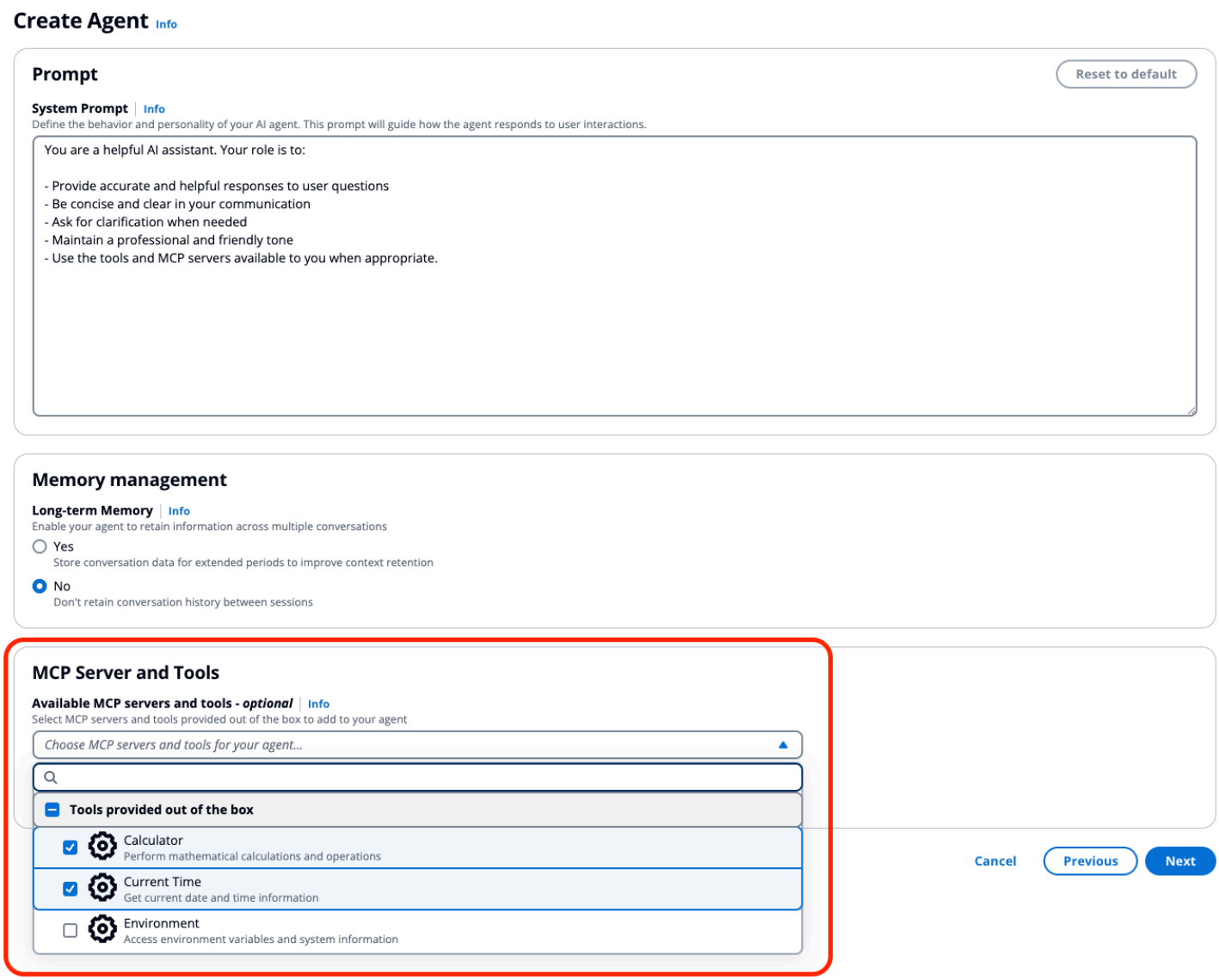
지원되는 Strands 도구 확장

솔루션을 사용하면 MCP 서버, AI 에이전트 및 다중 에이전트 워크플로를 구축하고 배포할 수 있습니다. 에이전트 빌더 경험 내에서 MCP 서버를 연결하여 에이전트에게 추가 기능을 제공할 수 있습니다. MCP 서버 외에도 Strands(솔루션에서 사용하는 기본 프레임워크)에서 제공하는 내장 도구를 활용할 수 있습니다.

기본적으로 솔루션은 다음 Strands 도구로 사전 구성되어 제공됩니다.

- 현재 시간(기본적으로 활성화됨)
- 계산기(기본적으로 활성화됨)
- 환경

기본 제공 Strands 도구를 보여주는 Agent Builder 마법사의 MCP 서버 및 도구 선택



추가 Strands 도구로 에이전트를 확장하려면이 섹션에 설명된 4단계 프로세스를 따르세요.

1단계: Strands 도구 찾기

사용 가능한 Strands 도구를 검색하여 사용하려는 도구를 식별합니다. 각 도구에는 특정 기능 및 구성 요구 사항이 있습니다.

예를 들어 Amazon Bedrock 지식 기반 검색 기능을 추가하려면 검색 도구를 사용합니다.

2단계: SSM 파라미터 업데이트

Agent Builder 배포 UI에서 도구를 사용할 수 있도록 하려면 지원되는 Strands 도구를 정의하는 AWS Systems Manager Parameter Store 파라미터를 업데이트합니다.

1. AWS 계정의 AWS Systems Manager 파라미터 스토어로 이동합니다.
2. 파라미터를 찾습니다. `/gaab/<stack-name>/strands-tools`
3. 다음 JSON 구조를 사용하여 기존 목록의 끝에 도구 구성을 추가합니다.

```
{
  "name": "Bedrock KB Retrieve",
  "description": "Retrieve information from Bedrock Knowledge Base",
  "value": "retrieve",
  "category": "AI",
  "isDefault": false
}
```

필드	설명
이름	Agent Builder UI에 표시된 표시 이름
description	도구 기능에 대한 간략한 설명
USD 상당	Strands 도구 패키지에 정의된 정확한 도구 이름
category	UI에서 도구 그룹화를 위한 조직 범주
isDefault	새 에이전트에 대해 도구를 기본적으로 활성화해야 하는지 여부

3단계: 환경 변수 구성

많은 Strands 도구에는 구성을 위한 환경 변수가 필요합니다. 다음 두 가지 방법으로 이러한 변수를 설정할 수 있습니다.

옵션 1: AgentCore 런타임의 직접 구성

Amazon Bedrock AgentCore 런타임에서 배포된 에이전트를 필요한 환경 변수로 직접 업데이트합니다.

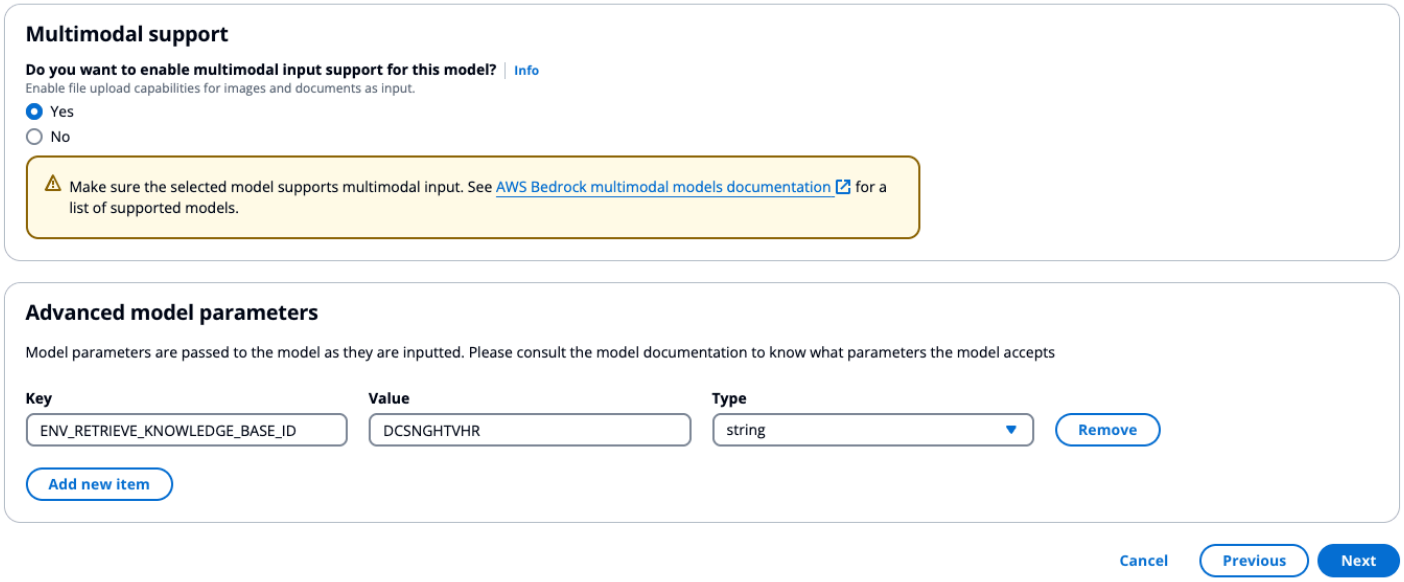
옵션 2: 배포 마법사의 모델 파라미터

모델 파라미터 섹션을 사용하여 Agent Builder 마법사의 모델 선택 단계에서 환경 변수를 추가합니다. 명명 규칙을 따르는 환경 변수ENV_<ALL_CAPS_TOOL_NAME>_<env_variable_name>는 런타임 시 에이전트의 실행 환경에 로 자동으로 로드됩니다<env_variable_name>.

예제:

- ENV_RETRIEVE_KNOWLEDGE_BASE_ID는 KNOWLEDGE_BASE_ID가 됩니다.
- ENV_RETRIEVE_MIN_SCORE는 MIN_SCORE가 됩니다.

ENV_RETRIEVE_KNOWLEDGE_BASE_ID 구성을 보여주는 고급 모델 파라미터 섹션



필요한 환경 변수를 식별하려면 특정 도구의 설명서 또는 소스 코드를 참조하세요. 검색 도구의 경우 소스 코드에서 구성 옵션을 찾을 수 있습니다.

4단계: IAM 권한 추가

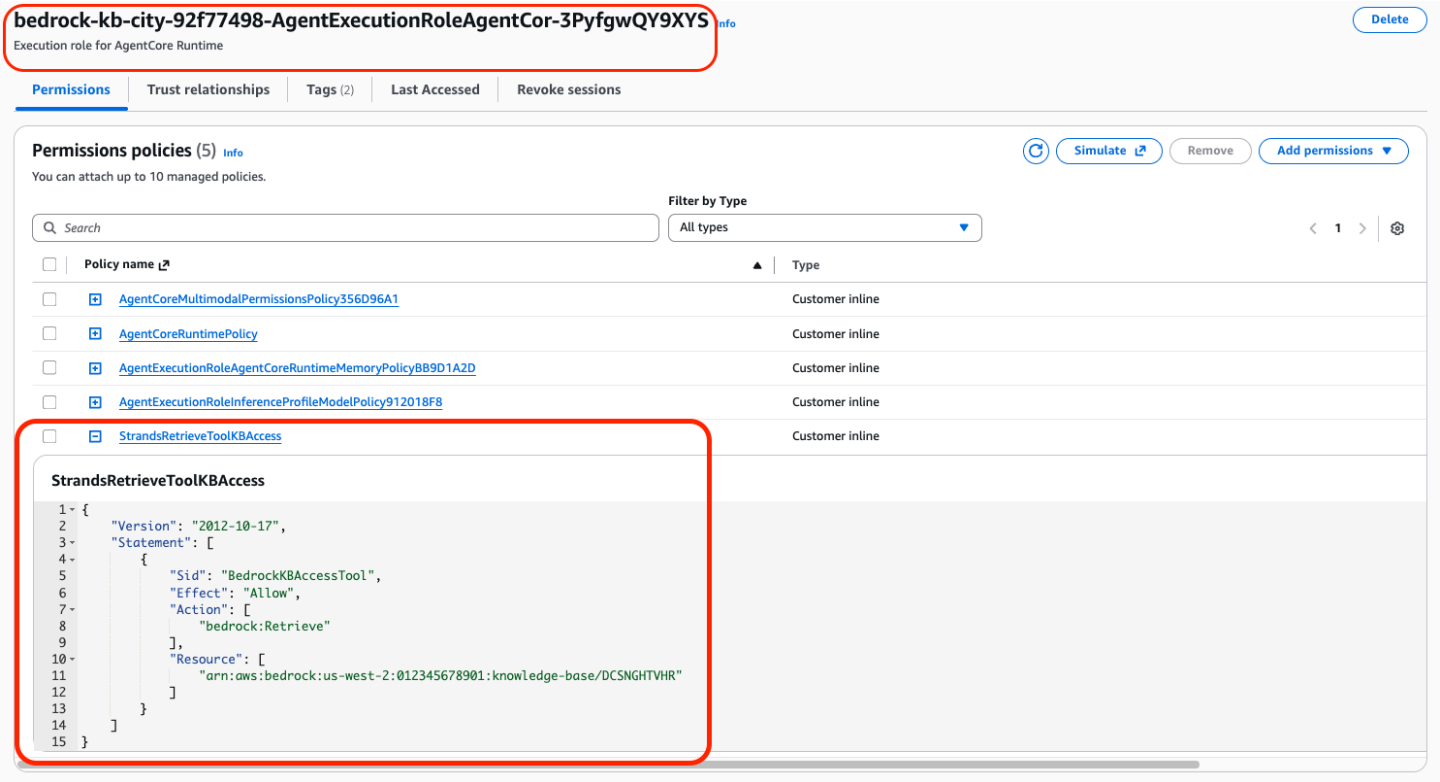
에이전트가 도구를 사용할 수 있도록 AgentCore 런타임 실행 역할에 필요한 IAM 권한을 수동으로 추가합니다.

예를 들어 Amazon Bedrock 지식 기반과 함께 검색 도구를 사용하려면

1. AWS 계정의 IAM 콘솔로 이동합니다.
2. 에이전트의 AgentCore 런타임 실행 역할을 찾습니다.
3. 다음 권한을 추가합니다.

```
{
  "Effect": "Allow",
  "Action": "bedrock:Retrieve",
  "Resource": "arn:aws:bedrock:region:account-id:knowledge-base/knowledge-base-id"
}
```

AgentCore 런타임 실행 역할에 연결된 StrandsRetrieveToolKBAccess 정책을 보여주는 IAM 콘솔

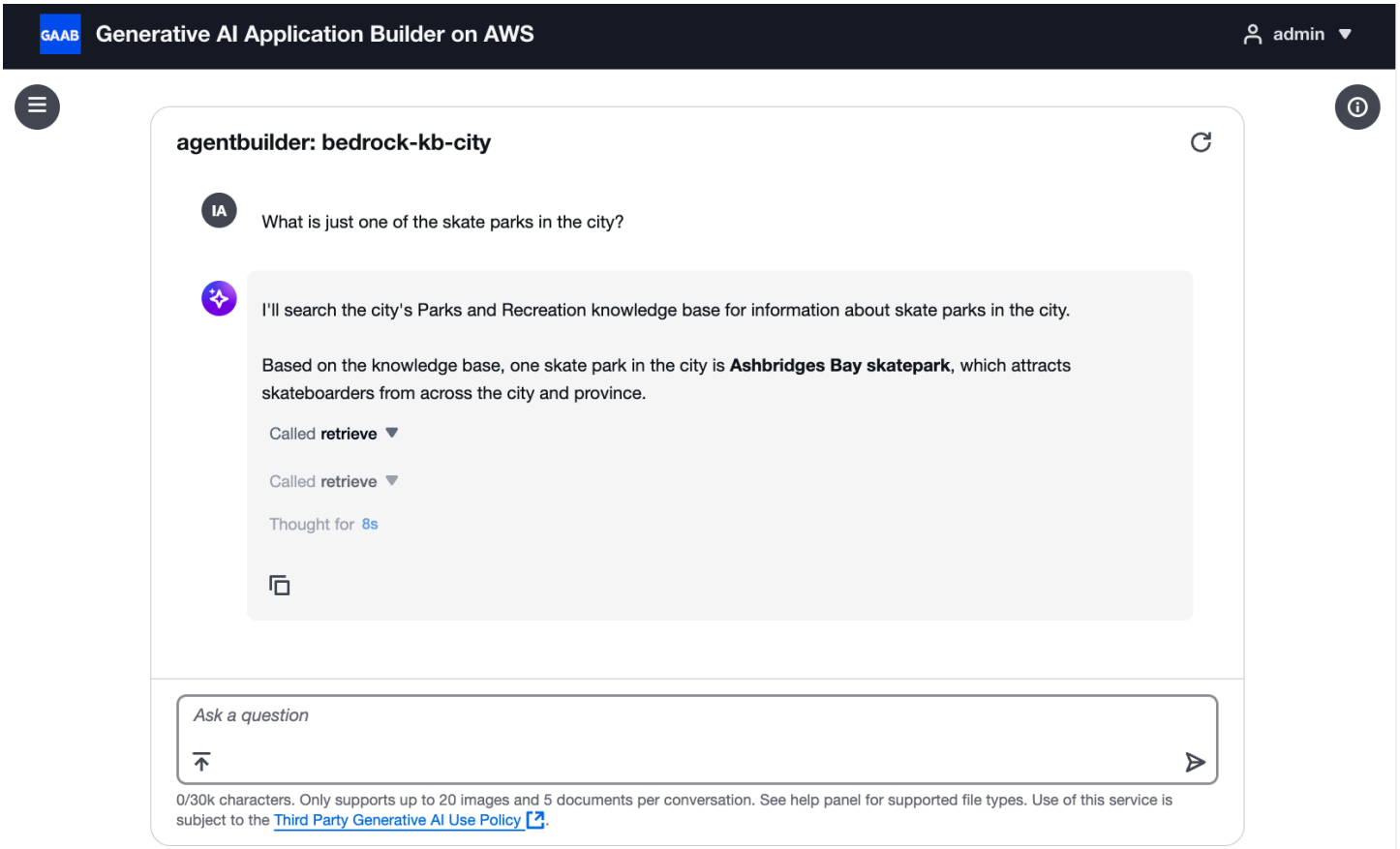


필요한 특정 권한은 도구에 따라 달라집니다. 도구 설명서 및 AWS 서비스 설명서를 참조하여 적절한 IAM 권한을 결정합니다.

5단계: 에이전트 테스트

구성 단계를 완료한 후 에이전트를 테스트하여 도구가 올바르게 작동하는지 확인합니다. 에이전트의 실행 로그 및 응답에 도구 호출이 표시되어야 합니다.

에이전트가 검색 도구를 사용하여 스케이트파크에 대한 질문에 성공적으로 답변했습니다.



> **Note**
>
> 사용 가능한 Strands 도구 및 해당 기능의 전체 목록은 Strands 커뮤니티 도구 설명서를 참조하세요.

지원되는 지식 기반 및 대화 메모리 유형 확장

대화 메모리 또는 지식 기반 구현을 추가하려면 `shared` 폴더에 필요한 구현을 추가한 다음 팩토리와 적절한 열거를 편집하여 이러한 클래스의 인스턴스를 생성합니다.

파라미터 스토어 내에 저장되는 LLM 구성을 제공하면 LLM에 적합한 대화 메모리와 지식 기반이 생성됩니다. 예를 들어이 DynamoDB로 `ConversationMemoryType` 지정되면 `DynamoDBChatMessageHistory` (내에서 사용 가능`shared_components/memory/ddb_enhanced_message_history.py`)의 인스턴스가 생성됩니다. `KnowledgeBaseType`이 Amazon Kendra로 지정되면 `KendraKnowledgeBase` (내에서 사용 가능`shared_components/knowledge/kendra_knowledge_base.py`)의 인스턴스가 생성됩니다.

코드 변경 사항 빌드 및 배포

`npm run build` 명령을 사용하여 프로그램을 빌드합니다. 오류가 해결되면를 실행`cdk synth`하여 템플릿 파일과 모든 Lambda 자산을 생성합니다.

1. 스크립트를 사용하여 생성된 자산을 계정의 `0/stage-assets.sh` 스테이징 버킷으로 수동으로 스테이징할 수 있습니다.
2. 다음 명령을 사용하여 플랫폼을 배포하거나 업데이트합니다.

```
cdk deploy DeploymentPlatformStack --parameters AdminUserEmail='admin-
email@amazon.com'
```

   추가 AWS CloudFormation 파라미터도 AdminUserEmail 파라미터와 함께 제공해야 합니다.

사용자 지정 가이드

Cognito 사용자 풀 관리

배포 대시보드가 배포되면 관리자 사용자와 함께 Amazon Cognito 사용자 풀이 생성되어 애플리케이션에 대한 인증을 제공합니다. 이 사용자 풀은 배포 대시보드와 모든 사용 사례에서 공유됩니다. 대시보드 배포 시 생성된 관리자 사용자에게는 대시보드를 사용하여 배포된 모든 사용 사례에 대한 액세스 권한이 자동으로 부여됩니다. 이 메커니즘은 Amazon Cognito 사용자 풀 그룹을 통해 제공됩니다.

대시보드에서 사용 사례가 배포되면 이메일이 제공되면 특정 사용 사례에 대해 이름이 지정된 사용자 그룹과 함께 공유 사용자 풀에 사용자가 생성됩니다. 그런 다음 새로 생성된 사용자가 그룹에 추가되어 사용자에게 사용 사례에 대한 액세스 권한을 부여합니다.

지정된 사용 사례에 사용자를 추가하려는 경우 Cognito 사용자 풀에서 사용자를 생성하고 사용자가 액세스하도록 하려는 사용 사례(들)에 해당하는 그룹(들)에 추가하면 됩니다. step-by-step 가이드는 AWS Management Console에서 새 사용자 생성을 참조하세요.

마찬가지로 추가 관리자 사용자를 생성하려면 새 사용자를 생성하여 사용자 풀의 관리자 그룹에 추가해야 합니다.

사용자 이름은 제공된 이메일의 일부를 앞에 가져@와서 생성된 사용 사례 UUID(또는 관리자 사용자의 `-admin` 경우)를 추가하여 생성됩니다.

그룹 탭에서 사용 사례의 이름(마법사에 제공됨)과 사용 사례 UUID를 사용하여 각 사용 사례에 대한 관리자 그룹과 그룹이 자동으로 생성되었음을 확인할 수 있습니다.

API 참조

이 섹션에서는 솔루션에 대한 API 참조를 제공합니다.

배포 대시보드

REST API	HTTP 메서드	기능	승인된 호출자
`/deployments`	`GET`	모든 배포를 가져옵니다.	Amazon Cognito 인증 JWT 토큰
`/deployments`	`POST`	새 사용 사례 배포를 생성합니다.	Amazon Cognito 인증 JWT 토큰
`/deployments/{useCaseId}`	`GET`	단일 배포에 대한 배포 세부 정보를 가져옵니다.	Amazon Cognito 인증 JWT 토큰
`/deployments/{useCaseId}`	`PATCH`	지정된 배포를 업데이트합니다.	Amazon Cognito 인증 JWT 토큰
`/deployments/{useCaseId}`	`DELETE`	지정된 배포를 삭제합니다.	Amazon Cognito 인증 JWT 토큰
`/model-info/use-case-types`	`GET`	배포에 사용할 수 있는 사용 사례 유형을 가져옵니다.	Amazon Cognito 인증 JWT 토큰
`/model-info/{useCaseType}/providers`	`GET`	지정된 사용 사례 유형에 사용할 수 있는 모델 공급자를 가져옵니다.	Amazon Cognito 인증 JWT 토큰
`/model-info/{useCaseType}/{providerName}`	`GET`	지정된 공급자 및 사용 사례 유형에 사용할 수 있는 모델의 IDs를 가져옵니다.	Amazon Cognito 인증 JWT 토큰

REST API	HTTP 메서드	기능	승인된 호출자
`/model-info/{useCaseType}/{providerName}/{modelId}`	`GET`	기본 파라미터를 포함하여 지정된 모델에 대한 정보를 가져옵니다.	Amazon Cognito 인증 JWT 토큰

> **Note**
>
> API와 더 쉽게 통합하기 위해 API Gateway에서 OpenAPI 및 Swagger 파일을 내보낼 수도 있습니다. API Gateway에서 REST API 내보내기를 참조하세요.

POST 및 PATCH 페이로드

새 사용 사례를 생성하는 `/deployments` 엔드포인트에 대한 POST 페이로드의 예는 아래를 참조하세요.

```
{
 "UseCaseName": "usecase1",
 "UseCaseDescription": "Description of the use case to be deployed. For display
 purposes", // optional
 "DefaultUserEmail": "placeholder@example.com", // optional, if not provided, the
 Cognito Group and User will not be created
 "DeployUI": true, // optional
 "VpcParams": {
 "VpcEnabled": true,
 "CreateNewVpc": false,
 // provide these if not creating new vpc
 "ExistingVpcId": "vpc-id",
 "ExistingPrivateSubnetIds": ["subnet-1", "subnet-2"],
 "ExistingSecurityGroupIds": ["sg-1", "sg-2"]
 },
 "ConversationMemoryParams": {
 "ConversationMemoryType": "DynamoDB",
 "HumanPrefix": "user", // optional
 "AiPrefix": "ai", // optional
 "ChatHistoryLength": 10 // optional
 },
 "KnowledgeBaseParams": {
```

```
 "KnowledgeBaseType": "Bedrock",
 // one of the following based on selected provider
 "BedrockKnowledgeBaseParams": {
 "BedrockKnowledgeBaseId": "my-bedrock-kb",
 "RetrievalFilter": {}, // optional
 "OverrideSearchType": "HYBRID" // optional
 },
 "KendraKnowledgeBaseParams": {
 "AttributeFilter": {}, // optional
 "RoleBasedAccessControlEnabled": true, // optional
 "ExistingKendraIndexId": "12345678-abcd-1234-abcd-1234567890ab",
 // provide the following in place of ExistingKendraIndexId if you want the solution to
 deploy an index for you
 "KendraIndexName": "index",
 "QueryCapacityUnits": 1, // optional
 "StorageCapacityUnits": 1, // optional
 "KendraIndexEdition": "DEVELOPER" // optional
 },
 "NoDocsFoundResponse": "Sorry, I couldn't find any relevant information for your
 query.", // optional
 "NumberOfDocs": 3, // optional
 "ScoreThreshold": 0.7, // optional
 "ReturnSourceDocs": true // optional
 },
 "LlmParams": {
 "ModelProvider": "Bedrock | SAGEMAKER",
 // one of the following based on selected provider
 "BedrockLlmParams": {
 "ModelId": "model-id", // use this for on demand models. Can't use with ModelArn
 "ModelArn": "model-arn", // use this for provisioned/custom models. Can't use with
 ModelId,
 "InferenceProfileId": "profile-id"
 "GuardrailIdentifier": "arn:aws:bedrock:us-east-1:123456789012:guardrail/my-
guardrail", // optional
 "GuardrailVersion": "1" // optional. Required if GuardrailIdentifier provided.
 },
 "SageMakerLlmParams": {
 "EndpointName": "some-endpoint",
 "ModelInputPayloadSchema": {},
 "ModelOutputJSONPath": "$."
 },
 // optional. Passes on arbitrary params to the underlying LLM.
 "ModelParams": {
 "param1": {
```

```
 "Value": "value1",
 "Type": "string"
 },
 "param2": {
 "Value": 1,
 "Type": "integer"
 }
 },
 // optional
 "PromptParams": {
 "PromptTemplate": "some template",
 "UserPromptEditingEnabled": true,
 "MaxPromptTemplateLength": 1000,
 "MaxInputTextLength": 1000,
 "DisambiguationPromptTemplate": "some disambiguation template",
 "DisambiguationEnabled": true
 },
 "Temperature": 1.0, // optional
 "Streaming": true, // optional
 "RAGEnabled": true, // optional. Must be true if providing KnowledgeBaseParams above.
 "Verbose": false // optional
 },
 "AgentParams": {
 "AgentType": "Bedrock",
 "BedrockAgentParams": {
 "AgentId": "agent-id",
 "AgentAliasId": "alias-id",
 "EnableTrace": true
 }
 },
 // optional
 "AuthenticationParams": {
 "AuthenticationProvider": "Cognito",
 "CognitoParams": {
 "ExistingUserPoolId": "user-pool-id",
 "ExistingUserPoolClientId": "client-id" // optional. If not provided, the solution
 will create a client for you in the provided pool
 }
 }

}
```

업데이트의 경우 구조는 위와 동일하며 몇 가지 주의 사항이 있습니다.

- 사용 사례 이름은 변경할 수 없습니다.
- 사용 사례는 VPC에 배포된 후에만 보안 그룹 및 서브넷을 변경할 수 있습니다. VPC 자체는 변경할 수 없습니다.
- Kendra 인덱스가 지식 기반으로 생성된 경우 해당 인덱스의 구성을 변경할 수 없습니다(예: KendraIndexName, QueryCapacityUnits).

공유 사용 사례 APIs

텍스트 및 Bedrock Agent 사용 사례 모두에 사용할 수 있는 REST API 엔드포인트는 다음과 같습니다.

REST API	HTTP 메서드	기능	승인된 호출자
`/details/{useCaseConfigKey}`	`GET`	특정 사용 사례에 대한 구성 세부 정보를 가져옵니다.	Amazon Cognito 인증 JWT 토큰

WebSocket API	기능	승인된 호출자
`/$connect`	WebSocket 연결을 시작하고 사용자를 인증합니다.	Amazon Cognito 인증 JWT 토큰
`/$disconnect`	WebSocket 연결이 끊어졌을 때 호출되는 엔드포인트입니다.	Amazon Cognito 인증 JWT 토큰

사용 사례 세부 정보 API

세부 정보 API 엔드포인트는 특정 사용 사례에 대한 정보를 검색합니다.

```
GET /details/{useCaseConfigKey}
```

이 엔드포인트는 모델 파라미터, 지식 기반 설정 및 기타 배포 정보를 포함하여 특정 사용 사례에 대한 구성 세부 정보를 반환합니다. 권한 부여를 위해 Amazon Cognito 인증 JWT 토큰이 필요합니다.

텍스트 사용 사례

WebSocket API	기능	승인된 호출자
`/sendMessage`	구성된 LLM 환경으로 처리하기 위해 사용자의 채팅 메시지를 WebSocket으로 전송합니다.	Amazon Cognito 인증 JWT 토큰

REST API	HTTP 메서드	기능	승인된 호출자
`/feedback/{useCaseId}`	`POST`	특정 사용 사례에 대한 사용자 피드백을 제출합니다.	Amazon Cognito 인증 JWT 토큰

sendMessage 페이로드

`/sendMessage` API와 직접 통합하는 경우 다음 요청 및 응답 페이로드 형식을 준수해야 합니다.

페이로드 요청

```
{
 "action": "sendMessage",
 "question": "the message to send to the api",
 "conversationId": "", // If not provided, a new conversation will be created, with the
 conversationId returned in the response. All subsequent messages in that conversation
 (where history is retained), should provide the conversationId there.
 "promptTemplate": "", // Optional. Overrides the configured prompt
 "authToken": "XXXX" // Optional. accessToken from cognito flow. Required for RAG with
 RBAC
}
```

파라미터 이름	Type	설명
action	`String`	현재는 WebSocket에 대한 "sendMessage" 작업만 지원합니다.

파라미터 이름	Type	설명
질문	`String`	LLM으로 전송할 사용자 입력
conversationId	`String`	대화를 식별하는 UUID입니다. 제공되지 않으면 새 대화가 생성되고 응답에 conversationId가 반환됩니다. 해당 대화의 모든 후속 메시지(기록/컨텍스트를 보존하려는 경우)는 여기에 conversationId를 제공해야 합니다.
promptTemplate	`String` [선택 사항]	이 메시지에 대한 프롬프트 템플릿을 재정의합니다. 비어 있거나 제공되지 않은 경우는 배포 시 기본적으로 프롬프트 세트로 설정됩니다. 모든 배포에 RAG를 사용하는 경우 {context}를 추가하여 지정된 구성(예: 비 RAG Sagemaker AI 배포의 경우 {history} 및 {input})에 대해 적절한 자리 표시자를 지정해야 합니다.

파라미터 이름	Type	설명
authToken	`String` [선택 사항]	cognito 인증 흐름에서 얻은 accessToken입니다. 이는 역할 기반 액세스 제어(RBAC)를 사용하여 RAG에 대해 구성된 채팅 웹 소켓 엔드포인트를 호출할 때 필요합니다. 이 JWT 토큰의 cognito:groups 클레임 목록은 Kendra 인덱스의 문서에 대한 액세스를 제어하는 데 사용됩니다. 이 파라미터는 비 RAG 사용 사례에 필요하지 않습니다. RBAC가 비활성화된 RAG 사용 사례에는 필요하지 않습니다.

응답 페이로드

질문 응답

WebSocket API는 각 쿼리에 대해 다음과 같이 구조화된 1개(스트리밍이 비활성화된 경우) 또는 여러 개(스트리밍이 활성화된 경우)의 JSON 객체로 응답합니다.

```
{
 "data": "some data",
 "conversationId": "id",
}
```

파라미터 이름	Type	설명
data	`String`	스트리밍이 활성화된 경우 LLM의 응답 청크 또는 전체 응답입니다. 스트리밍을 사용하는 경우 데이터 콘텐츠가 END_CONVERSATION인이 형식의 응답이 전송되어 단일

파라미터 이름	Type	설명
		질문에 대한 응답의 끝을 나타냅니다.
conversationId	String	이 sourceDocument 응답이 속한 대화의 ID입니다.

소스 문서 응답

소스 문서를 반환하도록 RAG 사용 사례를 구성한 경우 응답을 생성하는 데 사용되는 각 소스 문서에 대한 모든 응답이 끝날 때 다음 페이로드도 받게 됩니다.

```
{
 "sourceDocument": {
 "excerpt": "some excerpt from the",
 "location": "s3://fake-bucket/test.txt",
 "score": 0.500,
 "document_title": null,
 "document_id": null,
 "additional_attributes": null
 },
 "conversationId": "some-id"
}
```

파라미터 이름	Type	설명
발췌문	String	소스 문서에서 발췌한 입니다.
location	String	소스 문서의 위치입니다. 이는 사용된 데이터 소스 및 지식 기반 유형에 따라 다르지만 s3 URIs 또는 웹 사이트와 같을 수 있습니다.
점수	Number | String	문서가 질문한 질문에 해당한다는 신뢰도입니다. Bedrock의 경우 0~1의 부동 소수점이

파라미터 이름	Type	설명
		고 Kendra의 경우 문자열(예: HIGH, LOW 등)입니다.
document_title	`String`	반환된 소스 문서의 제목입니다. Kendra를 사용할 때만 사용할 수 있습니다.
document_id	`String`	반환된 소스 문서의 ID입니다. Kendra를 사용할 때만 사용할 수 있습니다.
additional_attributes	`String`	이 필드에는 수집 시 지식 기반에 사용자 지정된 대로 문서의 모든 추가 속성이 포함됩니다.
conversationId	`String`	이 sourceDocument 응답이 속한 대화의 ID입니다.

피드백 API 페이로드

다음은 특정 사용 사례에 대한 사용자 피드백을 제출하는 `/feedback/{useCaseId}` 엔드포인트에 대한 POST 페이로드의 예입니다.

```
{
  "useCaseRecordKey": "12345678-12345678",
  "conversationId": "12345678-1234-1234-1234-123456789012",
  "messageId": "12345678-1234-1234-1234-123456789012",
  "feedback": "positive",
  "feedbackReason": ["accurate", "helpful"],
  "comment": "This response was very helpful.",
  "rephrasedQuery": "What are the key features of Amazon Bedrock?",
  "sourceDocuments": [
    "s3://bucket-name/document1.pdf",
    "s3://bucket-name/document2.pdf"
  ]
}
```

Bedrock Agent 사용 사례

WebSocket API	기능	승인된 호출자
`/invokeAgent`	구성된 에이전트로 처리할 수 있도록 사용자의 메시지를 WebSocket으로 전송합니다.	Amazon Cognito 인증 JWT 토큰

invokeAgent 페이로드

와 직접 통합하는 경우 다음 요청 및 응답 페이로드 형식을 준수해야 `/invokeAgent` API합니다.

요청 페이로드

```
{
"action": "invokeAgent",
"inputText": "User query to the agent",
"conversationId": "", // Optional. Empty conversationId implies a new conversation.
 When not provided, a new conversationId will be created and returned with the
 response. All subsequent messages in the same conversation should provide the same
 conversationId (i.e. chat memory/history is maintained).
"authToken": "XXXX" // Optional. accessToken from cognito flow. If provided, it needs
 to be a valid JWT token associated with the user
}
```

파라미터 이름	Type	설명
action	`String`	WebSocket에 대한 `invokeAgent` 작업만 지원합니다.
inputText	`String`	LLM으로 전송할 사용자 입력입니다.
conversationId	`String[Optional]`	대화를 고유하게 식별하는 UUID입니다. 이 값을 제공하지 않으면 솔루션은 새 대화를 생성하고 응답에 conversat

파라미터 이름	Type	설명
		ionId를 반환합니다. 해당 대화의 모든 후속 메시지(기록 및 컨텍스트를 유지하려는 경우)는 여기에 conversationId를 제공합니다.
authToken	`String[Optional]`	accessToken은 Amazon Cognito 인증 흐름에서 가져옵니다. 이 파라미터는 필요하지 않습니다. 제공하면 JWT 토큰이 검증됩니다. 이렇게 하면이 솔루션을 더 쉽게 확장할 수 있습니다.

응답 페이로드

질문 응답

WebSocket API는 각 쿼리에 대해 다음과 같이 구조화된 JSON 객체 하나(스트리밍이 비활성화된 경우) 또는 여러 개(스트리밍이 활성화된 경우)로 응답합니다.

```
{
 "data" "some data",
 "conversationId": "id",
 }
```

파라미터 이름	Type	설명
data	`String`	에이전트 호출의 응답입니다.
conversationId	`String`	대화의 ID입니다.

레퍼런스

이 섹션에는이 솔루션의 데이터 수집, 관련 리소스에 대한 포인터,이 솔루션에 기여한 빌더 목록에 대한 정보가 포함되어 있습니다.

지원되는 LLM 공급자

솔루션은 다음 LLM 공급자와 통합할 수 있습니다.

1. Amazon Bedrock
   - 설명서: https://aws.amazon.com/bedrock/
   - 지원되는 모델:
     - Amazon
       - Nova Lite
       - Nova Micro
       - Nova Pro
     - AI21 Labs
       - Jamba 1.5 Mini
       - Jamba 1.5 Large
     - Anthropic
       - Claude v3 하이쿠
       - Claude v3.5 Sonnet
       - Claude v3.7 Sonnet(추론 프로파일 사용)
     - Cohere
       - Command R
       - Command R+
     - Deepseek
       - Deepseek-R1(추론 프로파일 사용)
     - Meta
       - Llama 3
       - Llama 3.2(추론 프로파일 사용)
     - Mistral AI

        - Mistral 7B 지침
        - Mistral 8x7B 지침
    - 교차 리전 추론
        - 배포 대시보드와 동일한 리전에 정의된 추론 프로파일을 사용할 수 있는 기능

2. Amazon SageMaker AI
    - 설명서: https://aws.amazon.com/sagemaker/
    - 지원되는 모델: 텍스트 투 텍스트 모델

최신 모델 파라미터, 모범 사례 및 권장 용도는 모델 공급자의 설명서를 참조하세요.

데이터 수집

이 솔루션은이 솔루션 사용에 대한 운영 지표("데이터")를 AWS로 전송합니다. 당사는 고객이이 솔루션과 관련 서비스 및 제품을 사용하는 방법을 더 잘 이해하기 위해이 데이터를 사용합니다. AWS의이 데이터 수집에는 AWS 개인 정보 보호 고지가 적용됩니다.

기여자

- Tarek Abdunabi
- Majd Arbash
- 조지 비어덴
- Mukit Bin Momin
- Michael Connor
- Johny Duval
- Nihit Kasabwala
- Ahern Knox
- 사이몬 크롤
- 마이클 린
- 팀 메카리
- 이브라힘 모하메드
- Omar Radwan Mohsen
- 제임스 닉슨

- Dekshitha Ravikumar
- 심재
- Ajay Swamy
- 모하메드 타하
- Reet Takkar
- Dimitri Tchikatilov
- Jason Wreath
- 카미아 자바리

개정

게시일: 2023년 10월(마지막 업데이트: 2025년 1월)

GitHub 리포지토리의 CHANGELOG.md 파일을 보고 소프트웨어에 대한 모든 주요 변경 사항 및 업데이트를 확인하세요. 변경 로그는 각 버전의 개선 및 수정 사항에 대한 명확한 기록을 제공합니다.

Notices

고객은 본 문서의 정보를 독립적으로 평가할 책임이 있습니다. 이 문서는 (a) 정보 제공만을 목적으로 하고, (b) AWS의 현행 제품 제공 및 관행을 나타내며, (c) AWS와 그 계열사, 공급업체 또는 라이선스 제공자로부터 어떠한 약정이나 보증도 하지 않습니다. AWS 제품 또는 서비스는 명시적이든 묵시적이든 어떠한 종류의 보증, 진술 또는 조건 없이 “있는 그대로” 제공됩니다. 고객에 대한 AWS의 책임 및 채무는 AWS 계약에 준거합니다. 본 문서는 AWS와 고객 간의 어떠한 계약도 구성하지 않으며 이를 변경하지도 않습니다.

AWS의 생성형 AI Application Builder는 Apache 라이선스 버전 2.0의 약관에 따라 라이선스가 부여됩니다.

Important

AWS의 생성형 AI Application Builder를 사용하면 AWS가 소유하지 않거나 다른 방식으로 제어할 수 있는 타사 생성형 AI 모델(“타사 생성형 AI 모델”)을 포함하여 원하는 생성형 AI 모델을 참여시켜 AWS에서 생성형 인공 지능 애플리케이션을 구축하고 배포할 수 있습니다.
타사 생성형 AI 모델 사용에는 타사 생성형 AI 모델 공급자가 사용 라이선스를 획득할 때 제공한 약관(예: 서비스 약관, 라이선스 계약, 허용 가능한 사용 정책 및 개인 정보 보호 정책)이 적용됩니다.
사용자는 타사 생성형 AI 모델 사용이 타사 생성형 AI 모델에 적용되는 약관과 사용자에게 적용되는 모든 법률, 규칙, 규정, 정책 또는 표준을 준수하도록 할 책임이 있습니다.
또한 출력 및 타사 생성형 AI 모델 공급자가 배포에 따라 전송할 수 있는 데이터를 사용하는 방법을 포함하여 사용하는 타사 생성형 AI 모델을 독립적으로 평가할 책임이 있습니다. AWS는 AWS와의 계약에 따라 '타사 콘텐츠'인 타사 생성형 AI 모델과 관련하여 어떠한 표현, 보증 또는 보장도 하지 않습니다. AWS의 생성형 AI Application Builder는 AWS와의 계약에 따라 "AWS 콘텐츠"로 제공됩니다.

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.