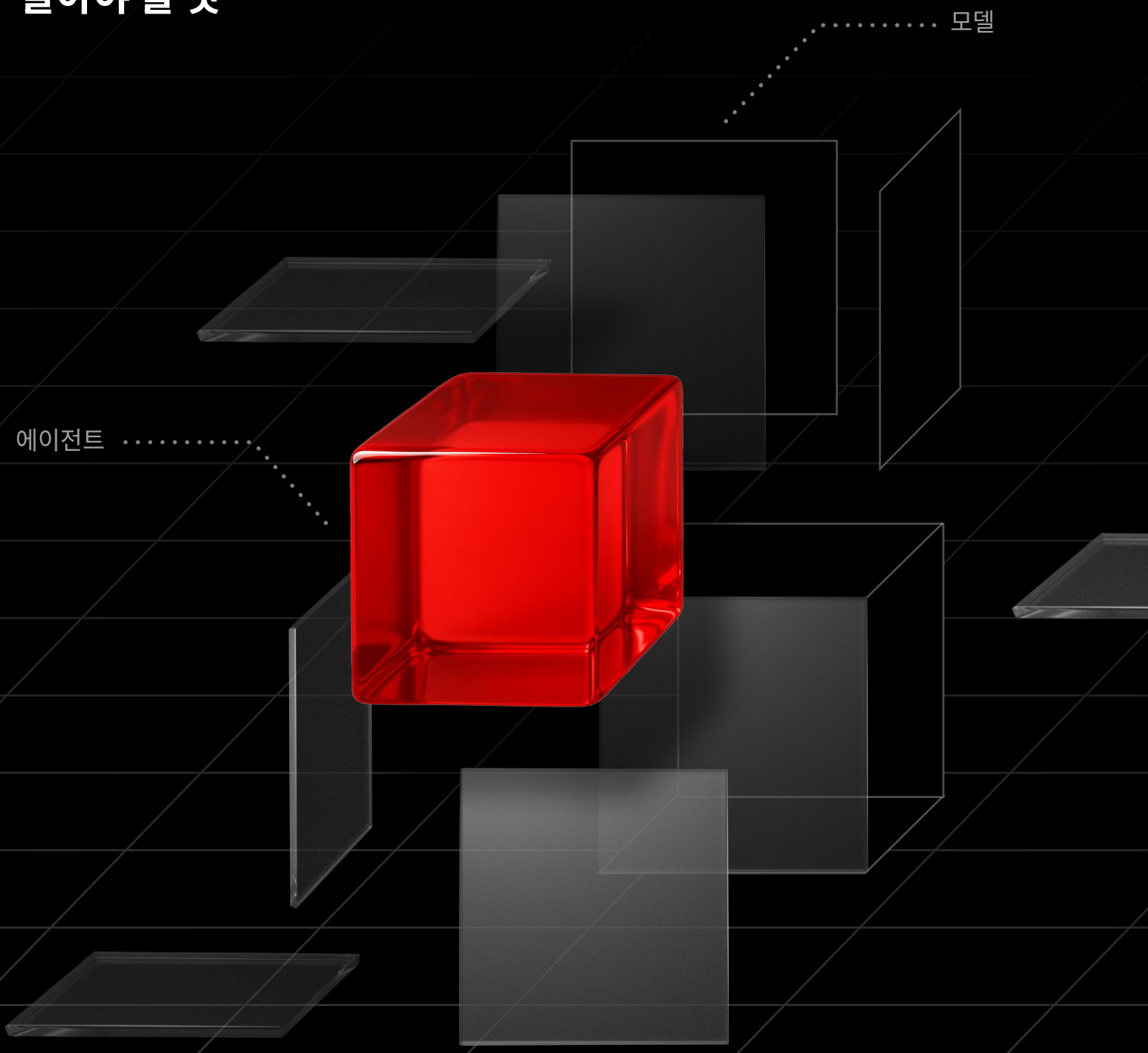
구축 전
알아야 할 것



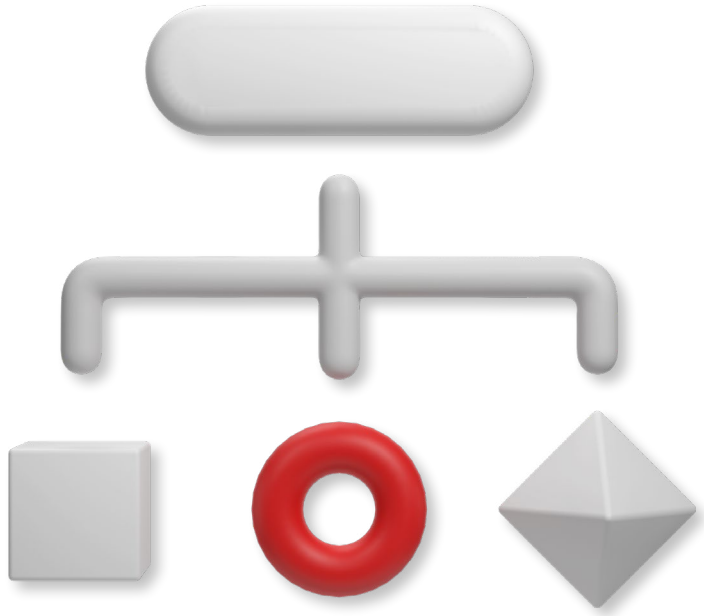
Red Hat AI를 통해 AI 에이전틱 시스템 살펴보기

Red Hat

목차

개요

에이전틱 AI로의 기술 진화



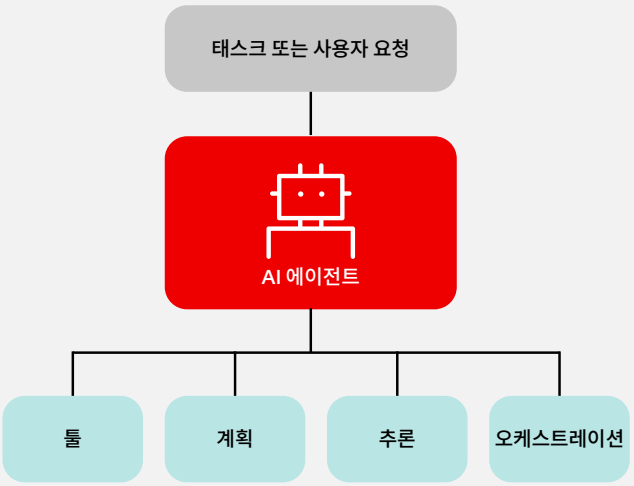
에이전틱(Agentic) AI가 자동화, 의사 결정, 확장성에 대한 기업의 생각을 바꾸고 있습니다. 그러나 실제 환경에서 작동하는 에이전트를 구축하는 것은 챗봇에 프롬프트를 입력하는 것만큼 간단하지 않습니다. 프로덕션 레디 에이전틱 시스템에는 언어 모델 이상의 것이 필요합니다. 추론을 조율하고 툴을 오케스트레이션하며 메모리를 유지하고 데이터를 보호하며 동작을 제어하는 통합 아키텍처가 필요합니다.

이 e-Book에서는 Red Hat의 오픈소스 접근 방식에 기초한 이러한 아키텍처를 기술적 관점에서 살펴봅니다. 추론과 툴 오케스트레이션에서 안전과 관측성에 이르기까지 각 구성 요소는 모듈식이어야 하고 보안에 중점을 둬야 하며 프로덕션에서 바로 사용 가능해야 합니다. Red Hat® AI는 바로 이런 솔루션입니다. Red Hat OpenShift® AI를 통해 구축되고 Llama Stack으로 구동되는 Red Hat AI는 지능형 에이전트를 규모에 맞게 구축, 배포, 관리하기 위한 토대를 제공하는 플랫폼입니다.

조직은 분산된 프레임워크와 비일관적인 툴체인을 다룰 필요 없이 Red Hat AI를 사용하여 에이전틱 워크플로우를 구축하고 확장하는 방법을 표준화할 수 있습니다. 모델 컨텍스트 프로토콜(MCP)과 같은 오픈 표준을 통해 Red Hat은 조직이 에이전트의 툴 검색 및 사용 방법을 통합하도록 돕습니다. Red Hat은 안전, 평가 및 인프라 자동화를 통합하여 팀이 향상된 반복성과 신뢰를 바탕으로 기술 검증(POC)에서 프로덕션까지 이동할 수 있도록 지원합니다.

이것은 엔터프라이즈 AI의 다음 단계이며, 표준화는 이를 실현하는 기반입니다.

에이전틱 AI 시스템의 구성 요소



- **툴 사용:** 외부 툴을 사용하여 데이터를 수집하고 태스크를 수행합니다.
- **계획 및 실행:** 목표를 자율적으로 달성하기 위한 멀티 스텝 계획을 수립하고 실행합니다.
- **추론:** 논리와 맥락적 이해를 적용해 정보에 입각한 의사 결정을 내립니다.
- **오케스트레이션:** 작업, 툴, 에이전트를 조율하여 태스크를 동적으로 조정하고 완료합니다.
- **통신 프로토콜:** 구성 요소 간 연결을 허용합니다.

그림 1. 표준 LLM보다 기능이 많은 에이전틱 시스템

알아야 할 용어

에이전틱 AI 시스템 및 워크플로우

에이전틱 AI 시스템은 단순한 대규모 언어 모델(LLM) 그 이상의 것입니다. 추론, 메모리, 계획, 외부 툴 등을 사용하여 시간이 지남에 따라 복잡한 태스크를 수행하기 위해 연동하는 여러 AI 시스템들의 집합입니다. 이러한 시스템들은 구조화된 워크플로우를 따르므로 AI가 실제 상황에 대응하여 자율적 또는 반자율적으로 작동할 수 있습니다.

모델 컨텍스트 프로토콜(MCP)

MCP는 AI 에이전트가 일관되고 해석 가능한 방식으로 툴, 데이터 및 메모리와 상호작용하는 방법을 정의하는 오픈 표준입니다. MCP는 개발자가 재사용 가능하고 디버그하거나 확장하기가 더 간편한 모듈식 AI 시스템을 설계하는 데 도움이 됩니다.

Llama Stack

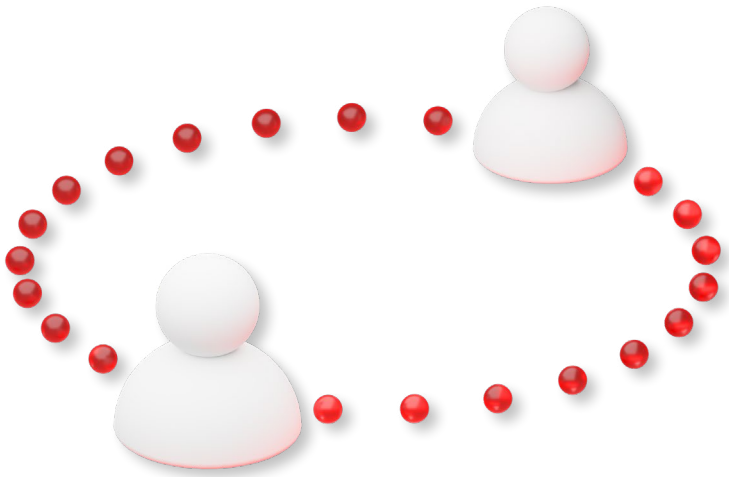
Llama Stack은 애플리케이션 프로그래밍 인터페이스(API), 오케스트레이션, 로깅, 툴 통합 등을 비롯한 프로덕션 레디 툴에 Llama 모델을 감싸는 통합 소프트웨어 계층입니다. Llama Stack은 엔터프라이즈 환경에서 Llama 기반 에이전틱 AI 시스템의 배포와 운영을 간소화합니다.

LangChain

LangChain은 개발자가 언어 모델을 사용하여 복잡한 애플리케이션을 빌드하도록 지원하는 오픈소스 프레임워크입니다. MCP, Llama Stack과 매우 유사하게 LangChain은 모델을 외부 데이터, 메모리 및 툴과 연결하는 툴을 제공합니다. LangChain은 Red Hat AI와 함께 사용되는 기본 프레임워크가 아니지만 Red Hat AI는 프로젝트에 적합한 다른 접근 방식을 지원할 수 있는 유연한 개방형 플랫폼입니다.

1장

Red Hat AI를 통한 AI 에이전트 생성 및 배포



에이전틱 AI 시스템을 구축하려면 대규모 언어 모델(LLM)을 애플리케이션에 임베드하는 것보다 훨씬 더 많은 작업이 필요합니다. 모든 효과적인 에이전트의 중심에는 추론, 계획, 동작, 학습이 가능한 시스템이 존재하며, 이를 위해 다양한 기술 구성 요소가 필요합니다. 여기에는 문제를 하위 태스크들로 나누는 추론 사슬, 에이전트 동작을 정의하는 프롬프트, 컨텍스트 유지를 위한 메모리, 그리고 LLM의 사전 학습된 가중치 이상으로 조치를 취할 수 있는 역량을 제공하는 외부 툴 등이 포함됩니다.

Red Hat AI는 이러한 복잡한 아키텍처를 간소화하는 데 도움이 됩니다. OpenShift AI를 핵심 구성 요소로 하는 이 플랫폼은 추론, 오케스트레이션, 보안, 관측성, 컴플라이언스와 같은 필수 기능을 통합하고 이를 AI 에이전트가 효과적으로 기능하는 데 필요한 툴에 연결합니다.

Red Hat AI를 사용하는 팀은 관리 가능한 활용 사례부터 시작해 시간이 지남에 따라 확장할 수 있습니다. 많은 조직이 처음에는 내부 검색 에이전트부터 구축하며, 회사 내부 데이터를 기반으로 질문에 답하는 LLM 기반 지식 봇을 배포합니다. 어떤 조직들은 로그 문제 해결과 IT 인시던트 해결을 위해 에이전트를 구축하고, Red Hat OpenShift 관측성을 Red Hat Ansible® Automation Platform 및 외부 애플리케이션 프로그래밍 인터페이스(API)와 통합합니다. 또 다른 일반적 시나리오는 AI 지원 코드 마이그레이션으로, 에이전트가 리포지토리를 분석하고 업그레이드 경로를 제안함으로써 기존 방식에서 현대적인 방식으로의 전환을 지원합니다.

무엇보다 중요한 것은 이러한 에이전트들은 단순히 일회성 어시스턴트가 아니라 추론과 메모리를 장착한 멀티 스텝 워크플로우의 일부라는 것입니다. 복잡성이 증가함에 따라 Red Hat은 명확한 위임 및 결정 체크포인트를 바탕으로 멀티 에이전트 애플리케이션을 위한 통합 라이프사이클 관리 및 오케스트레이션 기능을 제공합니다.

엔터프라이즈급 에이전틱 AI를 구축하는 경로는 구성 가능한 아키텍처, 반복 가능한 툴링, 실험을 운영으로 전환하는 통합 플랫폼을 기반으로 시작됩니다. Red Hat AI는 팀이 자신감 있게 구축하고 의도에 맞게 배포할 수 있도록 돕습니다.

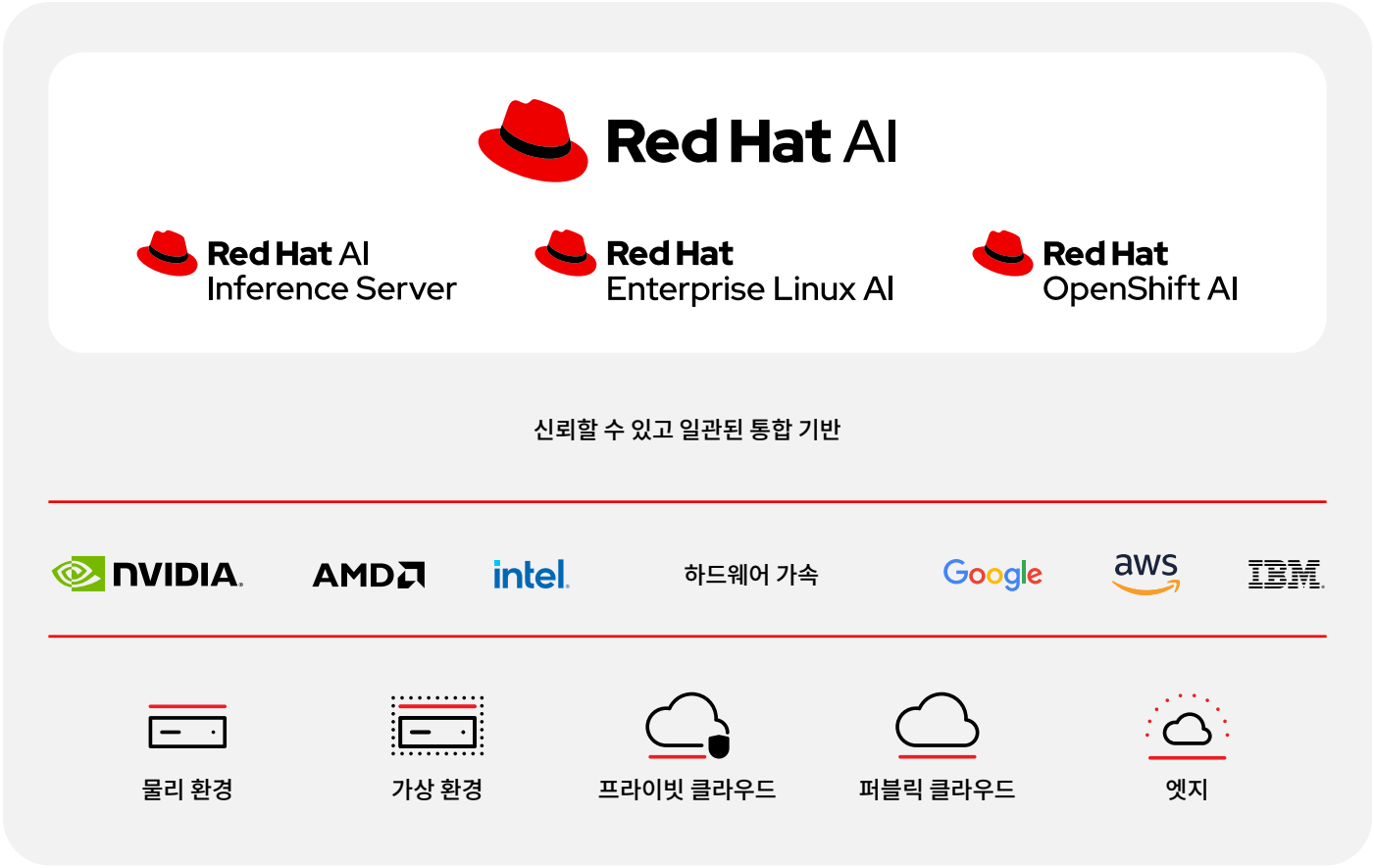


그림 2. Red Hat AI의 일부인 OpenShift AI

활용 사례

오늘날 조직이 에이전트를 구축하는 방법

대규모 고객 지원

한 사이버 보안 회사는 대기 시간을 줄이고 인간 에이전트의 부담을 덜기 위해 AI 기반 지원 시스템을 구현하여 라이브 채팅과 티켓 해결을 자동화했습니다. 접수 에이전트가 문의를 처리하면 문의가 분류 에이전트로 전달됩니다. 그러면 분류 에이전트가 키워드 목록과 생성형 AI 감성 분석을 사용하여 감성 태그 지정, 긴급성 분석, 욕설 및 비속어 검사를 수행합니다. 문제 언어가 발견될 경우 수동 검토가 트리거됩니다.

그다음, 문제 해결 에이전트가 검색 증강 생성(RAG)을 사용하여 내부 문서와 과거 티켓을 가져와 응답을 생성합니다. 필요한 경우 라우팅 에이전트가 미해결된 티켓 또는 확신이 낮은 티켓을 인간 지원 담당자에게 에스컬레이션합니다. 인간 에이전트는 AI가 제안한 응답을 검증하며 특히 컴플라이언스에 민감한 콘텐츠에 더욱 주의합니다. Airtable 로그는 인간 에이전트가 재작성한 답변과 오류를 트래킹하여 지속적인 모델 개선을 지원합니다. 작업 종료 후에는 고객 만족도 확인을 위해 감성을 재평가합니다.

2장

Red Hat OpenShift AI: AI 라이프사이클을 위한 엔터프라이즈급 플랫폼

프로토타입에서 프로덕션으로의 전환은 엔터프라이즈 AI, 특히 에이전틱 애플리케이션과 관련하여 가장 오래된 과제 중 하나입니다. OpenShift AI는 조직이 에이전틱 애플리케이션을 빌드, 실행, 관리하는 데 도움이 됩니다.

OpenShift AI는 핵심 Red Hat 기술들을 통합하여 에이전틱 워크로드를 위한 프로덕션급 기반을 제공합니다. 여기서 핵심은 오퍼레이터 기반 모델로, 배포 모범 사례를 암호화하고 플랫폼 구성을 자동화합니다. 오퍼레이터는 자동 스케일링(autoscaling)과 관측성 통합에서부터 성능 튜닝과 수평/수직 스케일링 전략까지 모든 것을 간소화하므로 엔지니어링 팀은 인프라 관리가 아닌 에이전트 구축에 집중할 수 있는 시간을 확보할 수 있습니다.

OpenShift AI는 Red Hat OpenShift의 기능을 기반으로 생성형 AI 모델과 예측 AI 모델의 라이프사이클을 대규모로 관리하는 플랫폼입니다.

이 플랫폼은 Llama Stack, MCP와 같은 구성 요소와 기본적으로 통합되어 있어 개발 및 배포 사례를 통합하는 데 도움이 됩니다. 로컬 테스트에서 사용되는 인터페이스가 프로덕션 환경에서도 지원되므로 개발자는 Llama Stack의 OpenAI 호환 API를 기반으로 개발하고 에이전트를 자신감 있게 배포할 수 있습니다. MCP 서버는 프레임워크 간 호환성과 Red Hat 보안 및 관측성 스택과의 강력한 통합을 바탕으로 툴을 노출 및 관리하는 표준 방식을 제공합니다.

보안과 컴플라이언스는 OpenShift AI에 포함되어 있습니다. Red Hat의 오랫동안 규제 환경에 집중해 왔습니다. 따라서 에이전틱 애플리케이션도 이미 내장된 가드레일과 역할 기반 액세스를 통해 배포될 수 있습니다. 통합된 관측성 툴링은 팀이 에이전트의 결정을 추적하고, 툴 호출을 모니터링하고, 지속적인 평가를 지원하는 분석 파이프라인을 구축하는 데 도움이 됩니다.

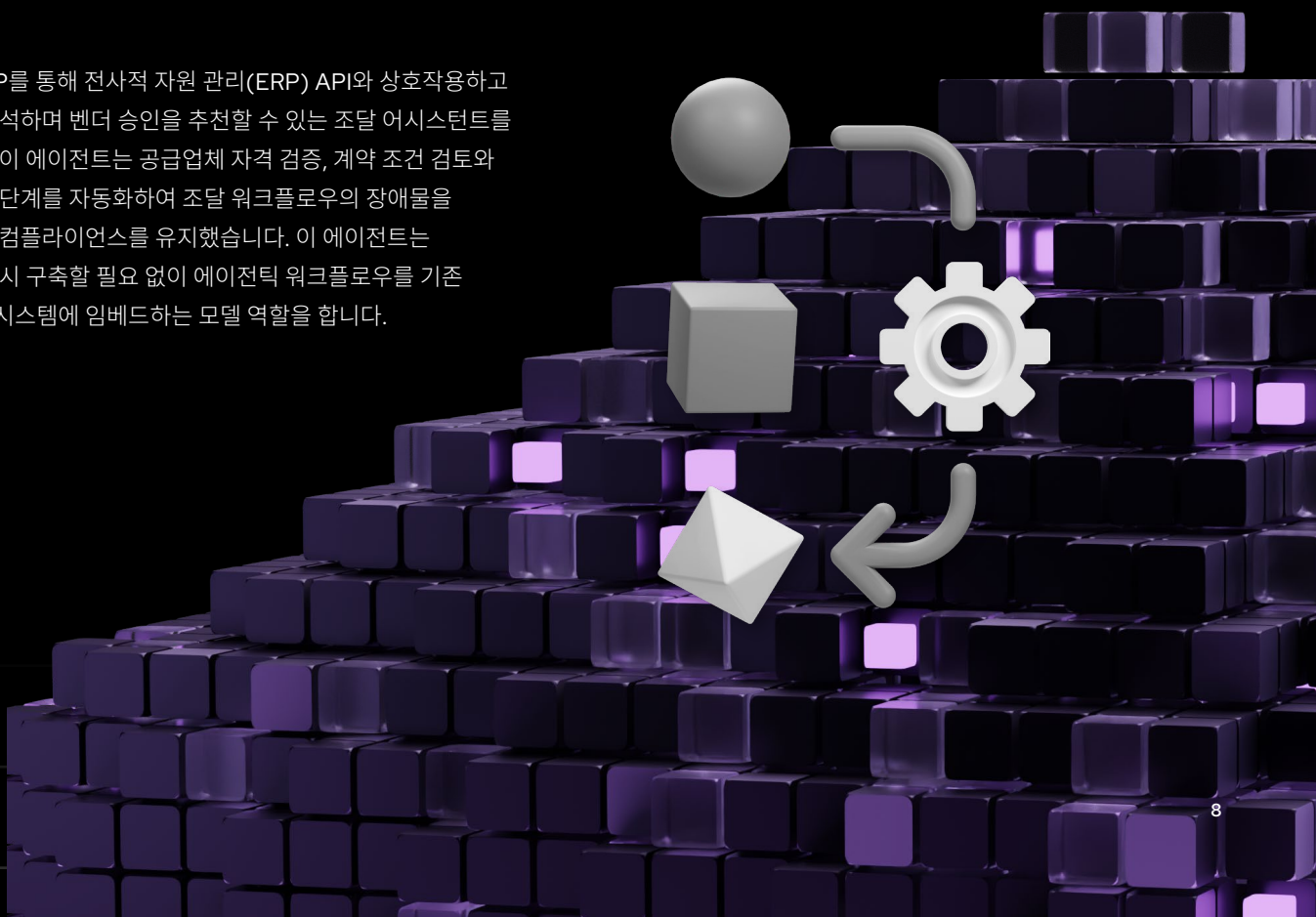
그 결과가 바로 AI 엔지니어, IT 팀, 보안 이해관계자가 더 효과적으로 협업할 수 있는 통합 플랫폼입니다. 간단한 지원 에이전트를 배포하든, 내부 자동화를 위한 멀티 에이전트 오케스트레이션 패턴을 배포하든 조직은 OpenShift AI를 사용하여 에이전틱 AI 시스템을 안전하고 반복적이며 대규모로 운영할 수 있습니다.

활용 사례

오늘날 조직이 에이전트를 구축하는 방법

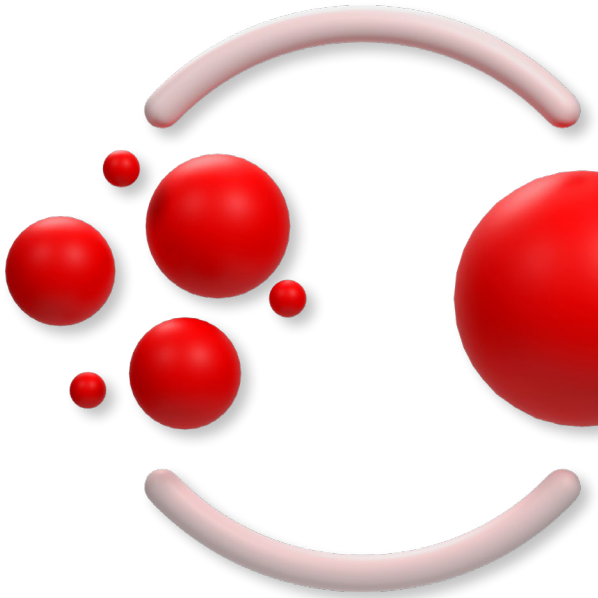
AI 중심 비즈니스 프로세스 자동화

한 고객이 MCP를 통해 전사적 자원 관리(ERP) API와 상호작용하고 정책 문서를 해석하며 벤더 승인을 추천할 수 있는 조달 어시스턴트를 개발했습니다. 이 에이전트는 공급업체 자격 검증, 계약 조건 검토와 같은 일상적인 단계를 자동화하여 조달 워크플로우의 장애물을 줄이는 동시에 컴플라이언스를 유지했습니다. 이 에이전트는 핵심 로직을 다시 구축할 필요 없이 에이전틱 워크플로우를 기존 엔터프라이즈 시스템에 임베드하는 모델 역할을 합니다.



3장

모델 컨텍스트 프로토콜(MCP): 에이전트-툴 통합 표준화



현대적인 에이전틱 시스템은 툴을 사용해서 LLM의 기능을 확장합니다. LLM이 추론을 제공하는 동안에는 조치를 취하고 엔터프라이즈 시스템에 액세스하며 실시간 정보를 수집하기 위한 툴이 필요합니다. MCP는 에이전트와 이러한 외부 툴 사이에 결여된 결합 조직을 제공합니다. MCP는 최근에 도입되었지만 에이전틱 AI의 새로운 표준으로 빠르게 채택되었습니다.

MCP 이전에는 툴 통합이 수동적이고 비일관적이었으며 확장하기 어려웠습니다. 개발자는 에이전트가 API를 검색하여 상호작용하는 방식을 정의하기 위해 사용자 정의 코드를 작성해야 했습니다. 이로 인해 팀 간 중복이 발생하고 업데이트의 안정성이 떨어졌으며 시스템이 변경될 때마다 리스크가 발생했습니다. MCP는 AI 워크플로우의 USB-C 표준처럼, 에이전트가 툴을 안정적으로 검색, 선택, 호출할 수 있는 상호 운용 가능한 모듈식 사양으로 이러한 프로세스를 표준화합니다.

MCP를 사용하는 개발자는 공통 프로토콜을 사용하여 기능을 툴로 노출합니다. 이러한 툴은 API, 데이터베이스, 비즈니스 시스템은 물론, 내부 유틸리티까지도 포괄할 수 있습니다. MCP 서버는 툴을 호스팅하고 문서화하며 보호하는 허브의 역할을 합니다. 이후 에이전트는 이 서버를 동적으로 쿼리하고, 툴 사용을 추론하고, 여러 시스템 전반에서 워크플로우를 호출할 수 있습니다.

이러한 표준화는 더 폭넓은 참여의 기회가 됩니다. MCP 기반 아키텍처를 통해 팀은 기존 툴 정의를 재사용하고 온보딩을 가속화하며 플랫폼 파편화를 줄일 수 있습니다. 또한 소규모 또는 대규모 LLM의 경우 조직은 MCP를 통해 모델을 재학습시키거나 로직을 재작성할 필요 없이 툴을 더욱 일관되게 사용할 수 있습니다.

그러나 MCP에 문제가 없는 것은 아닙니다. 툴 설명이 제대로 작성되지 않은 경우 에이전트에게 혼란을 가져와 오작동이나 할루시네이션으로 이어질 수 있습니다. 심지어 안전하지 않은 프롬프트 문자열이나 지나치게 광범위한 권한을 노출하는 툴은 악용의 매개체가 될 위험이 있습니다.

Red Hat은 거버넌스, 보안 고려 사항, 관측성을 MCP 서버 아키텍처에 직접 임베드하여 이러한 리스크를 완화하는 데 도움이 되는 MCP 게이트웨이를 개발할 계획입니다. MCP 서버가 OpenShift AI 내에 배포되면 MCP 서버는 Red Hat의 플랫폼 수준 정책 시행, 역할 기반 액세스 제어, 감사 메커니즘과의 네이티브 통합을 상속받습니다. 따라서 MCP를 통해 노출된 툴에 대한 액세스를 역할과 네임스페이스별로 엄격하게 관리할 수 있으며, Identity 및 권한 부여 정책을 준수할 수 있습니다.

또한 MCP 내에 호스팅된 툴 설명과 프롬프트 스키마의 취약점을 Red Hat OpenShift의 컨테이너 및 애플리케이션 보안 툴체인을 사용하여 자동 스캔할 수 있습니다. 이를 통해 플랫폼 팀은 프롬프트 인젝션을 허용하거나 적절한 입력 검증 없이 민감한 백엔드 시스템을 노출하는 툴과 같이 잠재적으로 위험한 툴 구성을 식별할 수 있습니다.

운영 관점에서 볼 때 Red Hat OpenShift 관측성 툴을 사용하면 에이전트-툴 상호작용을 지속적으로 모니터링할 수 있습니다. 팀은 툴 호출 패턴을 확인하고, 사용량 메트릭을 트래킹하고, 비정상적 동작에 대한 경고를 설정할 수 있습니다. 통합된 감사 로그는 툴 사용 및 의사 결정 사슬에서 추적성을 제공하므로 기업이 내부 및 외부 컴플라이언스 요구 사항을 충족하는 데 도움이 됩니다.

MCP를 다른 Red Hat AI 구성 요소(예: Llama Stack의 평가, 툴, 안전 API) 및 통합 플랫폼과 결합함으로써 고객은 MCP 서버를 안정적으로 배포하고 운영할 수 있어 확장 가능하고 보안에 중점을 둔 에이전틱 AI 워크플로우의 생성이 원활해집니다. 이를 통해 조직은 작업을 제안할 뿐만 아니라 직접 실행하는 에이전트를 구축할 수 있습니다.

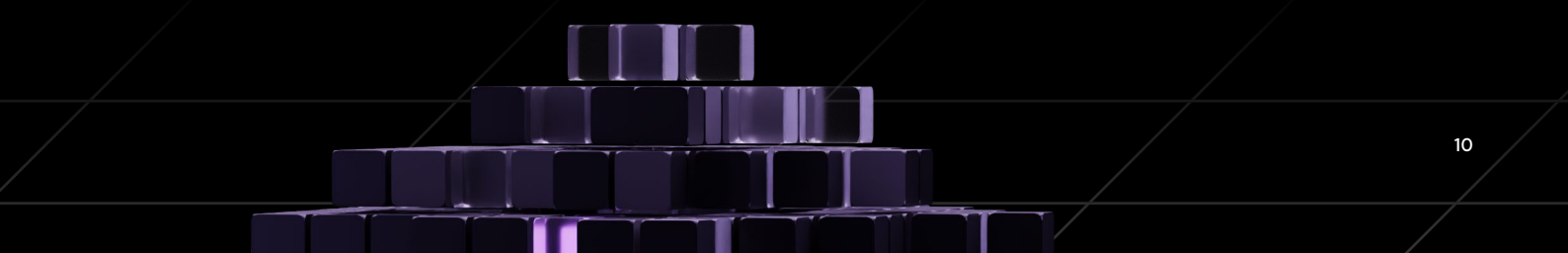
활용 사례

오늘날 조직이 에이전트를 구축하는 방법

대규모 고객 지원

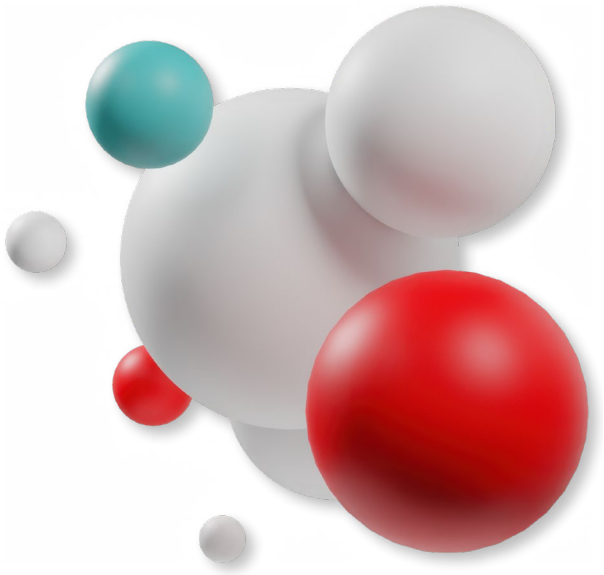
한 사이버 보안 회사는 대기 시간을 줄이고 인간 에이전트의 부담을 덜기 위해 AI 기반 지원 시스템을 구현하여 라이브 채팅과 티켓 해결을 자동화했습니다. 접수 에이전트가 문의를 처리하면 문의가 분류 에이전트로 전달됩니다. 그러면 분류 에이전트가 키워드 목록과 생성형 AI 감성 분석을 사용하여 감성 태그 지정, 긴급성 분석, 욕설 및 비속어 검사를 수행합니다. 문제 언어가 발견될 경우 수동 검토가 트리거됩니다.

그다음, 문제 해결 에이전트가 검색 증강 생성(RAG)을 사용하여 내부 문서와 과거 티켓을 가져와 응답을 생성합니다. 필요한 경우 라우팅 에이전트가 미해결된 티켓 또는 확신이 낮은 티켓을 인간 지원 담당자에게 에스컬레이션합니다. 인간 에이전트는 AI가 제안한 응답을 검증하며 특히 컴플라이언스에 민감한 콘텐츠에 더욱 주의합니다. Airtable 로그는 인간 에이전트가 재작성한 답변과 오류를 트래킹하여 지속적인 모델 개선을 지원합니다. 작업 종료 후에는 고객 만족도 확인을 위해 감성을 재평가합니다.



4장

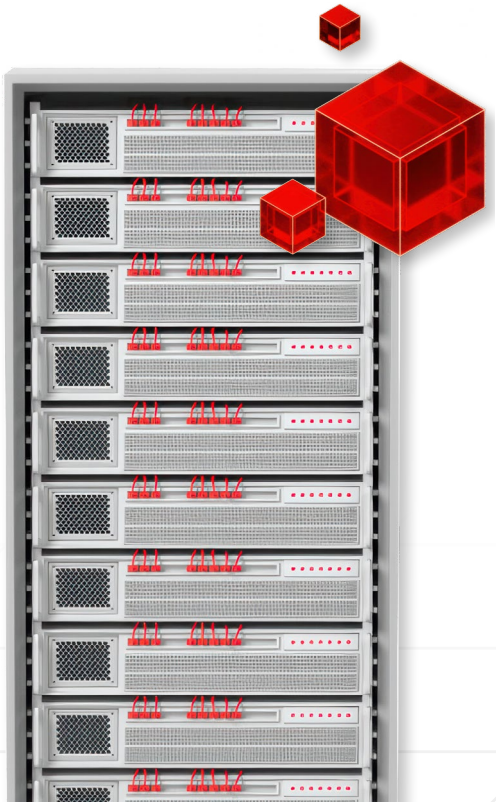
Llama Stack: OpenShift AI를 통해 제공되는 통합 AI API 서버



Llama Stack은 Red Hat의 통합 AI 컨트롤 플레인으로, 에이전틱 AI 애플리케이션의 개발, 배포, 관리를 간소화하도록 설계된 OpenAI 호환 API 서버입니다. 추론, 메모리, 툴 오케스트레이션, 평가를 위한 핵심 인터페이스 역할을 하는 Llama Stack은 개발자에게 다양한 환경에서 정교한 에이전트를 구축하는 데 필요한 일관성과 유연성을 제공합니다.

보통 호스팅된 서비스 형태인 다른 많은 솔루션들과 달리 Llama Stack은 조직이 자체 하드웨어 또는 클러스터에서 호스팅 방식의 환경을 만드는 데 도움이 됩니다. 데이터 주권이 필요하거나, 특정 인프라 요구 사항이 있거나, 특정 벤더에 종속되기를 원치 않는 조직의 경우 이러한 유연성이 반드시 필요합니다.

Llama Stack의 핵심은 단순한 추론을 넘어 OpenAI API와의 호환성을 포함한 전체 에이전틱 라이프사이클을 지원하는 표준 API 계층을 제공하는 것입니다. Llama Stack은 OpenShift AI와 직접적으로 통합되고 검색 증강 생성(RAG), 안전, 평가, 텔레메트리, 상황 인식 추론과 같은 일반적인 에이전트 태스크를 지원합니다. 개발자는 동일한 API, 라이브러리, 추론을 지속적으로 사용하여 경량화 로컬 배포부터 시작해 엔터프라이즈급 인프라까지 확장할 수 있습니다.



Red Hat은 팀의 준비 상태와 프로젝트 복잡성에 따라 Llama Stack을 도입할 수 있는 여러 방식을 지원합니다. 에이전틱 시스템이 처음인 팀은 툴 호출, 메모리, 컨텍스트 관리와 같은 사전 구성된 구성 요소를 갖춘 빌트인 클라이언트와 SDK를 사용하여 빠르게 시작할 수 있습니다. 이러한 기능은 개발을 간소화하고 복잡성을 완화하는 데 도움이 됩니다. 고급 사용자라면 OpenAI의 툴 사용 인터페이스와 API가 호환되고 다양한 인기 프레임워크를 지원하는 Llama Stack을 유용하게 활용할 수 있습니다. 따라서 팀은 재설계할 필요 없이 기존 에이전트 및 워크플로우를 통합하는 동시에 여러 환경에서 일관된 인터페이스와 라이프사이클 관리 툴을 계속해서 활용할 수 있습니다.

Llama Stack의 특징은 공급업체 간 AI 애플리케이션 개발을 표준화하는 오픈소스 API 계층과 서버입니다. Llama Stack은 OpenAI 호환 추론은 물론, 평가, 사후 학습, 벡터 저장소 등 다른 AI 기능을 위한 추가 API를 지원하고, RAG 및 에이전트 워크플로우를 지원하기 위한 보호 장치를 제공합니다. Llama Stack은 셀프 호스팅된 환경, 온프레미스 또는 클라우드에서 실행될 수 있어 조직이 유연하게 배포할 수 있습니다.

OpenShift AI 내에서 쿠버네티스 오퍼레이터는 자동 스케일링, 관측성, 액세스 제어와 같은 Llama Stack 라이프사이클 태스크를 관리하고 개발자와 플랫폼 팀에 에이전트를 안정적으로 확장하기 위한 통합 툴 세트를 제공합니다. 또한 Llama Stack에는 OpenTelemetry를 포함한 평가 및 텔레메트리에 대한 네이티브 지원이 포함되므로 팀은 AI 시스템 성능을 확인하고 안전 메트릭을 모니터링하며 프로덕션 환경 전반에서 에이전트 동작을 추적할 수 있습니다.

추론 외에도 Llama Stack은 에이전틱 시스템의 여러 유동적인 부분을 지원하도록 설계되었습니다. 호스팅된 모델과 로컬 모델 사이를 이어주고, TrustyAI와 같은 안전 툴에 대한 액세스를 표준화하고, MCP 서버와의 상호작용을 촉진할 수 있습니다. 개발자는 기술 검증(POC) 데모를 실행하든, 프로덕션 IT 워크플로우를 자동화하든, 에이전트를 테스트, 반복, 운영하기 위한 일관된 플랫폼을 확보하게 됩니다.

Llama Stack은 에이전트 오케스트레이션의 복잡성을 반복 가능한 모듈식 프로세스로 전환합니다. Red Hat 지원이 제공되고 OpenShift AI와 통합되는 Llama Stack을 통해 팀은 에이전틱 AI 시스템을 안전하고 예측 가능한 방식으로 확장하는 데 필요한 컨트롤 플레인을 확보할 수 있습니다.

AI 에이전트 구축에 대한 모듈식 접근 방식

Red Hat AI 플랫폼의 기능은 다음과 같습니다.

- **Llama Stack의 네이티브 기능 및 구현**을 통해 에이전트를 구축합니다.
- OpenShift AI에 **호환되는 Llama Stack 구현을 통합합니다**.
- **조직 자체의 에이전트 프레임워크를 사용**하고 Llama Stack API를 선별적으로 통합합니다.
- **핵심 기본 요소를 사용하여 구축**하고 자체 에이전트 프레임워크를 표준 워크로드로 관리합니다.

그림 3. 모듈식 개방형 프레임워크에 적합한 Llama Stack

활용 사례

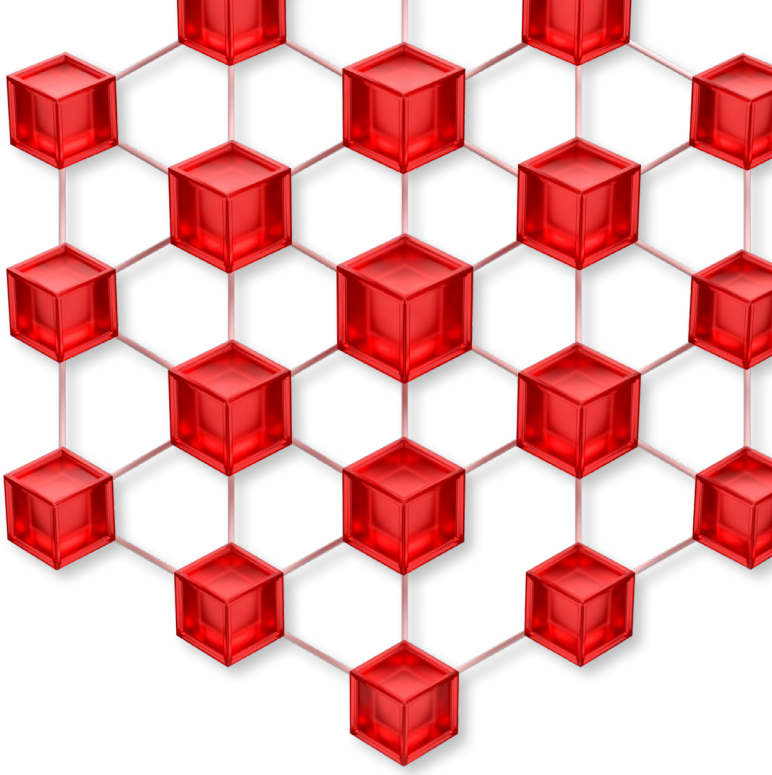
오늘날 조직이 에이전트를 구축하는 방법

개발자 생산성

한 소프트웨어 회사가 기존 Java 애플리케이션을 분석하고 현대적인 프레임워크에 맞춰 업데이트하기를 권장하는 코드 마이그레이션 어시스턴트를 구축했습니다. 이 에이전트는 Llama Stack을 추론에 적용하고, MCP 통합 툴을 사용하여 호환성을 검증하고 업그레이드 경로를 제안했습니다. 그 결과 마이그레이션 프로세스가 간소화되어 기술 부채가 줄고 애플리케이션 복원력이 강화되었습니다. 또한 개발 팀은 기존 코드를 재작성하는 데 시간을 할애할 필요 없이 기능 혁신에 집중하게 되었습니다.

결론

Red Hat AI: 에이전틱 AI 성공을 위한 기술 기반



에이전틱 AI는 더욱 지능적인 적응형 애플리케이션을 약속하지만 표준화가 이루어지지 않을 경우 엔터프라이즈 팀은 파편화, 비효율성, 운영 리스크에 직면할 수 있습니다. Red Hat AI는 이러한 시스템을 안정적으로 구축하는 데 필요한 토대를 제공합니다. OpenShift AI, MCP, Llama Stack과 같은 구성 요소를 통해 Red Hat은 AI 에이전트를 기술 검증에서 프로덕션까지 적용하는 일관된 기반을 제공합니다.

추론, 툴 사용, 메모리, 안전, 평가를 통합 플랫폼에 결합함으로써 Red Hat AI는 팀이 AI 에이전트를 실험, 확장, 운영하는 방법을 간소화합니다.

- 개발자는 프로덕션급 API에 대한 액세스 권한을 확보합니다.
- 플랫폼 팀은 라이프사이클을 관리하고 보안을 시행할 수 있습니다.
- 조직은 투자 자산을 보호하는 오픈 표준을 통해 혜택을 볼 수 있습니다.

간단한 내부 봇을 배포하든, 멀티 에이전트 시스템을 설계하든, Red Hat AI로 보안 중심의 엔터프라이즈급 에이전틱 애플리케이션을 조직 고유의 조건에 맞게 반복 가능한 방식으로 구축할 수 있습니다.